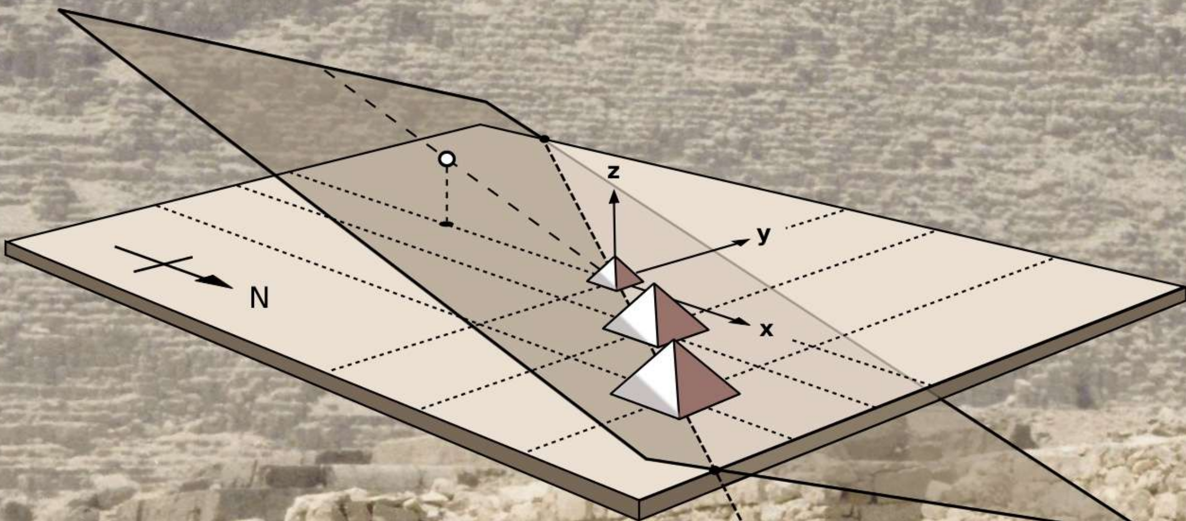
Hans Jelitto

Planetary Correlation of the Giza Pyramids

P4 Program Description



Hamburg, June 2015

Planetary Correlation of the Giza Pyramids

P4 Program Description

Hans Jelitto

1. edition September 2014

2. edition June 2015. Beside minor revisions, the book title is rearranged, section 4.7.4 about transit series has been added, the P4 source code has been slightly worked over, and two more publications of the author [3, 4] can be downloaded via provided links.

Hans Jelitto, Ewaldsweg 12, D-20537 Hamburg, Germany

Hamburg, June 2015

This work is dedicated
to my parents
Karl and Käthe Jelitto

Preface

A correlation between the pyramids of Giza and the inner planets of our solar system has been found. This manual is not only a user guide for the P4 computer program regarding this correlation, but it also provides some basic information about the technical and theoretical background, including archaeological, mathematical, and astronomical aspects. Further details and several other related results, which are not included here, are presented in the book "Pyramiden und Planeten" (in German). A subsequent book (in preparation) will provide more details about the results given here. However, we tried to include all the necessary information so that the reader can work properly with the manual and the program. This manual is intended for scientists and for anyone, who is interested in the secret of the pyramids.

For a basic overview of the planetary correlation, it is sufficient to read chapter 1 (introduction), sections 3.1.1–3.1.3, 4.6.3, 4.10, and chapter 5 (summary). Related lecture videos of the author on YouTube (English subtitles) can be found with the search items "pyramiden planeten jelitto." For the essential ideas of the calculations, but not in the programming itself, chapter 4 provides the underlying basic concepts.

Additionally, the appendix contains the entire source code of the program, which is provided mainly for programmers. When printing the manual, and if the source code is not needed, the corresponding double pages 83–136 can be omitted. If possible, the printout should be in color and double-sided if the printer supports that feature. (So, an adequate ring binder can be made.)

Contents

Pyramids of Giza from the South

1. Introduction

The purpose of the P4 program is to perform astronomical calculations with respect to the planets of our solar system and the three pyramids of Giza (Fig. 1). P4 is based on the French planetary theory VSOP87 [1, 2] (see more below). The fundamental idea is that a correlation exists among the three inner planets and the three pyramids in Giza. The first papers about this hypothesis were published in the Austrian journal *Grenzgebiete der Wissenschaft* (in German) in 1995 [3, 4]. About two years earlier, the development of the program P3 began, allowing for the mathematical comparison of pyramid positions and planetary positions. Because of three equations (see section 3.1.1) that define the size of each pyramid, it seems that the Cheops Pyramid (Great Pyramid), the Chefren Pyramid, and the Mykerinos Pyramid represent the planets Earth, Venus, and Mercury, respectively. Furthermore, the pyramid positions correlate with the planetary positions. Because the planets are moving all the time, their arrangement and distances between each other change continuously. This implies that the geometric arrangements of pyramids and planets match for only one or a few points of time. Such dates were found, depending on the mathematical approach and further boundary conditions. So, among other things, the program calculates the dates when Mercury, Venus, and Earth stand in a constellation according to the arrangement of the Giza pyramids (see Fig. 1 and [5, p. 95]). The data in Fig. 1 were measured by Sir W. M. F. Petrie [6, 6a]. Excellent geographical maps, reproducing the pyramids in Egypt, are available, for example, in Cairo [7, 8].

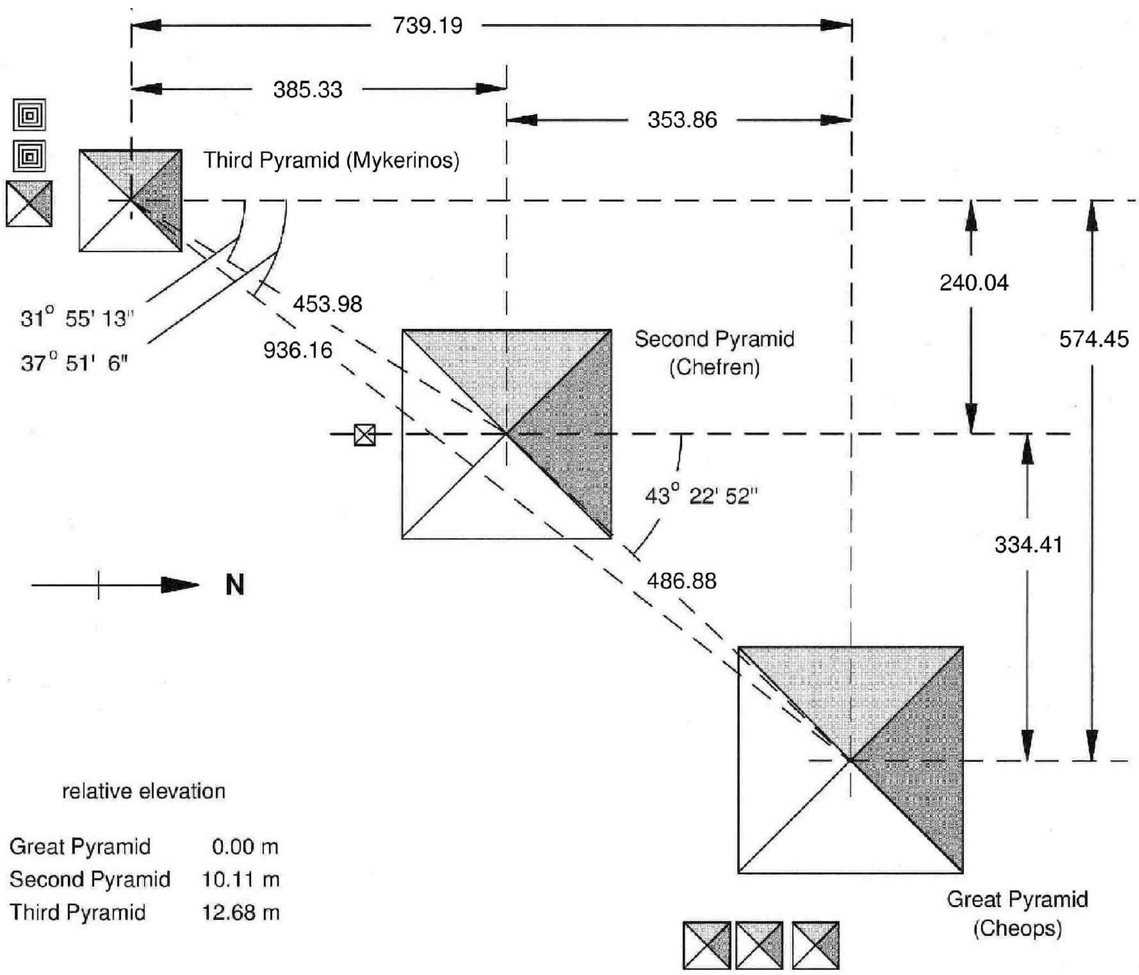


Figure 1: *Alignment of the Giza pyramids with measured data of W. M. F. Petrie [6, 6a] (distances in meters). Two numbers are slightly corrected: The large diagonal is 936.16 m instead of 936.19 m, and one angle is 31° 55' 13" instead of 34° 10' 11" (explanation in [5, p. 96; 6, p. 125]). The relative elevations stem from S. Perring (see: [9, part IV, map 1]). Detailed information is provided in the drawings of Maragioglio and Rinaldi [9]. The angles were calculated from the original distances, given in inches (1 inch = 2.54 cm).*

The archaeological state of knowledge is that the three great pyramids in Giza were built by the Egyptian pharaohs Chufu, Chaefre, and Menkaure within the 4th dynasty. In addition to these Egyptian names, the Greek names are Cheops, Chefren, and Mykerinos. In the archaeological chronology, the 4th dynasty is dated roughly between the years 2600 and 2480 BC [10, vol. I, p. 970]. ("BC" means "before Christ.") On the other hand, in 1987 and 1994 it was reported that the age of several buildings of the Old Kingdom, including the pyramids of Giza, was determined independently with the "accelerator mass spectrometry" (AMS) [11, 12], ordered by the ETH Zürich in Switzerland. This is a modern variant of radiocarbon dating in which a particle accelerator is used to determine the amount of radioactive ^{14}C-isotopes. The result is that, for example, the Cheops Pyramid has to be dated between the years 3030 BC and 2905 BC with a probability of 95 %. This is a discrepancy of approximately 400 years! Because it is impossible to shift the chronology of the pharaohs by 400 years, the reader should keep this point in mind (see details in [5, pp. 361 ff.]).

The first program version was named P3 because of the 3 great pyramids in Giza and the 3 planets Mercury, Venus, and Earth. It was used for computing the astronomical tables in the book "Pyramiden und Planeten" [5]. After this book was published in 1999, another correlation was found, namely between the planetary positions and the chamber positions in the Great Pyramid – with an unexpected connection between both correlations. This led to an extension of the program P3 with several other options. The new program name is P4 because it is an upgrade of P3 and includes the fourth planet Mars. P4 covers all features of P3, the processing speed has been optimized, and the application is much easier. The results, which cannot all be provided in this manual, are described in detail in the subsequent book, here named "Book 2" [13]. Unfortunately, until now all publications are in German. As a first remedy, this description is written in English.

The comparison of the arrangements is performed mathematically by a coordinate transformation. An interesting point of this correlation is that, by using the transformation of the planetary arrangement, the position of the Sun can be precisely transferred to the pyramid area (Fig. 2). This means that we have a "Sun position" at the Giza plateau. Furthermore, the positions of the chambers define another "Sun position" inside the Cheops Pyramid. In the following, "Sun position" is written in quotations because here we do not refer to the real Sun but to the corresponding position in the pyramid area. Later, we will also find a "Mars position" in the Cheops Pyramid.

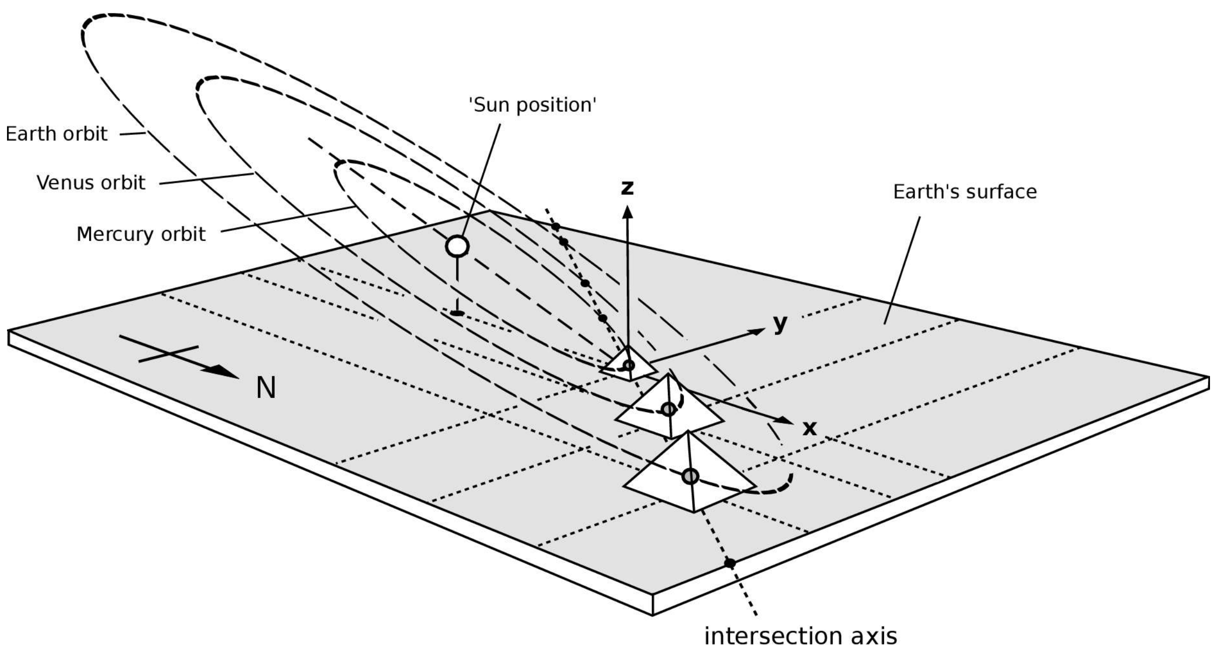


Figure 2: *Schematic representation of the Earth's surface around the Giza pyramids and the orbits of the three inner planets Mercury, Venus, and Earth, after adaption of pyramid and planetary positions. The geometric arrangement of constellation number 12 (section 3.4.3) looks very similar. Because of different inclinations, the orbits are slightly tilted against each other. This fact is neglected in the drawing but is taken into account in the calculations.*

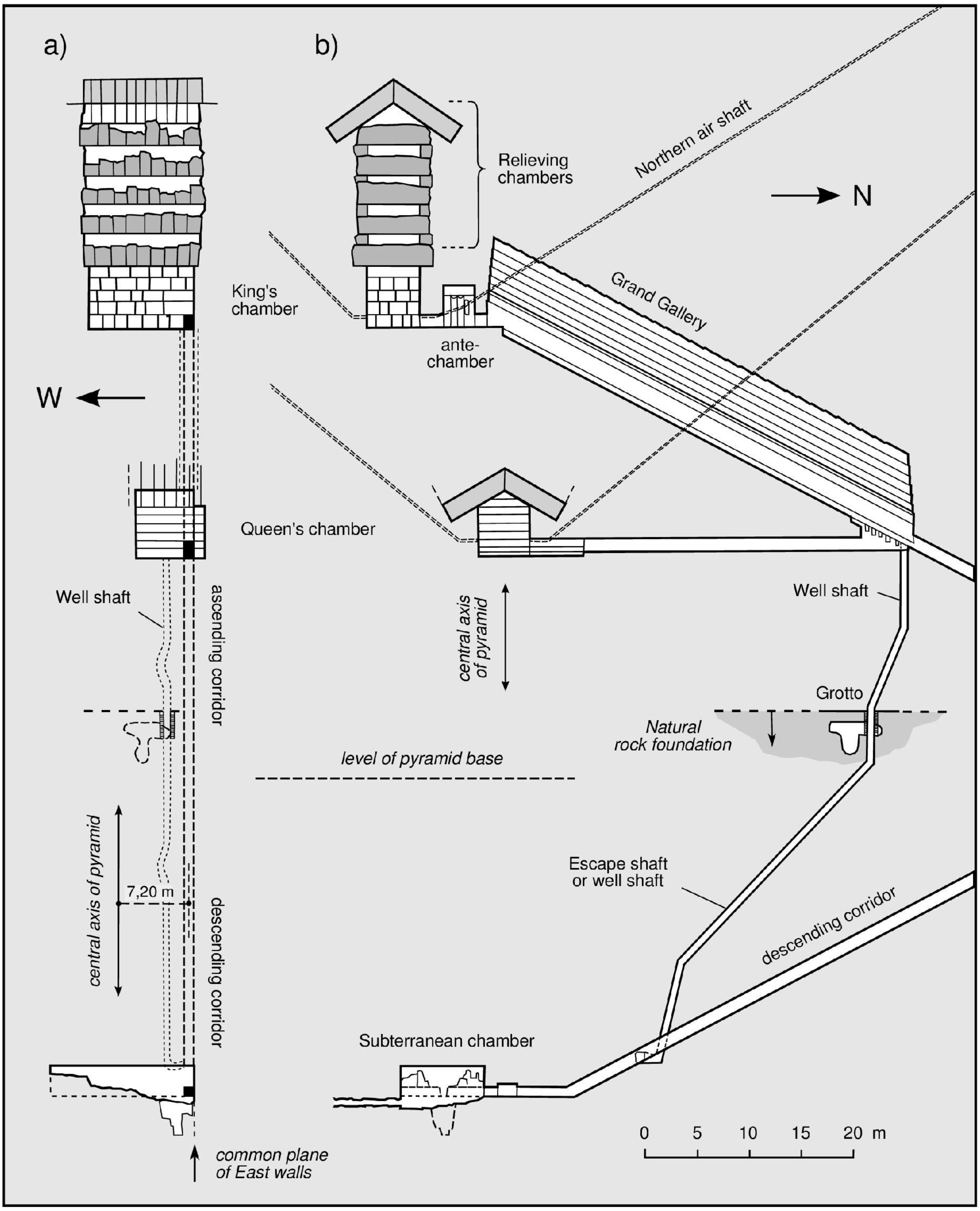


Figure 3: *Chamber and corridor system inside the Great Pyramid, as seen from the south* ***(a)*** *and from the east* ***(b)****. The figure is based on drawings of Maragioglio and Rinaldi [9, part IV, maps 3–7] (arrangement of (a) and (b) like in Ref. [14] of R. Stadelmann).*

In Fig. 3 the remarkable system of chambers and corridors in the Cheops Pyramid is given, which also plays a major role in the correlation between pyramids and planets. The names of the chambers, like “King's chamber” and “Queen's chamber,” originate from classical archaeology and are based on the explanation that the pyramids were tombs of the pharaohs. At first glance, this explanation seems reasonable because it is written in countless books. However, this interpretation is not necessarily correct because a mummy has never been found in an Egyptian pyramid! Numerous mummies of kings and queens have been discovered, but all of them were found in hidden tombs in the desert, like in the Valley of the Kings. More about this is provided in [5].

Some of the boundary conditions for the comparison of pyramid and planetary positions are the following: The “Sun position” can be fixed by placing the “Sun” on the Earth’s surface exactly on the center line 726 m south of the Mykerinos Pyramid (see Fig. 4). The “Sun position” can also be free in the two horizontal coordinates and would be fixed only to the Earth’s surface by adapting the planetary positions, or it can be free in all 3 dimensions. In order to get a better idea, an example of the two systems “pyramids” and “planets,” using a 3-dimensional fit, is shown in Fig. 2. The two planes, “Earth’s surface” and “ecliptic plane” (plane of the Earth orbit), are not coplanar but tilted against each other. (More about this is given in section 4.10.2.)

Furthermore, the P4 program computes the dates of “linear constellations” of the celestial bodies Mercury, Venus, Earth, Mars, and the Sun, which means that the planets have nearly the same ecliptic longitude. These “linear arrangements” of celestial bodies (conjunction and opposition) are called “syzygy.” In addition, the exact geocentric transit phases, when Mercury or Venus passes the Sun's disk, can be determined. All options and parameters of the P4 program are provided in chapter 3.

The calculations were performed as accurately as possible. Strong emphasis was put on the use of the most recent and precise scientific data. This refers to the astronomical data and computations as well as to the archaeological data. Concerning the exact dimensions of the Cheops-Pyramid, the latest results were not always the most accurate one. The reason is that due to weathering effects the measurement conditions at the Pyramid about 100 years ago were partly better than today. This is discussed in detail in Ref. [5, pp. 249–255]. However, the corresponding small differences are important only for the exact shape but not for the size of the Cheops-Pyramid and do not have any effect on the results in this manual.

2. General technical information

The astronomical calculations are based on the planetary theory VSOP87, developed by P. Bretagnon and G. Francou at the Bureau des Longitudes, Paris (today: IMCCE, Institut de mécanique céleste et de calcul des éphémérides) [1, 2]. VSOP means “Varations Séculaires des Orbites Planétaires” and 87 is the year of publication (1987). The files for the VSOP87 theory can be downloaded from the FTP server of the IMCCE homepage: ftp://ftp.imcce.fr/pub/ephem/planets/vsop87/.

A multi-parameter fit program “FITEX” (fit experiment) [15, 16], included in P4, was developed by G. W. Schweimer, Zyklotron-Laboratorium, KfK (Kernforschungszentrum Karlsruhe, today: KIT, Karlsruher Institut für Technologie). For calculating calendar dates, an algorithm from the book “Astronomical Algorithms” by J. Meeus [17] is converted into a subroutine of P4. The conversion of “terrestrial time” (TT) to “universal time” (UT) is performed by using ΔT = TT – UT, calculated by F. Espenak and J. Meeus (“NASA Eclipse Web Site,” Polynomial Expressions for Delta-T). The P4 program, all subroutines, and other programs from the author are written in Fortran. The whole package of programs, and all associated files, can be downloaded from the author's homepage: www.pyramiden-jelitto.de/downloads.html. Note: Through the provided links, many of the given references can also be downloaded from the Internet. For details of the theoretical basis, see chapter 4.

The previous program (P3) was originally developed with the IBM Professional Fortran 77 Compiler (Version 1.0, Ryan-McFarland) using the SPF/PC editor and the Windows operating system. Later on, we switched to the GNU-Fortran compiler g77 with Ubuntu Linux, and then to GNU gfortran. Of course, it is possible to use other Fortran compilers and other operating systems.

It would also be of interest to port the program to languages like C, C++, or Java. However, because the architectures of these programming languages are quite different to Fortran, it is probably easier to write a new program. In addition, it would be a good test of the results if the calculations were performed independently and based, for example, on a theory other than VSOP87.

2.1 Data files and other related programs

Table 1 contains a compilation of all files belonging to the astronomical program P4. A few comments about other available programs, and more information about the files in Table 1, are provided in the following. In this section, program, catalogue, and file names are highlighted in blue.

Table 1: *All 32 (36) files of the P4 program: program, text, and data files (download from the author's homepage).*

File	Brief description
p4.f95	Fortran source code, (p4-4.f95, p4-4.pdf: parallelized version, see footnote on page 82)
p4-32	Executable program file for a 32-bit system (can also be used on a 64-bit system)
p4-64	Executable program file for a 64-bit system, (p4-4-64: parallelized version, 4 threads)
p4-32.sh	This shell-script clears the screen display and starts p4-32.
p4-64.sh	Shell-script – as above – starts p4-64, (p4-4-64.sh: starts p4-4-64)
p4-manual-06-2015.pdf	User manual with details of the P4 program and main aspects of the planetary correlation concerning the Giza pyramids (this text)
README	Notice to the theory planetary solutions VSOP87 from Bretagnon and Francou
vsop87.doc	Technical information about the VSOP87 theory from Bretagnon and Francou
out.txt	Output file (if it does not exist, it will be created by the program with the corresponding option)
inedit.t	Ancillary input file (can be used to create a new set of parameters for inparm.t)
inparm.t	This file contains all input parameters for the quick start options 1 to 15, for Tables 39 to 51 in [5], and for Tables 17 to 33 and 35 to 37 in [13].
inpdata.t	Parameters for FITEX and coordinates of pyramid and chamber positions in Giza
inserie.t	Dates of transit series for Mercury and Venus (used only at program start)
invsopl.t	Shortened VSOP87D data for the planets Mercury to Mars, typewritten manually from: J. Meeus, "Astronomical Algorithms" [17, pp. 381 ff.]
invsop3.t	Polynomial representation of orbital elements, derived from VSOP82 and taken from: J. Meeus, "Astronomical Algorithms" [17, pp. 200 ff.]
VSOP87A.mer	Mercury: VSOP87A, heliocentric rectangular coordinates, ecliptic J2000.0
VSOP87A.ven	Venus: ... " ...
VSOP87A.ear	Earth: ... " ...
VSOP87A.mar	Mars: ... " ...
VSOP87A.jup	Jupiter: ... " ...
VSOP87A.sat	Saturn: ... " ...
VSOP87A.ura	Uranus: ... " ...
VSOP87A.nep	Neptune: ... " ...
VSOP87A.emb	Earth-Moon barycenter: ... " ...
VSOP87C.mer	Mercury: VSOP87C, heliocentric rectangular coordinates, dynamical equinox
VSOP87C.ven	Venus: ... " ...
VSOP87C.ear	Earth: ... " ...
VSOP87C.mar	Mars: ... " ...
VSOP87C.jup	Jupiter: ... " ...
VSOP87C.sat	Saturn: ... " ...
VSOP87C.ura	Uranus: ... " ...
VSOP87C.nep	Neptune: ... " ...

The files README and vsop87.doc in Table 1 provide details about the theory versions of VSOP87 and are given directly by the authors Bretagnon and Francou (download from the homepage of IMCCE, link provided previously). The file out.txt contains the results after running P4, if the output parameter is not set otherwise. The next six files in Table 1, beginning with "in..." are input files, necessary to run P4. In the file inparm.t, all parameter sets for the quick start options are compiled. File inedit.t is a combined input-output file. During each run, all input parameters are stored at the end of this file. This ancillary file helps to create new parameter sets, which can be added as new quick start options to the file inparm.t. In this case, the subroutine "inputdata" in p4.f95 has to be properly adapted. The input parameters in the file inedit.t can also be edited manually and are adopted by the program with the quick start option 999. This allows for testing new parameter sets. The file inpdata.t contains parameters for the subprogram FITEX as well as the exact coordinates of the pyramid chambers and of the pyramids themselves. In inserie.t, several dates (*JDE*) are listed to determine the serial numbers of the first Mercury or Venus transits, found after program start. In invsop1.t are the shortened parameter series of the VSOP87D version, taken from [17, pp. 381 ff.]. The file invsop3.t contains coefficients for polynomials of third degree for the elements of planetary orbits, deduced from the version VSOP82 [17, pp. 200 ff.]. All remaining files from VSOP87A.mer to VSOP87C.nep represent full versions of the planetary theory [1, 2] with a very high accuracy. They are also available from the FTP server of the IMCCE homepage.

This paragraph provides some general information about the other programs used within the present pyramid research; the P4 and TOPO programs are new and used in reference [13]. TOPO calculates the exact volume of the Earth, including the volume of all ice and land masses. All other programs, including P3, are used and described in the first book [5]. This includes the programs FORM, SEKAN, PYT, and 7916, which enable geometric calculations concerning the shapes of the three pyramids of Giza, especially their casing angles. The program DATUM-2 converts the time system "Julian Ephemeris Day"[1] into a calendar date and is based on an algorithm from the book by Jean Meeus [17, p. 63]. The program SKYGLOBE [18] is a "planetarium" simulation of the sky and shows the celestial bodies like stars, planets, Sun, and Moon, as well as Milky way and constellations for every date and location on Earth. It was written by Mark A. Haney as shareware and is available at http://astro4.ast.vill.edu/skyglobe.htm. In this project, it has been applied only to check the "ORION correlation" propounded by R. Bauval and A. Gilbert [19]. When the positions and proper motions of the corresponding stars were taken into account for a quantitative analysis of the "ORION theory," large errors and deviations were found. On the one hand, Bauval and Gilbert were the first to correlate the pyramids with celestial bodies. On the other hand, their ORION hypothesis did not pass the test [5, pp. 157 ff., 349 ff.]. The analysis in [5] is based on the Star Catalogue PPM (Positions and Proper Motions) [20, 21].

For those who are interested, the text, formulas, and most figures, including the book cover, were created using Ubuntu with OpenOffice (now LibreOffice), Inkscape, and GIMP.

2.2 How to start the program

The P4 program does not need any installation. After downloading and unpacking the files, the easiest way is to store all of them in the same folder (directory), which can be named "P4," for example. It is assumed now that the operating system is Linux, because the P4 program was finally developed on the Linux distribution Ubuntu. If another operating system is installed, it is normally necessary to compile the source code p4.f95 again. In the case of a Windows system, p4.f95 can be compiled with a Windows compliant Fortran 95 compiler (e.g., ifort or gfortran with MinGW or Cygwin). Other possibilities are to create a Linux partition beside Windows, to use a Linux life CD like Knoppix, or to apply "VirtualBox." Special characters are not used in the program output, so for character encoding, the Unicode UTF-8 or ISO 8859-15 can be applied.

1 In order to keep consistency with [5] and with the notation of Meeus [17], JDE ("Julian Ephemeris Day" or "Julian Day") was used, based on terrestrial time (TT). Today, JD respectively JD(TT) has the same meaning.

After creating the folder, we open a terminal in Ubuntu (classic Gnome) through the menus "Applications – Accessories – Terminal." (In most cases, a terminal window width of 80 characters is sufficient; only one option needs a line length of 148 characters – see section 3.3.5.) In the following, all texts on the monitor screen like commands, menus, input data, and program results are printed in blue (not program names and file names). If, for instance, the folder has the name "P4" and is located in the path `~/Desktop/P4$`, we type the following commands at the command line in the terminal: `cd Desktop/P4` ↵. Now we are in the right folder. The sign "↵" denotes the return key. To start the program on a 64-bit computer system, we type `./p4-64.sh` ↵, which clears the screen, and the start menu appears. Another possibility is to start P4 directly with `./p4-64` ↵ without clearing the screen. If the program does not start type `chmod +x p4-64*` ↵. In the case of a new compilation of the source code with GNU Fortran, use the command `gfortran -static -O3 -Wall p4.f95` ↵. For a 32-bit system, the files `p4-32.sh` or `p4-32` have to be used. (Replace `64` by `32` in the above commands.) That's it! In the next chapter we will see how to proceed.

3. Program features

After having typed the start command, the main menu appears on the monitor:

```
                        -----------------------
                         PLANETARY CORRELATION
                         Program P4, June 2015
                        -----------------------

      pyramids of Giza         chambers, Great P.       transits, syzygy
    -----------------------------------------------------------------------
     3D Mer. at aph. (1)      3D Mer. at per. (6)      Mercury tr.    (11)
     2D Mer. at aph. (2)      Keplers equ.    (7)      Venus tr.      (12)
     const. 12, 3088 (3)      const. 12, 3088 (8)      syzygy, 3 pl. (13)
     1.5 days, 3088  (4)      1.5 days, 3088  (9)      syzygy, 4 pl. (14)
     near aphelion   (5)      F minimized    (10)      TYMT-test      (15)
    -----------------------------------------------------------------------
       info (111)   detailed options (0)   (0..15 or book options) : _
```

The date in the title indicates the last update of the program. In the table we find three different categories. The first options, (1) to (5), belong to the pyramids in Giza, the options (6) to (10) have to do with the chambers in the Great Pyramid, and the options (11) to (15) represent planetary conjunctions. The latter case includes different astronomical events: The three inner planets or the four inner planets of our solar system stand in conjunction (syzygy). Additionally, if Mercury or Venus are in conjunction with the Sun, it happens sometimes that they pass in front of the solar disc, which is called a "transit." However, before confusion arises, the astronomical relationships are explained in more detail in the following sections.

3.1 Quick start options (1) – (15)

Normally, about 10 to 15 different parameters have to be fixed before the astronomical calculation starts. These parameters determine, for example, the kind of astronomical event, the used VSOP-version, the coordinate system, the mode of calculation for the "Sun position," the time period to be examined, the complexity of the output, and so on. In order to avoid this, the general quick start options (1) to (15) start the program with predefined parameters just after having typed a short number. For example, typing `12` ↵ makes the program calculate all Venus transits for the years from 1500 AD to 4000 AD. ("AD" means "Anno Domini" or "after Christ.") The program output is:

```
                          TRANSITS OF VENUS
         (geocentric transit phases, terrestrial time TT)
                          < option  12 >

 VSOP87C, comb. search,     ecliptic of date,              all Venus transits
      Period   (years) from   1500.00  to   4000.00,  Jul./Greg. calendar

co/p    date/  time:   I         II       nearest     III        IV    sep["]a  S
===============================================================================
    26. May  1518  22:32: 2  22:49:14   1:59:45   5:10:16   5:27:28 -505.3   3
    23. May  1526  16:17:38  16:38: 9  19:14:43  21:51:18  22:11:49  666.7   5
---- (Greg. cal.) -------------------------------------------------------------
  v  7. Dec. 1631   3:53:17   5: 2:16   5:20:49   5:39:22   6:48:20  939.3/  6
     4. Dec. 1639  14:58: 4  15:16:26  18:26:47  21:37: 8  21:55:29 -523.6/  4
     6. June 1761   2: 2:20   2:20:35   5:19:30   8:18:25   8:36:40 -570.4   3
     3. June 1769  19:15:49  19:34:52  22:25:36   1:16:20   1:35:23  609.3   5
     9. Dec. 1874   1:49:12   2:18:56   4: 7:22   5:55:49   6:25:33  829.9/  6
     6. Dec. 1882  13:56:41  14:17:10  17: 5:54  19:54:38  20:15: 7 -637.3/  4
     8. June 2004   5:14:47   5:34:13   8:20:49  11: 7:24  11:26:51 -626.9   3
     6. June 2012  22:10:56  22:28:53   1:30:43   4:32:33   4:50:30  554.4   5
    11. Dec. 2117   0: 2:31   0:25:39   2:52: 8   5:18:38   5:41:46  723.6/  6
     8. Dec. 2125  13:19:29  13:43:10  16: 5:49  18:28:28  18:52: 8 -736.4/  4
    11. June 2247   8:51:10   9:12:31  11:42:27  14:12:24  14:33:45 -691.3   3
     9. June 2255   1:17:39   1:34:41   4:47:36   8: 0:31   8:17:33  491.9   5
    13. Dec. 2360  22:47:17  23: 7:30   1:58:44   4:49:57   5:10:10  625.7/  6
    10. Dec. 2368  12:44:56  13:15:24  15: 0:28  16:45:33  17:16: 0 -836.4/  4
    12. June 2490  12: 1:48  12:25:17  14:39:42  16:54: 7  17:17:36 -741.1   3
    10. June 2498   4:12: 4   4:28:32   7:48:35  11: 8:38  11:25: 6  442.7   5
    16. Dec. 2603  21:14:54  21:33: 8   0:44:29   3:55:49   4:14: 3  517.1/  6
  v 13. Dec. 2611  12:36:50  13:40:18  14: 6: 9  14:31:59  15:35:27 -934.8/  4
    15. June 2733  15:45: 8  16:13:18  18: 0:58  19:48:39  20:16:49 -808.3   3
    13. June 2741   7:17: 8   7:33: 5  11: 0:24  14:27:43  14:43:40  385.6   5
    17. Dec. 2846  20:24:29  20:41:44   0: 5:13   3:28:41   3:45:55  432.1/  6
  v 14. Dec. 2854     --        --     13:14:26     --        --   -1026.7/  4
    16. June 2976  18:54: 5  19:27:43  20:53: 7  22:18:32  22:52: 9 -850.5   3
    14. June 2984  10:10:33  10:26: 9  13:58:46  17:31:23  17:46:59  336.3   5
->  18. Dec. 3089  19: 1:49  19:18:10  22:53:36   2:29: 2   2:45:23  320.6/  6
  v 20. June 3219  22:31:18  23:28: 6   0: 0: 6   0:32: 6   1:28:55 -908.1   3
    17. June 3227  13: 3:37  13:18:56  16:55:19  20:31:43  20:47: 2  293.4   5
    20. Dec. 3332  18:14:30  18:30:23  22:12: 4   1:53:44   2: 9:38  235.5/  6
  v 22. June 3462   1:48:43     --      2:46:32     --      3:44:19 -948.1   3
    19. June 3470  15:51:28  16: 6:35  19:46:41  23:26:48  23:41:55  247.9   5
    23. Dec. 3575  17: 7:58  17:23:32  21:10:32   0:57:31   1:13: 5  131.5/  6
  v 24. June 3705     --        --      5:35:19     --        --    -989.3   3
    21. June 3713  18:30:27  18:45:25  22:27:21   2: 9:18   2:24:17  215.2   5
  c 25. Dec. 3818  16:23: 6  16:38:31  20:27:15   0:15:58   0:31:22   41.1/  6
    24. June 3956  21:17:37  21:32:30   1:16:53   5: 1:17   5:16:10  175.2   5
===============================================================================
 Computed constellations:     11092                 ("/" means ascending node)
 Tested planet. passages:      1564
 Detected transits      :        37
 Centr./grazing transits:    1 /   6       CPU-time  0: 0: 0.728 -- end of run.
```

The general appearance of the printed output is always similar. The first line shows the title of the program run followed by the second line, giving some basic information. In the third line we find the number of the selected option, which is often a quick start option. Two up to five lines follow, providing the remaining information in a brief form so that it is later possible to understand what has been calculated. These two up to five lines include the following data: the theory version of VSOP, the astronomical coordinate system, some data about the planets, pyramids or chambers, the time period, the allowed angular range (e.g., the range of the ecliptic longitudes), and other information.

In principle, there are two kinds of output of different magnitudes. At first, each astronomical event, like a transit, is written down in a single line as provided in the previous table. This kind of output is useful to get an overview when large time periods are investigated and when many planetary constellations are found. The other possibility is to characterize every astronomical event by much more information in several lines. At the end of the output, one or more lines give a summary of the program run. This includes, for example, the number of calculated and detected astronomical events as well as the CPU-time in hh:mm:ss.sss. More information about these different kinds of output is provided in section 3.4.

The following provides an example of an extended output for each found constellation. In order to illustrate it and avoid an output, which is too long, the time limits (3000 to 3200 AD) are chosen in such a way that only one planetary constellation is found and printed. The calculation is performed with a simple approach by iteratively solving Kepler's equation. In this program run, the main condition is that the four planets Mercury, Venus, Earth, and Mars stand in a straight line. It means that they have almost the same ecliptic longitude, which is called a conjunction or syzygy. The first line of numbers in the table contains the information that is also shown in a short program output. The additional lines provide the orbital elements for all eight planets. For more details, see sections 3.3 and 3.4. (This conjunction seems to be an important event with respect to the Giza pyramids.)

```
                         PLANETS IN A LINE (SYZYGY)
              (angular range of eclipt. longitudes dL minimized, JDE)
                                < option   0 >

 "Keplers equation",        ecliptic of date,        linear c. Mercury to Mars
 Period   (years)   3000.00  to   3200.00 (c2)      angular range:  5.0000 deg

co       k         JDE          year    dt[days]  Lm-Lv    Lm-Le  Lm-Lma    dLmin
===============================================================================
12      4519  2849066.03400   3088.376 -13.729  -3.366  -2.601   0.0      3.366
-------------------------------------------------------------------------------
pla.   mean long.   a [AU]   eccentr.  asc.node    incl.      per.    per.[AU]
-------------------------------------------------------------------------------
Mer    218.24880   0.38710   0.20585   61.26192  7.02274   94.43121   0.30741
Ven    237.78863   0.72333   0.00626   86.53534  3.40547  146.69081   0.71880
Ear    236.06015   1.00000   0.01624     ---     0.00000  121.70696   0.98376
Mar    244.75076   1.52368   0.09438   57.96608  1.84469  356.11360   1.37988
Jup    319.97784   5.20261   0.05021  111.62426  1.24399   31.99903   4.94137
Sat     46.22049   9.55489   0.05166  123.19406  2.44653  114.53500   9.06126
Ura    312.51632  19.21845   0.04601   79.86019  0.78595  189.20812  18.33424
Nep    177.48520  30.11039   0.00906  143.80990  1.66784   63.69138  29.83766
===============================================================================
 Computed constellations:      1052
 Number of syzygies     :         1       CPU-time  0: 0: 0.008 -- end of run.
```

The 15 quick start options, representing typical program runs, are specified in more detail. The mode, in which the parameters are defined individually one after the other, can be entered with the option “0.” A detailed description of all corresponding menus is provided in section 3.3. It is anticipated here, that there are many more quick start options than 15. The additional quick start options, having three digits, are intended to reproduce the results in the tables of the two books [5, 13].

For a better understanding, we must mention that in Ref. [5] 14 different dates and associated planetary constellations within the period 13,000 BC to 17,000 AD were analyzed in detail, depending on the geometrical approach, when comparing pyramid positions with the positions of the planets. These constellations were numbered 1 to 14. Five of them were more significant than the others, but later it became clear that the constellation with the number 12 is the most important one [13].

3.1.1 Pyramid positions

One of the main results of the first book [5] is that the three inner planets correlate with the three pyramids of Giza. More precisely, the Cheops Pyramid represents the planet Earth, the Chefren Pyramid represents Venus, and the Mykerinos Pyramid represents Mercury. The book describes how the correlation between pyramids and planets was discovered. Three basic equations were found that define the sizes of the pyramids. The maximum relative error of these equations is 0.2 %. From recent systematic studies, it came out that the relative uncertainty of the first equation is approximately 0.001 % [13] ! With S being the base length of the pyramid, V the volume, Q the aphelion distance (largest distance to the Sun), and c the speed of light, these equations are as follows:

$$\frac{S_{Cheops}}{c \cdot 1\,\mathrm{s}} = \frac{V_{Earth}}{V_{Sun}} \quad , \quad \frac{V_{Cheops}}{V_{Chefren}} = \frac{V_{Earth}}{V_{Venus}} \quad , \quad \frac{S_{Cheops}}{S_{Mykerinos}} = \frac{Q_{Earth}}{Q_{Mercury}} \qquad (1)-(3)$$

(Cheops Pyramid) (Chefren Pyramid) (Mykerinos Pyramid)

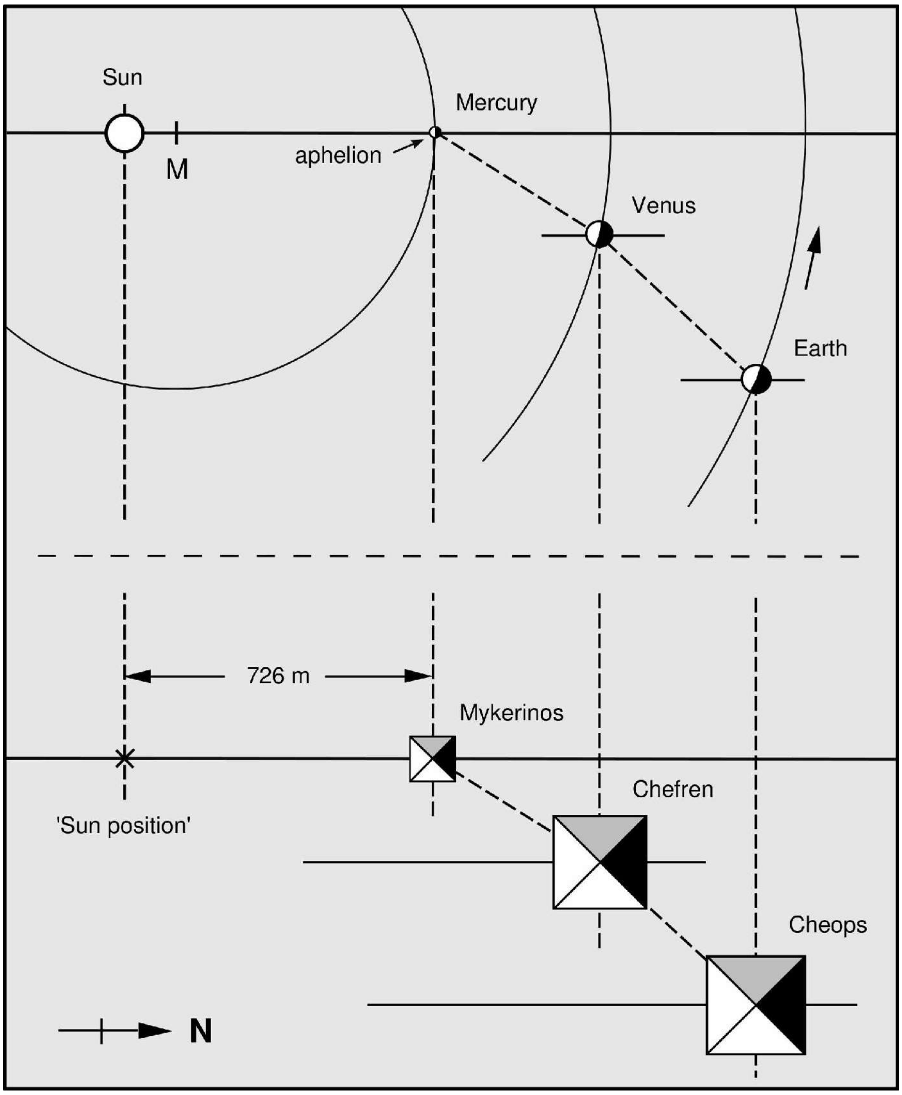


Figure 4: *Correlation between the inner three planets of our solar system and the three pyramids of Giza. The positions are each projected vertically into the main plane. Mercury is placed exactly at the aphelion. For better visibility, the Sun is magnified by a factor 6 in relation to the planetary orbits, and the planets by a factor 500 [5].*

Option 1: `3D Mer. at aph. (1)` (Compare with main menu at beginning of chapter 3.)

"3D" means 3-dimensional calculation: The comparison of the positions of pyramids and planets is performed by considering all 3 dimensions. The vertical position of a pyramid is given here by its center of mass, which is located at a quarter of the pyramid's height. The date is restricted in the way that Mercury is always placed at the aphelion of the orbit, having the largest distance to the Sun. The investigated time period are the years 13,000 BC to 17,000 AD. The results of the VSOP87 theory become less precise, if the date proceeds thousands of years into the past or into the future. Nevertheless, an estimate of the accuracy [2] (see also section 4.2.4) shows that within the given years it is by far sufficient for our purpose. In the P4 program, the dates for the application of VSOP87 are mostly restricted to the described time period. It means that start and end dates can be chosen freely within that period, but cannot exceed those limits.

The detected dates are listed in a table, where every date is represented by one line. Special dates are marked at the beginning of the line with the number of the corresponding constellation. These numbers, 1 to 14, indicate certain planetary constellations, which are defined and described primarily in [5]. The output table using this option is also given in [5, p. 346, upper part of Table 50]. For more details, see sections 3.3 and 3.4.2.

Option 2: `2D Mer. at aph. (2)`

This calculation is similar to that of option 1 with the distinction that the calculation is restricted to 2 dimensions. This means that the positions of the pyramids are projected into the horizontal plane of the Earth's surface. Accordingly, the positions of the planets are projected into the main plane, given by the plane of the Earth's orbit, respectively. Therefore, the vertical coordinates are not taken into account (see also [5, Table 45 on p. 327]).

Option 3: `const. 12, 3088 (3)`

This option calculates all relevant quantities for the constellation 12. This planetary constellation at May 31, 3088, 6:19:09 a.m. (TT, terrestrial time) represents the most relevant event out of the 14 constellations concerning the pyramid positions in Giza. Mercury is placed again at the aphelion. Notice that in the Eq. (3) (above Fig. 4), the aphelion distance $Q_{Mercury}$ appears. Additionally, the heliocentric coordinates of all planets from Mercury to Neptune for this special date are transformed to coordinates at the Giza plateau (see also [13, Table 26 in appendix A2]). The program output is given in section 3.4.3 (see also [13, chapter 4]).

Option 4: `1.5 days, 3088 (4)`

In this case a time scan around the date of constellation 12 (pyramid positions) is created. The positions of the planets are given in time steps of one hour beginning 18 hours before and ending 18 hours after the date of constellation 12 (therefore "`1.5 days`"). So, the slow change of all important parameters can be followed easily when times pass through the main moment. Compare with [13, Table 24 in app. A1] and see also section 3.4.4.

Option 5: `near aphelion (5)`

This search for planetary constellations represents the pyramid positions in Giza without the restriction that Mercury is placed at the aphelion. It came out that the planets Mercury, Venus, and Earth are in line with the pyramid positions only when Mercury is placed not too far away from its aphelion position. So, in order to keep the computation time short, the constellations are firstly checked with Mercury in the aphelion. If the agreement of the positions is good enough, Mercury is placed outside (but near) the aphelion position (short version VSOP87). At the beginning of each line, "F" means relative error ≤ 0.5 %; "M" means error of scale factor ≤ 2 %; and ">>>" means both errors ≤ 0.1 %. The errors and especially the theoretical scale factor "M" are described in [5].

3.1.2 Chamber positions

Interestingly, 44 days before the “pyramid date” of constellation 12, the planets Mercury, Venus, and Earth represent the arrangement of the chambers in the Great Pyramid. At this moment, Mercury is placed exactly at the perihelion of its orbit, the nearest point to the Sun. The correlation between planets and chambers can be seen in Fig. 5. Notice that the “chamber constellation” also defines a “Mars position” within the Great Pyramid above the King's chamber. Additionally, the “Sun position” could be the place of another (secret) chamber. For detailed information and exact coordinates of the new locations, see section 3.4.8. Between the two dates of chamber and pyramid positions, the five celestial bodies (Sun, Mercury, Venus, Earth, and Mars) are placed nearly in a straight line. This “linear constellation” (syzygy) is examined in section 3.1.3.

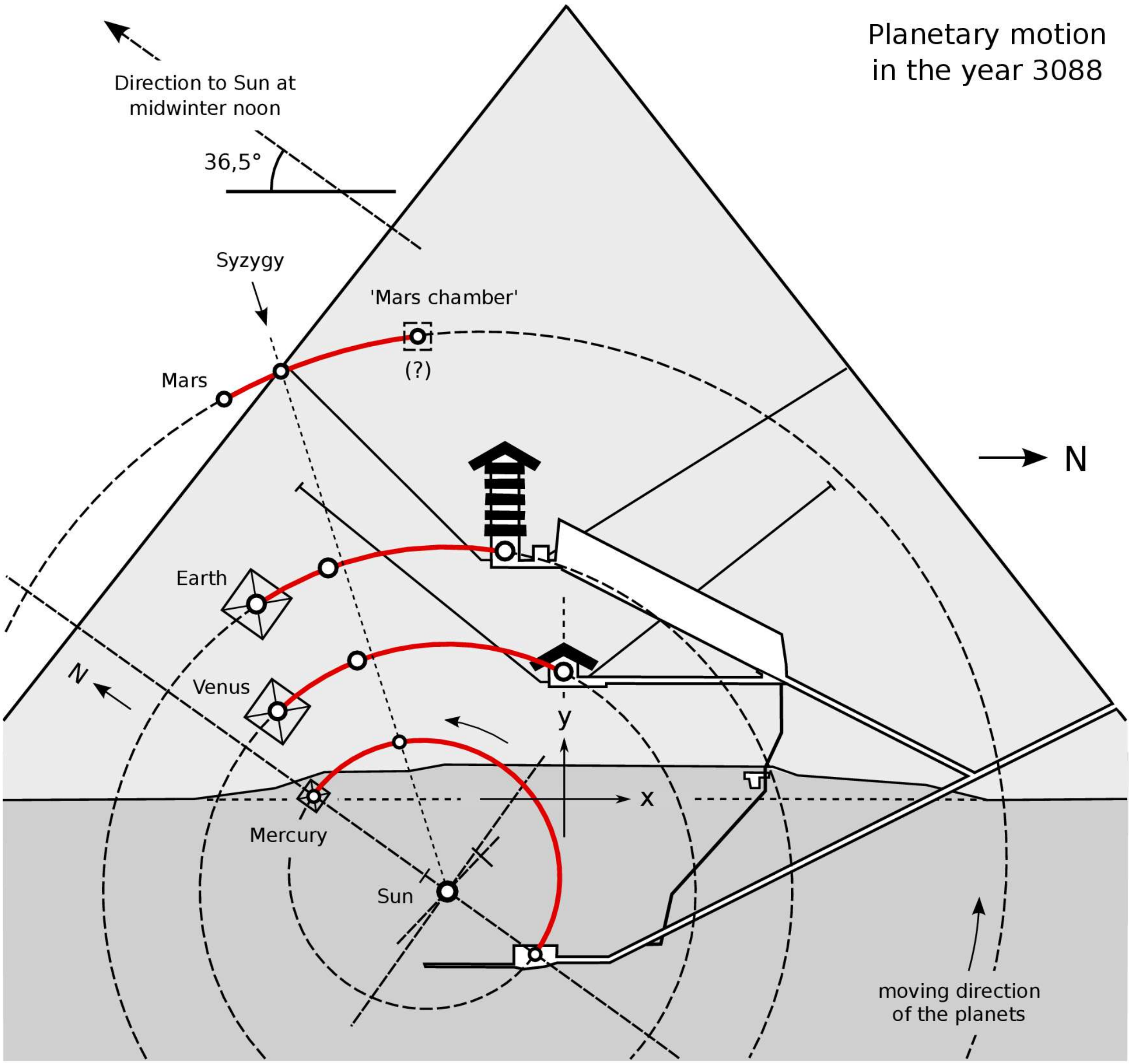


Figure 5: *Cross-sectional area of the Great Pyramid (Cheops Pyramid) as seen from the east [13]. Common representation of the known original chamber system in the pyramid, the arrangement of the pyramids themselves, and the planetary orbits. The time span between the constellations of chamber and pyramid positions is 44 days (red paths), which is half of Mercury's orbit. On the right side of the “Sun” Mercury is placed at perihelion, on the left side at aphelion. The whole figure corresponds to constellation 12. All “planetary positions” can be calculated with P4 using the option 380. The configurations in the figure are roughly true to scale. For the exact positions, use the calculated coordinates.*

The origin of the coordinate system is placed at the vertical middle axis of the east wall of the Queen's chamber on the level of the pyramid base. The x-axis points to the north, the y-axis points upward (compare with Fig. 5), and the z-axis points to the east. The quick start options, which have been implemented as main examples, are the following:

Option 6: `3D Mer. at per. (6)`

The calculation is analog to option 1. The positions of the pyramids are replaced by the positions of the chambers in the Great Pyramid, and Mercury is always located at its perihelion. The investigated time period lasts again from 13,000 BC to 17,000 AD.

Option 7: `Keplers equ.    (7)`

Here, the planetary positions are not calculated with the short or the full version of VSOP87. Instead, the positions are determined with the orbital elements by solving Kepler's equation (section 4.2.3). The orbital elements are derived from the VSOP82 theory [17, pp. 197 ff.], and the transcendental equation of Kepler is solved numerically. The other boundary conditions are similar to option 6. This method does not have the accuracy of the full version of VSOP87, but it has the advantage that the calculation is rather fast. When the same time period of 30,000 years is investigated, 124,558 constellations are calculated and checked, and the overall computation time is less than 1 second. Moreover, it is a good test of the other results (see also section 3.4.5).

Option 8: `const. 12, 3088 (8)`

This computation is analog to option 3. Only the positions of the pyramids are replaced by the positions of the chambers in the Great Pyramid, and Mercury is placed at its perihelion (section 3.4.8). The exact date is April 17, 3088, 6:41:13 a.m. (TT, terrestrial time). Now, the planetary positions are all transformed to the coordinate system of the Great Pyramid. With a deviation of 4.2°, the transformed ecliptic plane (plane of the orbits) is almost parallel to the central vertical plane in the pyramid, oriented in north–south direction (x-y-plane). The corresponding origin of this coordinate system can be seen in Fig. 5 on the ground level of the pyramid, as described previously. The z-axis, which is not shown, points perpendicularly out of the drawing plane. From this calculation it was determined that the “Mars position” is also placed inside the Great Pyramid about 40 meters above the King's chamber (see Fig. 5 and [13, section 4.5, Tab. 25 in app. A2]).

Option 9: `1.5 days, 3088  (9)`

Analogously to option 4, this is a 36-hour time scan around the “date of the chambers.” This constellation also got the number 12 because it is closely related to the “date of the pyramids.” The time difference of 44 days is very short, compared to astronomical scales.

Option 10: `F minimized    (10)`

The results here are similar to those of option 5, but the algorithm is more sophisticated. For the time period from the year 3500 BC to the year 6500 AD, the planetary positions are compared with the chamber positions and the date is not restricted in any way. This means that Mercury can be placed anywhere on its orbit. For each date, where the positions match with each other and the found relative error is below a certain value (0.25 %), this error is minimized and the constellation is counted only, if the minimized error is smaller than another limit (0.05 %). The result is 38 dates within the investigated 10,000 years when these conditions are met. Of course, other boundary conditions imply that ultimately only one date is left (see option 8, and also [13, Tab. 20 in app. A2]).

3.1.3 Planetary conjunctions and transits

Conjunction means either that two or more celestial bodies have almost the same position in the sky, or that, for example, two or more planets have the same ecliptic longitude. The latter case can be seen in Fig. 6. The figure shows the correct dimensions of the orbits in 3088, when the four planets Mercury, Venus, Earth, and Mars together with the Sun are aligned nearly in a straight line. As mentioned before, such an arrangement is called “syzygy,” being a generic term of “conjunction” and “opposition.”

From time to time, Mercury and Venus pass in front of the Sun's disc, which is called a transit. In Fig. 7 the typical lapse of time is shown for the Venus transit in the year 2012. Here we take Venus instead of Mercury because of the recent Venus transit, which was a rare event.

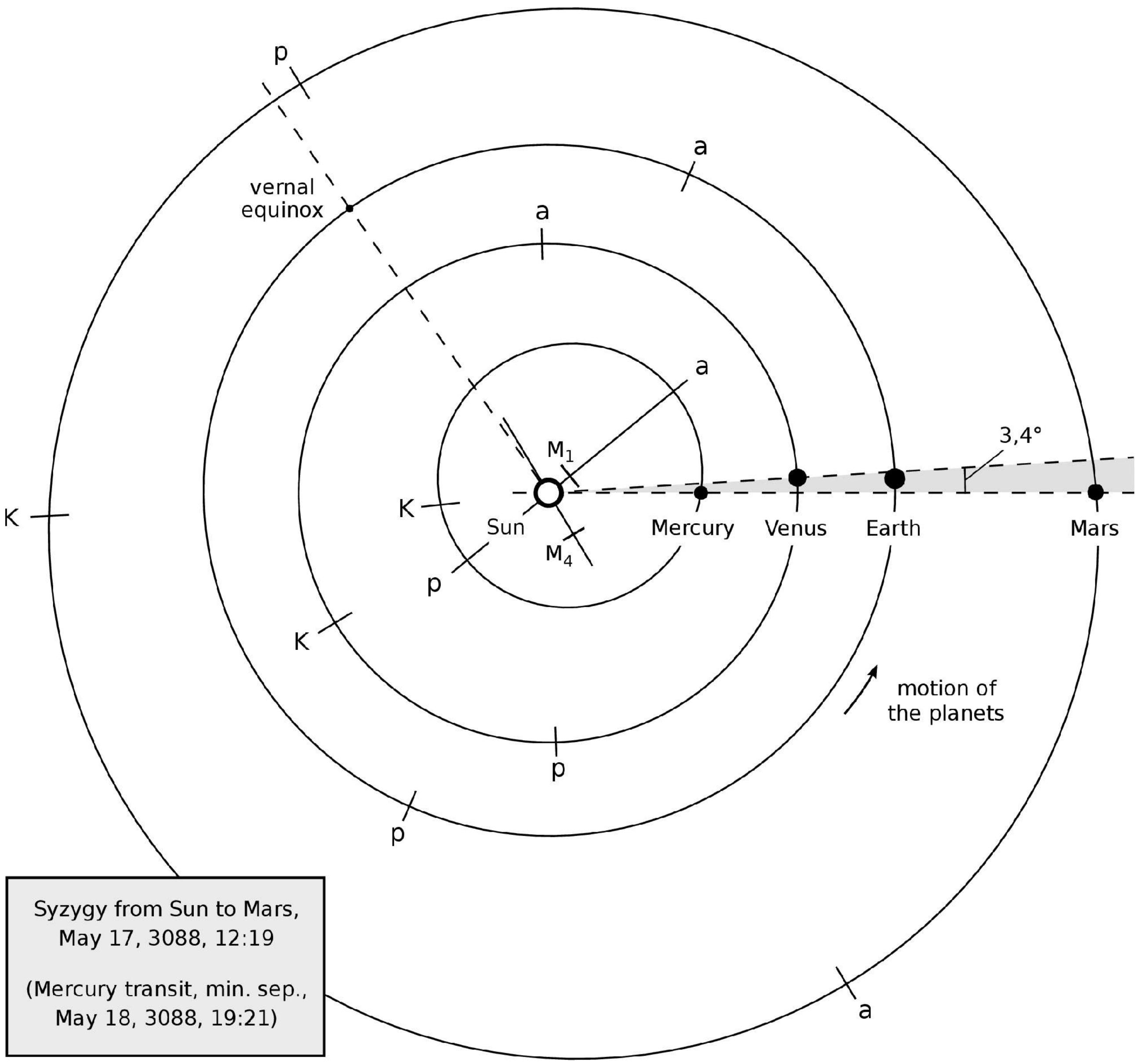


Figure 6: *Approximate true-to-scale representation of the orbits of Mercury, Venus, Earth, and Mars around the Sun. On May 17, 3088, the four planets and the Sun are positioned nearly in a straight line (syzygy), followed by a Mercury transit. The places p, a, and K represent perihelion, aphelion and ascending node, and M_1 and M_4 are the orbital centers of Mercury and Mars. For better visibility, the planets and the Sun are drawn bigger than they would normally look like, if they were drawn to scale. The dates are given in TT (terrestrial time). The range of ecliptic longitudes dL (or ΔL) is only 3.4°.*

The interesting point is that the two events "the 4 planets and Sun in a straight line" and a "transit of Mercury or Venus" normally do not take place simultaneously. Within the given 30,000 years (from 13,000 BC to 17,000 AD), this happens only six times, if we fix the maximum angular range of the ecliptic longitudes to 5°. This means that the coincidence of the given syzygy and a transit of Mercury or Venus happens only an average of every 5,000 years. This happens exactly between the two dates of the "chamber constellation" and the "pyramid constellation," which are separated by 44 days! Figure 5 shows that within this period of time the four planets and the Sun form nearly a straight line. About one day later, Mercury passes the solar disc (for more details see [13]).

Option 11: `Mercury tr.   (11)`

The contact dates of all Mercury transits are calculated for the years 2950 to 3200. Additionally, the minimum separation between Mercury and Sun, the case of ascending or descending node, and the serial number are given. For the transit series of Mercury and Venus see also section 4.7.4 and [22, pp. 7–13]. The period includes the year 3088, which is labeled automatically with the number 12. In the given time span, 34 Mercury transits are registered (section 3.4.6).

Option 12: `Venus tr.     (12)`

All Venus transits with their four contact points (phases) and minimum separation are listed for the years 1500 to 4000. This time period is larger than for Mercury because Venus transits occur less frequently than Mercury transits. Between the years 1500 and 3000, and with a period of roughly 120 years, two Venus transits occur, following each other with a time difference of 8 years. (The time limits of this option are chosen in such a way that all results are displayed on one monitor screen.) If we do not have a full transit but a grazing transit, the corresponding line gets a "v" at the beginning. (See also the first program output in section 3.1.) The same is true for a grazing Mercury transit, but instead the line gets an "m". In the previous time period of option 11, there are by chance no grazing transits of Mercury.

Option 13: `syzygy, 3 pl. (13)`

This option yields linear constellations (syzygy) of the three planets, Mercury, Venus, and Earth, together with the Sun. The condition for the syzygy is that the ecliptic longitudes of all three planets match within an angular range of $dL = 5°$. The investigated time period is 2900 to 3300. If a transit of Mercury or Venus also occurs during the syzygy event (within a few hours or a few days), the beginning of the line in the table gets an "M" or a "V" for a full transit of Mercury or Venus or an "m" or "v" for a corresponding grazing transit. It might also happen during such a "linear constellation" that both a Mercury and a Venus transit occur, so that the line is indicated with both letters like, for example, "MV". This happens only three times between the years 13,000 BC and 17,000 AD, assuming all ecliptic longitudes within a range of 5°. Note that this does not mean a simultaneous transit, because both transits might have a time difference of a few hours or a few days. If a syzygy is near a known constellation within a certain time limit (10 orbital periods of Mercury ≈ 880 days), the corresponding line is marked with a small arrow "->" [13, Tables 27, 28]. However, for transits in the remote future or remote past, the precision of VSOP87 has to be considered (section 4.2.4).

Option 14: `syzygy, 4 pl. (14)`

Now, Mars is also included. This means that the program searches for "linear constellations" of Mercury, Venus, Earth, Mars, and the Sun. The condition is that the ecliptic longitudes of all four planets are again placed within the 5° angle, meaning a fourfold planetary conjunction (syzygy). This happens very rarely. So, the whole time period from 13,000 BC to 17,000 AD is checked. The coincidence of the given syzygy together with a transit occurs only six times, which means an average of every 5,000 years (see section 3.4.7).

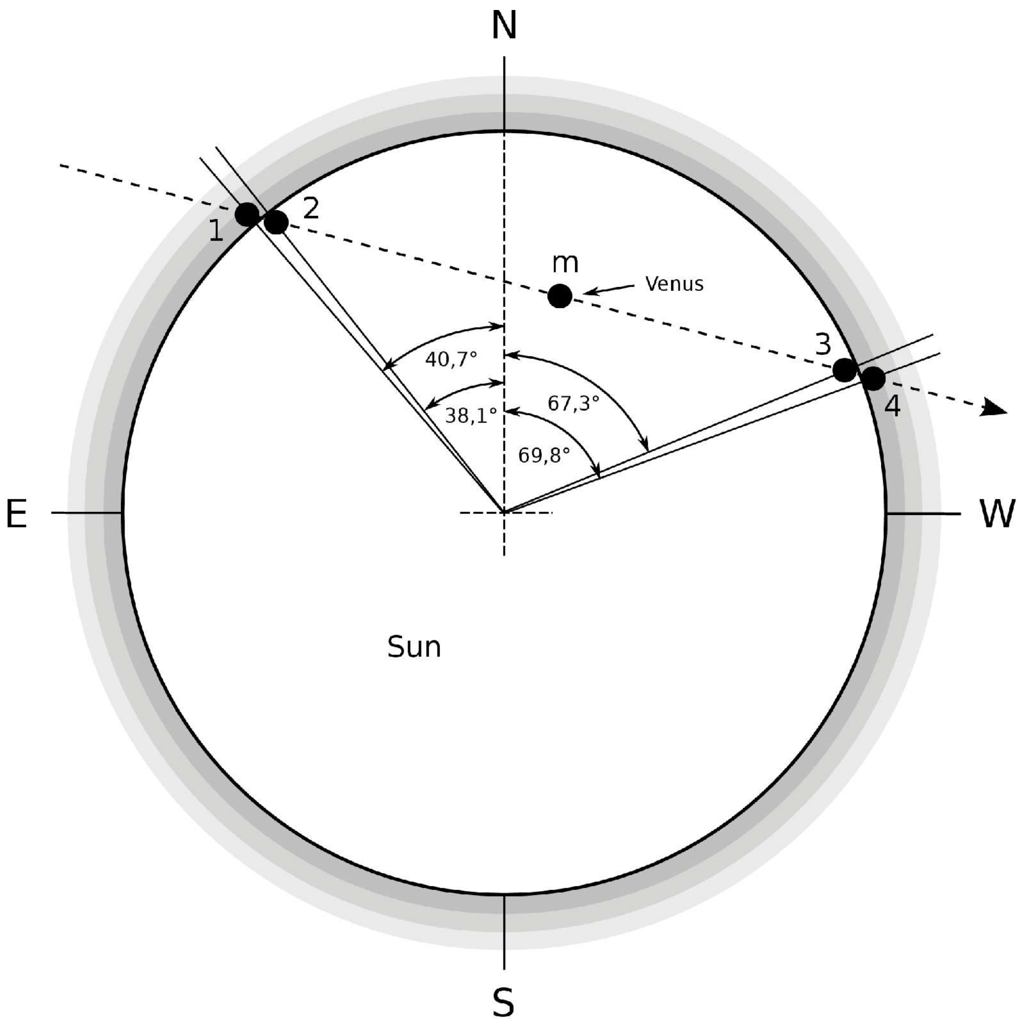


Figure 7: *Venus transit on June 5−6, 2012, as seen from the center of the Earth (geocentric phases). The positions 1 to 4 are the geocentric contact points (phases) and "m" represents the place of minimum separation between Venus and the center of the Sun. The size of Venus and the Sun are drawn to scale as seen from the Earth. N shows the north direction on the celestial sphere. The direction from E (east) to W (west) is the direction of the apparent motion of the Sun due to the rotation of the Earth. The angles of the contact points were calculated with P4 (compare with Meeus [22, p. 48]).*

Option 15: `TYMT-test (15)`

This option is mainly a test to check the processing speed. "TYMT" means "Ten thousend Years Mercury Transit." The transits of Mercury (geocentric phases) are calculated for the years 3000 BC to 7000 AD. Using an Intel Core i5-3210M processor (2.5 GHz, 8 Gbytes, dual channel), the TYMT-test needs 46.0 seconds. During the 10,000 years, 31,520 passages of Mercury along the Sun are tested and 1,340 transits are found. The results are calculated with the full version of VSOP87. Interestingly, on the used 64-bit system the 32-bit version of P4 runs faster (43.2 s) than the 64-bit version. Even higher speed can be obtained, e.g., by parallelizing the processing and using multiple threads. A corresponding program version already exists (see footnote in the appendix on page 82). About 20 years ago, without optimization of the software and using the computer hardware at that time, the TYMT-test would have needed about 1 month of computation time.

The CPU-time directly depends on the speed of the processor. More "GHz" means less CPU-time. A criterion, which is more or less independent of the clock frequency and a better measure of the software efficiency, would be the product of frequency and CPU-time: 2.5 GHz · 43.2 s = 108 GHz·s. This is just a number and means the number of clock cycles necessary for the whole computation. We can call it 108 Gc (gigacycles) which is $108 \cdot 10^9$ cycles. In P4, only the CPU-time is provided.

3.2 Quick start options for the book tables

Most of the astronomical tables in the two books [5] and [13] can be reproduced by additional quick start options, called “book options.” These options are not shown in the main menu, but they can be found easily. All book options have three digits. The first two digits represent the number of the table and the last digit indicates the section of the table. For example, Table 39 in the first book [5, p. 319] consists of three parts, one placed above the other. These parts can be reproduced by the options 390, 391, and 392, meaning the digits 39 plus one digit 0, 1, or 2 for the different parts. If a table has only one part, like Table 45 [5, p. 327], a zero has to be appended and the corresponding option is 450.

3.2.1 Book 1 “Pyramiden und Planeten”

If the Tables 39 to 51 in [5] are reproduced with the P4 program the program output is not always identical to the tables. In some cases, the program output is much larger, which means that in the book only the important quantities are printed.

In Table 50, the correlation between pyramids and planets is checked, in which Mercury is fixed to the aphelion of the orbit and the “Sun position” in the pyramid area is free in all 3 dimensions. The latter aspect is described in more detail in sections 3.3.10–3.3.12 and 4.6. There are three possibilities to define the vertical position of a pyramid: It can be the center of mass of the pyramid, the middle of the pyramid base, or the top of the pyramid. The first two cases are presented in Table 50 and can be calculated with the options 500 and 501. The constellations with the pyramid top as the vertical coordinate had been omitted in the table because there were no significant new results. Nevertheless, this case can be computed by using the option 502.

In Table 51, the correlation between pyramids and planets is again investigated. Not only is the “Sun position” on the Giza plateau free in all 3 dimensions, but the date is also free, which means that the dates are not restricted to the aphelion passages of Mercury. The first book searched for the constellations with the short version of VSOP87, and the relative error F''_{pos} was minimized by repeatedly starting the full VSOP87 version by hand using the P3 program. In P4, these results are calculated automatically with a fit-subroutine and the VSOP87 full version (see also section 3.4.10). Here, the results sometimes differ in the last digit from those in the book [5] because in [5] the relative error was minimized by adapting the point of time manually. The search routine from quick start option (5) uses the short version of VSOP87. Later, this routine was also implemented for the full version of VSOP87. If the reader wants to check the results in Table 51 with this different search method, it can be done with the options 517, 518, and 519 (see also section 3.3.16).

3.2.2 Book 2 (in preparation)

Tables 17–33 and 35–38 in Ref. [13] can be reproduced using the corresponding quick start options as described previously. Table 20, for instance, indicates quick start option 200. The tables in book 1 (numbers 39–51) and those of book 2 (numbers 17–38) have no overlap. Thus, the options can be used without addressing explicitly book 1 or book 2.

3.2.3 Special test option (999)

Let us assume that a special parameter setting is used and several runs have to be done by changing only one parameter. Then it is not convenient to set all other parameters each time by hand, as described in section 3.3. Instead, it is easier to use the input-output-file “inedit.t.” If the reader opens this file using an editor, they will see two sections: 1 and 2. An example of the content of inedit.t is provided below. Section 1 (big arrow) is read by the P4 program, if the quick start option 999 is used, and can be edited by hand. CAUTION! Not all combinations of parameters are possible and these parameters are not checked by the program when using option 999. The underlined parameters (see below) can be changed within their allowed values without any problem (section 3.3). For other parameters, their modes of operation and interdependencies must be known.

Section 2 is always overwritten with the presently used parameter values when the program is started with options other than “999”. Therefore, it is possible to compare the parameters of a current run (section 2) with the parameters in section 1. In the lines above section 1, the text should not be changed or deleted because for reading the parameters by the program, the number of lines must always be the same, and otherwise the original text may be lost.

Example of the content of inedit.t

```
-----------------------------------------------------------
(      User input and current input from program P4       )
-----------------------------------------------------------
(    The input data in field 1) can be edited by the      )
(    user and are read by the program with the option     )
(    "999". The input data in field 2) are written by     )
(    the program at each run and can be used for com-     )
(    parison. The manual input by the user in field 1)    )
(    allows for the creation of input data to be copied   )
(    into the file "inparm.t." Number and position of     )
(    the lines in this file must not be changed!          )
-----------------------------------------------------------
 Parameter names of the values further below
-----------------------------------------------------------
      XXXXX          ipla   ilin    imod   imo4   ikomb
      XXXXX          lv     ivers   itran  isep   iuniv
      XXXXX          ical   ika     iaph   iamax  step
  XXXXXXXXXXXXX      ison   ihi     irb    ijd
    XXXXXXXXX        zmin   zmax    ak     zjde1
      XXXXX          dwi    dwikomb dwi2   dwi3
        X            nurtr  iek     io     iout
-----------------------------------------------------------
1)  Input to edit (999) - CAUTION: No check of parameters!
===========================================================
 3  1  1  0  1
 1  3  1  3  1
 2  0  1     0   0.00000
 5  0  1  15
  1900.00000   2100.00000      0.00000        0.00000
  0.000    0.000    0.000    0.000
 1  1  1  2
===========================================================
2)  Last used input (all options except 999)
-----------------------------------------------------------
 2  4  3  0  0
 1  3  1  1  1
 2  1  2     0  24.00000
 1  0  1  15
-13000.00000  17000.00000      0.00000        0.00000
  1.850    0.000    0.000    0.000
 1  3  1  2
-----------------------------------------------------------
*************************** END ***************************
```

When using the option “999” the parameters from field 1 are taken by the program. The parameters in fields 1 and 2, provided here, are arbitrary. The parameter names beside the big arrow correspond to the numbers in fields 1 and 2. If the functionality of a given parameter is not known, see section 3.3. If the information in this section is not enough, the Fortran source code p4.f95, listed in the appendix, provides more information. Unfortunately, most comments are in German.

Note: The parameters in inedit.t are not checked with respect to correct input. So, the user should follow the hints, given above.

3.3 Detailed options (0)

In contrast to the quick start options, the single parameters in the program can also be set individually one after the other, each given by its own menu. In the main menu on page 7 (last line) we find `detailed options (0)`. So, the option to get into this modus is “`0`”. In a program run, not all combinations of the parameters are meaningful. Those that are not allowed are not presented. Sometimes a reduced menu is shown. If a number is typed, which is not offered in the menu, an error message appears. So, the program start with the option “`0`” is controlled by the program and protected against any false input. It follows a brief overview of menus and options.

1. Planetary positions: pyramids, chambers in Cheops Pyramid, linear constellations (syzygy).
2. Linear constellations and transits: transits of Mercury/Venus, conjunction of 3 or 4 planets.
3. VSOP theory versions: short or full VSOP version, combination of both, planetary elements.
4. VSOP coordinate systems: ecliptic of epoch (VSOP87C, VSOP87D), J2000.0 (VSOP87A).
5. Transit options: equal ecliptic longitudes, nearest separation, transit phases, position angles.
6. Calendar systems: Julian/Gregorian calendar or only Gregorian calendar.
7. Time systems: terrestrial dynamical time (JDE, TT), universal time (UT).
8. Mapping of planets and chambers: assignment of Mercury, Venus, Earth to the chambers.
9. Search method for the dates: Mercury passages at aphelion, perihelion, date not restricted.
10. “Sun position”: south of Mykerinos or Chefren Pyramid, “Sun position” free.
11. Computation of free “Sun position”: free in 2 or 3 dimensions, 3D calc. with SLE or FITEX.
12. Vertical level of pyramid positions: pyramid base, center of mass, top of pyramid.
13. The z-coordinate of chamber positions: east wall, west wall, spatial middle of each chamber.
14. Datum plane for Earth's surface: projection on plane of Earth, Mercury, or Venus orbit (2D).
15. Specification of timing: number of constellation (1–14), k-number, years, Julian Day.
16. Tolerance in degree or percent: tolerance/angular range of ecliptic longitude, relative error.
17. Syzygy with simultaneous transit: all planetary conjunctions, only with simultaneous transit.
18. Polarity (orientation of planetary orbits): view from ecliptic north, south or both options.
19. Complexity of output: normal output, extended output.
20. Mode of program output: output only on monitor, monitor + file, special output, exit.

In the following sections the menus are described one by one in the order of their appearance during program start. Each menu is given at the beginning in blue. Note: Not all menus appear at program start, depending on the kind of computation. On the right side of each menu, the corresponding internal parameter is given, like for example: “ipla”.

3.3.1 Planetary positions

```
Constell.  pyr.(1),  chamb.(2),  lin.(3)  :
```
(internal: ipla)

(1) planetary constellation of Mercury, Venus, Earth = positions of the three pyramids in Giza
(2) planetary constellation of Mercury, Venus, Earth = positions of the three chambers in Great P.
(3) “linear constellations” (syzygy, transit)

There are three main categories. The positions of the planets are compared with (1) the positions of the three pyramids in Giza, or (2) with the system of the three chambers in the Great Pyramid. Option (3) investigates when the planets build a planetary conjunction and “linear constellation,” respectively, or a Mercury or Venus transit. The origin of the coordinate system for option (1) is middle of base area of Mykerinos Pyramid, x-axis points to the north, y-axis points to the west, and z-axis points upward; for option (2), middle axis of the Queen's chamber in its east wall on the level of the pyramid base, x-axis points to the north, y-axis points upward, and z-axis points to the east.

3.3.2 Linear constellations and transits

```
Tr. Mer.(1), Ven.(2), 3-co.(3), 4-co.(4)  :
```
(internal: ilin)

(1) transits of Mercury
(2) transits of Venus
(3) triple conjunction of the three planets Mercury, Venus, and Earth
(4) fourfold conjunction of the four planets Mercury, Venus, Earth, and Mars

The “linear constellations” (above option “`lin.(3)`”) are subdivided into the four given menu points. Options (1) and (2) are clear. Option (3) means a syzygy of the planets Mercury, Venus, Earth, including the Sun, and option (4) a syzygy of Mercury, Venus, Earth, Mars, and the Sun.

When running option (3), it becomes clear that Mercury and Earth always have the same ecliptic longitude. This seems reasonable, but it is not put into the program as a boundary condition; it just comes out as a result. With option (4), three different cases are observed. Either the ecliptic longitudes of Mercury and Mars are identical, those of Mercury and Earth, or those of Venus and Mars. In principle, there are other combinations of two out of four planets, but other solutions do not exist. If one thinks about the problem, it becomes clear why. (If p is the number of planets, then the number of cases, different pairs of planets with equal longitude, is $N = (p–1)·(p–2)/2$ with $p ≥ 3$.)

3.3.3 VSOP theory versions

```
VSOP87     combi. (1), short version (2),
       Kepl. equ. (3), full version (4)  :
```
(internal: imod)

(1) combination of the short and full version of the VSOP87 theory
(2) short version of the VSOP87 theory, Meeus [17, pp. 381 ff.]
(3) planetary elements, polynomials of third degree [17, pp. 200 ff.] and solving Kepler's equation
(4) full version of the VSOP87 theory, Bretagnon and Francou [1, 2]

Option (3) (solving Kepler's equation) is the fastest algorithm, but it has the lowest accuracy. Option (2) (short VSOP87 version) is not so fast, but it has a higher precision. Option (4) (full VSOP87 version) has the highest precision, but it takes the most computation time. The option (1) (combination of short and full VSOP87 version) is fast and yields the same high precision as (4). So, the recommendation is: option (1) for larger time periods and option (4) for single constellations.

3.3.4 VSOP coordinate systems

```
System   ecl. of epoch (1),  J2000.0 (2)  :
```
(internal: lv)

(1) ecliptic of epoch (dynamical equinox)
(2) standard system J2000.0 (ecliptic of Jan. 1, 2000, 12:00, TT, or *JDE* = 2451545.0)

The two options are the two applied coordinate systems for the VSOP87 theory. The short VSOP87 version is provided only with the ecliptic of epoch (1). The VSOP87 full version and the orbital elements for solving Kepler's equation are given for both systems.

3.3.5 Transit options

```
Date  equ.L.(1), nearest (2), phases (3)
          phases and position angles (4)  :
```
(internal: isep)

(1) transit check at equal ecliptic longitudes (planet, Earth), finite speed of light not considered
(2) transit check at minimum separation (nearest approach), finite speed of light not considered
(3) geocentric transit phases, as seen from Earth
(4) geocentric transit phases and position angles (the output needs 148 characters line width)

The first two options are each calculated for a fixed moment. Thus, the travel time of light is not taken into account. These options were written during an early stage of the program development and now serve for test purposes. Option (3) yields the true geocentric transit phases 1 to 4 as well as the minimum separation by considering the finite speed of light. In option (4), the position angles on the solar disk and the semidiameters of the Sun and planet are also calculated. In addition, central transits (minimum separation < semidiameter of planet) are labeled with “C” (geocentric central transit) and “c” (central transit, seen from some place on Earth). Option (4) is the only option that needs 148 instead of 80 characters line width on the computer monitor.

3.3.6 Calendar systems

```
Calendar  Jul./Greg. (1), only Greg. (2)  :
```
(internal: ical)

(1) Automatic choice of Julian and Gregorian calendar
(2) Gregorian Calendar for all times

In the first option, the Julian calendar is used for the years 4712 BC to 1582 AD and the Gregorian calendar for all other times. It does not make sense to use the historical Julian calendar before 4712 BC because in this distant past the calendar gets more and more out of hand, and there are no historical events to apply this calendar. In contrast, the Gregorian calendar in these past times is in much better agreement with the seasons. Option (2) means that only the Gregorian calendar is used for all times. The calendar menu is presented also when no calendar dates are calculated. This has to do with the fact that the decimal year, displayed in all outputs, is slightly different for both calendars.

3.3.7 Time systems

```
Time system         JDE/ TT (1),  UT (2)  :
```
(internal: iuniv)

(1) JDE (Julian Ephemeris Day, equal to JD) and TT (terrestrial time), respectively
(2) UT (universal time)

JDE and TT are identical time scales with a constant length of days. UT takes into account the deceleration of the Earth's rotation due to tide friction, so that from time to time a leap second is introduced (in UTC). Because the slowing down of the Earth's rotation cannot be predicted precisely, the terrestrial time (TT) is the accurate measure. With option (2), TT can be transferred to UT by using the equations for ΔT = TT – UT of F. Espenak and J. Meeus [23, 24] (see section 4.8).

3.3.8 Mapping of planets and chambers

```
Planets  E-V-M (1), E-M-V (2), V-E-M (3),
         V-M-E (4), M-E-V (5), M-V-E (6)  :
```
(internal: ika)

(1) Earth – Venus – Mercury
(2) Earth – Mercury – Venus
(3) Venus – Earth – Mercury
(4) Venus – Mercury – Earth
(5) Mercury – Earth – Venus
(6) Mercury – Venus – Earth

The three planets each correspond in the given sequence to the King's chamber, the Queen's chamber, and the subterranean chamber (rock chamber) in the Cheops Pyramid. Option (1) is the case that actually makes sense. The other options are added as a test and for the sake of completeness.

3.3.9 Search method for the dates

```
Passage aph./per. area of aph./per. free
        (1)  (2)          (3)  (4)   (5)  :
```
(internal: iaph)

For options (3) and (4) it follows

```
Steps per Mercury passage :
Step width  (hours, real) :
```
(internal: iamax)
(internal: step)

(1) date: Mercury passage at aphelion
(2) date: Mercury passage at perihelion
(3) time interval around aphelion passage of Mercury
(4) time interval around perihelion passage of Mercury
(5) date completely free (within a given epoch)

Note: There are similar input menus in which not all of the options (1) to (5) are given, but the meaning of the numbers is always the same. In option (1), only the dates are tested, when Mercury passes the aphelion. Option (2) means the same for perihelion. In options (3) and (4), those constellations are tested that are in a given time interval around the aphelion and perihelion passage of Mercury, respectively. This could be, for example, an interval starting 7 days before and ending 7 days after each aphelion passage with equal time steps of (for instance) 12 hours. In this case, it means that $14 \cdot 2 + 1 = 29$ dates are checked for each aphelion passage. In option (5), the date is totally free. So, during the search the time increases in automatically chosen time steps, and if a promising constellation is found, the relative error is minimized by an automated fit procedure.

For the options (3) and (4), two additional input lines (see above) ask for a specific time interval for each aphelion or perihelion passage. First, it requires the number of steps per Mercury passage; and secondly, the step width in hours is required. In the given example, the number of steps would be 28 and the step width would be 12 (hours). When searching for “linear constellations” (syzygies) with the short VSOP87 version or the “planetary elements” version (Kepler's equation), the following input line allows for a search with fixed time steps:

```
Step width [hrs] (min.-search 0.) (real)  :
```
(internal: step)

Thus, in the case that the overall interval d*L* of ecliptic longitudes of the corresponding planets falls below a certain limit (e.g., below 5°) the program calculates all of the following constellations in the given steps (e.g., in "1 hour" steps) until d*L* again exceeds the previously given limit. If the input 0.0 is given as the step width, the time steps are automatically chosen, and if d*L* decreases below the given limit (like 5°), the date is optimized by minimizing d*L*. This case of automatically minimizing d*L* is always used in the combined search with the short and full VSOP87 version.

3.3.10 "Sun position"

```
Sun pos.  Myk.(1),  Chefr.(2),  free (3)  :
```
(internal: ison)

(1) "Sun position" fixed 726 m south of the center of the Mykerinos Pyramid
(2) "Sun position" fixed 963 m south of the center of the Chefren Pyramid
(3) "Sun position" free

In options (1) and (2), the "Sun position" south of Mykerinos Pyramid and south of Chefren Pyramid means that the "Sun position" is placed exactly on the north−south middle axis of the corresponding pyramid. The given distances were determined geometrically on the basis of figures like Fig. 1 or 8 (yielding the angles δ_1 and δ_2 in Fig. 8).

In option (3), the "Sun position" at the Giza plateau is not fixed, and the calculation of it has to be further specified by the following menu. Note that "not fixed" does not mean "not defined." The "Sun position" at the Giza plateau is defined exactly by the positions of the three planets Mercury, Venus, and Earth, when considering all 3 dimensions mathematically.

3.3.11 Computation of free "Sun position"

```
Sun     2D (1), 3D/SLE (2), 3D/FITEX (3)  :
```
(internal: ison2)

(1) "Sun position" free in the 2 horizontal dimensions (meaning "restricted to the Earth's surface")
(2) "Sun position" free in 3 dimensions, calculation with a System of Linear Equations (SLE)
(3) "Sun position" free in 3 dimensions, calculation with coordinate transformation and FITEX

When the planetary and pyramid constellations are adapted to each other by comparing the coordinates or by coordinate transformation, the "Sun position" can be predefined or not. "Predefined" means that it is restricted in the vertical dimension. This option (1) was applied mainly for the constellations 1 to 11. Options (2) and (3) are two different ways of calculating the "Sun position" in 3 dimensions on the Giza plateau south of the pyramids and also in the Great Pyramid (for details, see section 4.6 and [5, app. A16]). In the case of a rather small relative error between pyramid and planet configuration, both mathematical methods yield the same result and the same "Sun position," respectively. In the case of the chambers, the option "`2D (1)`" does not exist.

3.3.12 Vertical coordinate of pyramid positions

```
z-coord.    base (1),   C-M (2),   top (3)  :
```
(internal: ihi)

(1) z-coordinate at level of pyramid base
(2) z-coordinate at level of center of mass of the pyramid
(3) z-coordinate at top of pyramid

When fixing the pyramid positions in 3 dimensions, it is necessary to determine the height level of the positions. Three alternatives are given. The center of mass of a pyramid (option 2) can be shown to be located at a quarter of the pyramid height. Options (1) and (2) create mostly the same or similar results, whereas option (3) also generates other constellations.

3.3.13 The z-coordinate of chamber positions

`Wall      east (1), middle (2), west (3)  :` (internal: ihi)

(1) center of east wall of each chamber
(2) spatial middle of each chamber
(3) center of west wall of each chamber

When fixing the chamber positions in 3 dimensions, it is necessary to fix the east−west location (z-coordinate) of the positions. Because only the east walls of all three chambers are located in the same vertical plane, but not the west walls, three alternatives are also given here.

3.3.14 Datum plane for Earth's surface

`Coord.   ecl.(1),   Mer.(2-4),   Ven.(5)   :` (internal: irb)

(1) projection plane is ecliptic plane (plane of Earth orbit)
(2−4) projection plane is plane of Mercury orbit
(5) projection plane is plane of Venus orbit

For the 2-dimensional calculation (section 3.3.11, option (1)), the planetary positions are projected vertically into one plane, as also the pyramid positions are projected vertically onto the Earth's surface. Three different planes can be tested, defined by the orbits of Earth, Mercury, and Venus, respectively. The change from the Earth's orbit (heliocentric coordinate system, VSOP87C) to a system based on the Mercury or Venus orbit is performed with rotational matrices (see section 4.5 and [5, app. A15, pp. 328 ff.]). For Mercury, three different combinations of matrices are available (options 2–4), all yielding the same result. They were used for test purposes during the development of the program.

3.3.15 Specification of timing

`Constell. (1..14),  k-No. (15),  JDE (0)  :` (internal: ijd) or
`Constell. (1..14),  years (15),  JDE (0)  :` (internal: ijd)

(1–14) dates of the constellations 1 to 14 as given in [5, p. 315, Tab. 38]
(15) *k*-number (integer number of Mercury passages through aphelion or perihelion) or
(15) time period in years, specified in the menu lines following below
(0) JDE (Julian Ephemeris Day or Julian Day)

The numbers 1 to 14 belong to the planetary constellations 1 to 14 with the given dates. With these options, only one constellation is calculated. The *k*-number in option (15) counts the passages of Mercury through its aphelion or perihelion (see section 3.3.9, options (1) and (2)). The numbers start with zero after the beginning of the year 2000. Before that date, the numbers are negative. The format of the *k*-number is “real” (number with decimal point). Normally, the number (not the format) is an integer (the digits after the decimal point are zero), but it does not need to be an integer.

The latter case means that the date is not the passage through aphelion or perihelion, but somewhere between both moments. The option "`years (15)`" allows us to check the dates in a given time period. Here, the result normally consists of several planetary constellations. The last option "`JDE (0)`" enables us to specify directly a JDE number so that the constellation of this moment is calculated.

```
      k (real):                    (internal: ak)
or    from  year (real):           (internal: zmin)
      until year (real):           (internal: zmax)
or    JDE (real):                  (internal: zjde1)
```

The Julian Ephemeris Day can be any date, just like the *k*-number, which implies that the relative error between alignment of pyramids (chambers) and planets can be very large.

3.3.16 Tolerance in degree or percent

```
Tolerance ecl. long. Venus, Earth (real)  :   (internal: dwi)       or
Max. F-pos at aphelion/ per.  (real) [%]  :   (internal: dwi)       or
Tolerance ecl. long. VSOP short   (real)  :   (internal: dwi)
    "      "     "   VSOP full    (real)  :   (internal: dwikomb)   or
Max. F-pos    VSOP short ver. (real) [%]  :   (internal: dwi)
 "     "      VSOP  full ver. (real) [%]  :   (internal: dwikomb)   or
Max. F-pos, VSOP short, start fitmin [%]  :   (internal: dwi)
 "     "   VSOP short, final range   [%]  :   (internal: dwikomb)   or
Ang. range of eclipt. longitude   (real)  :   (internal: dwi)       or
Ecl. angular range, VSOP short v. (real)  :   (internal: dwi)
 "      "      "   , VSOP  full v. (real) :   (internal: dwikomb)
```

The accuracy between the theoretical arrangement of the planets – given by the positions of the pyramids, by the positions of chambers, or by a linear constellation – and the current positions of the planets is either measured in degree (difference in ecliptic longitude) or in percent (relative error F_{pos}, F'_{pos}, F''_{pos}, see sections 4.3.1, 4.3.2 and also [5]). The upper limit of these quantities has to be inserted as real number. A larger upper limit yields a larger number of detected constellations.

When the option "time interval around aphelion or perihelion" (section 3.3.9, options (3) and (4)) combined with a (large) time period is used, like 2000 BC to 4000 AD, then this indicates a "special search." At first, only passages through aphelion or perihelion with a maximum allowed error "dwi" are printed. If the current relative error is below another threshold "dwi2," a time interval of a few hours or days around this aphelion (perihelion) is also tested by scanning the interval in small time steps. Then, only constellations beyond aphelion or perihelion passage are printed, if their relative error, resp. angle, is below a third threshold "dwi3." Therefore, the corresponding input line (provided again) is followed by two additional lines:

```
Max. F-pos at aphelion/ per.  (real) [%]  :   (internal: dwi)
 "     "   consider without printing [%]  :   (internal: dwi2)
 "     "   print beyond aphelion/per.[%]  :   (internal: dwi3)
```

So, when checking a constellation with this search, there are three limits (relative errors), and the last limit is normally small compared to the other ones. A typical parameter set is: dwi = 3 %, dwi2 = 5 %, and dwi3 = 0.2 % (compare with quick start option 5). In combination with the VSOP87 short version, the latter search was used to create Table 51 in [5] with the previous program version (P3). The “fine adjustment” was done manually with the full VSOP87 version. Now, the search with quick start option 5 can be performed with the VSOP87 full version, too. Furthermore, the ”fine adjustment” is automatically done when using an automated minimum search with respect to dwi (section 3.3.9 with step width 0.0; see also example in section 3.4.10).

3.3.17 Syzygy with simultaneous transit

`All conjunctions (1),  only transits (2)  :` (internal: nurtr)

(1) all “linear constellations” (syzygies)
(2) only “linear constellations” with associated Mercury or Venus transit

Option (1): When searching for “linear constellations” with three or four planets, all constellations are printed that meet the given condition, meaning that the range of ecliptic longitudes is smaller or equal to a given angle d*L* (e.g., 5°). Constellations, which are accompanied by a Mercury or Venus transit within a few hours or a few days, are marked with an “M” or a “V.” If it is a grazing transit, the line is marked with “m” or “v”. In the second option, the detected “linear constellations” are printed only when there is an associated transit of Mercury and Venus. In this case, constellations without a transit are skipped, which reduces the size of the output.

3.3.18 Polarity (orientation of planetary orbits)

`View from ecliptic  North (1), South (2)  :` (internal: iek) or
`View from eclipt.  N (1), S (2), N/S (3)  :` (internal: iek)

(1) looking from ecliptic north
(2) looking from ecliptic south
(3) looking from ecliptic north and south

When comparing the pyramid and planetary positions in 2 dimensions, the pyramid positions are projected onto the Earth's surface, and the planetary positions, for example, onto the plane of the Earth's orbit. In this case, there are two possibilities when looking onto the planets. We can look from the ecliptic north (option (1)) or from the ecliptic south (option (2)). One figure is the mirror-inverted configuration of the other one. So, it makes a difference when comparing with the pyramid positions, where we always look from above the pyramids. Finally, option (3) simply combines both options (1) and (2). Thus, all constellations found in (1) and (2) are given in (3).

3.3.19 Complexity of output

`Output          normal (1),  extended (2)  :` (internal: io)

(1) one line for each detected constellation
(2) several lines for each detected constellation

In option (2), the size of the output for each date depends on other parameters. The orbital elements of all planets can be obtained only by using also the option: `Kepl. Equ. (3)`.

3.3.20 Mode of program output

`Mon.(1), file (2), special (3), exit (4)  :` (internal: iout) or
`Monitor (1),  mon. + file (2),  exit (4)  :` (internal: iout)

(1) output on monitor
(2) output on monitor and written into the file “out.txt”
(3) like (2) but with some output quantities being replaced (also special output for constellation 12)
(4) cancels program start

With option (1), the results are written only onto the computer monitor. When using option (2), all results are additionally written into the file “out.txt.” This file is overwritten after each program run if option (2) or (3) is used. So, in order to save the latest results, the created file “out.txt” must be renamed. Option (3) means that some parameters like

`JDE` Julian Ephemeris Day
`e` error code, when calculating the “Sun position” with FITEX (“0” means “no error”)
`it` number of iterations when using FITEX

are replaced by

`dt [days]` time difference to the next aphelion (perihelion) passage
`X5` tilt angle between the Earth's surface and the transformed ecliptic plane
`M` scaling factor between alignment of planets and pyramids (chambers)

The latter quantities allow for the reproduction of some tables in Ref. [13]. For constellation 12, in combination with some certain parameter settings, option (3) calculates all “planetary positions” in the Giza area as provided in Figs. 5 and 12 (book options 380 and 381). When typing the parameters individually for the “special” output of constellation 12, only the internal parameters ipla, imod, ivers, and ihi can be varied. With option (4), the program start is canceled. The program start and the running program can be terminated at any time by typing <Ctrl>-C or <Strg>-C, respectively.

Detailed technical information about the subroutines VSOP87X, FITEX, and all other program parts can be found in the FORTRAN source code p4.f95 (see appendix) and to some extent in chapter 4. Additional mathematical details of the astronomical calculations are provided in [5, app. A14–A16].

3.4 Some program outputs

Some basic information concerning the program output is provided already in section 3.1. With the following selected options and corresponding results, most of the expressions and parameters in the table head of the program output are explained. The output is always printed in blue and calculated with the 64-bit version of P4. Section 3.4.11 provides a compilation of all quick start options.

3.4.1 Option 0

When starting the program with the option “`0`,” the parameters can be determined individually in several menus, described previously and appearing one after the other. In the following example, the transits of Mercury in front of the Sun are calculated for the years between 1800 and 2130. The input numbers are underlined.

Input:

```
Constell.  pyr.(1),  chamb.(2),  lin.(3)  : 3
Tr. Mer.(1), Ven.(2), 3-co.(3), 4-co.(4)  : 1
VSOP87-version  full v.(1),  short v.(2)  : 1
Date  equ.L.(1), nearest (2), phases (3)
          phases and position angles (4)  : 3
Calendar  only Greg. (1), Jul./Greg. (2)  : 2
Time system         JDE/ TT (1),  UT (2)  : 1
   from  year (real): 1800
   until year (real): 2130
```

```
        Output          normal (1),  extended (2)  : 1
        Monitor (1),  mon. + file (2),  exit (4)   : 2
```

Output:

TRANSITS OF MERCURY
(geocentric transit phases, terrestrial time TT)
< option 0 >

VSOP87C, comb. search, ecliptic of date, all Mercury transits
Period (years) from 1800.00 to 2130.00, Jul./Greg. calendar

co/p	date/ time:	I	II	nearest	III	IV	sep["]a	S
	9. Nov. 1802	6:14:20	6:16: 1	8:58:36	11:41:15	11:42:57	60.9/	4
	12. Nov. 1815	0:18:43	0:20:45	2:33:28	4:46:14	4:48:16	556.1/	2
	5. Nov. 1822	1: 0:53	1: 4:17	2:24:47	3:45:19	3:48:43	-838.8/	8
	5. May 1832	9: 0:31	9: 4: 0	12:25:23	15:46:40	15:50: 9	484.7	7
	7. Nov. 1835	17:32:56	17:34:44	20: 7:49	22:40:58	22:42:46	-336.4/	6
	8. May 1845	16:20:41	16:24:23	19:36:49	22:49:10	22:52:52	-547.2	5
	9. Nov. 1848	11: 5:28	11: 7:10	13:47:39	16:28:13	16:29:56	163.0/	4
	12. Nov. 1861	5:18:34	5:20:51	7:19:35	9:18:22	9:20:38	657.9/	2
	5. Nov. 1868	5:25:43	5:28:19	7:14:10	9: 0: 3	9: 2:39	-735.1/	8
	6. May 1878	15:12:57	15:16: 6	18:59:52	22:43:30	22:46:39	287.3	7
	8. Nov. 1881	22:16:59	22:18:44	0:57:13	3:35:47	3:37:31	-231.8/	6
	10. May 1891	23:54:24	23:59:24	2:21:37	4:43:47	4:48:46	-753.6	5
	10. Nov. 1894	15:56:31	15:58:16	18:34:42	21:11:12	21:12:57	266.2/	4
	14. Nov. 1907	10:23:56	10:26:38	12: 6:52	13:47: 7	13:49:48	758.6/	2
	7. Nov. 1914	9:57:23	9:59:37	12: 3:23	14: 7:11	14: 9:25	-630.7/	8
	8. May 1924	21:44: 9	21:47:10	1:41:22	5:35:25	5:38:26	84.6	7
	10. Nov. 1927	3: 2:30	3: 4:12	5:45:57	8:27:46	8:29:28	-128.7/	6
m	11. May 1937	--	--	8:59:41	--	--	-955.5	5
	11. Nov. 1940	20:49:22	20:51:11	23:21:32	1:51:56	1:53:45	368.5/	4
	14. Nov. 1953	15:37:44	15:41:23	16:54:17	18: 7:11	18:10:50	861.8/	2
	6. May 1957	23:59:53	0: 9:54	1:14:46	2:19:37	2:29:37	907.3	9
	7. Nov. 1960	14:34:31	14:36:32	16:53:27	19:10:25	19:12:27	-527.9/	8
	9. May 1970	4:20:11	4:23:13	8:16:50	12:10:20	12:13:22	-114.1	7
	10. Nov. 1973	7:48:13	7:49:54	10:32:58	13:16: 7	13:17:49	-26.4/	6
	13. Nov. 1986	1:44: 5	1:46: 0	4: 7:57	6:29:57	6:31:52	470.5/	4
	6. Nov. 1993	3: 7:14	3:13:10	3:57:31	4:41:53	4:47:49	-926.7/	10
m	15. Nov. 1999	21:16:48	21:32:36	21:41:57	21:51:19	22: 7: 6	963.0/	2
	7. May 2003	5:14:16	5:18:44	7:53:28	10:28: 9	10:32:37	708.3	9
	8. Nov. 2006	19:13:17	19:15:10	21:42: 9	0: 9:13	0:11: 6	-422.9/	8
	9. May 2016	11:13:36	11:16:48	14:58:33	18:40:11	18:43:22	-318.5	7
	11. Nov. 2019	12:36:43	12:38:24	15:20:57	18: 3:35	18: 5:16	75.9/	6
	13. Nov. 2032	6:42:26	6:44:30	8:55:22	11: 6:17	11: 8:22	572.1/	4
	7. Nov. 2039	7:19:30	7:22:44	8:48: 4	10:13:25	10:16:39	-822.3/	10
	7. May 2049	11: 5:22	11: 8:54	14:25:43	17:42:26	17:45:58	511.8	9
	9. Nov. 2052	23:55:24	23:57:12	2:31:31	5: 5:55	5: 7:42	-318.7/	8
	10. May 2062	18:18:36	18:22:13	21:38:53	0:55:27	0:59: 4	-520.5	7
	11. Nov. 2065	17:26:32	17:28:15	20: 8:20	22:48:30	22:50:13	180.7/	6
	14. Nov. 2078	11:45: 6	11:47:26	13:43:36	15:39:48	15:42: 8	674.3/	4
	7. Nov. 2085	11:45:39	11:48:12	13:37:22	15:26:34	15:29: 7	-718.5/	10
	8. May 2095	17:24: 2	17:27:12	21: 8:40	0:50: 0	0:53: 9	309.8	9
	10. Nov. 2098	4:38:48	4:40:32	7:19:52	9:59:17	10: 1: 1	-214.7/	8
	12. May 2108	1:43:45	1:48:26	4:19:33	6:50:38	6:55:18	-724.7	7
	14. Nov. 2111	22:19:23	22:21: 8	0:56:51	3:32:38	3:34:24	283.3/	6
	15. Nov. 2124	16:54: 5	16:56:54	18:32:40	20: 8:28	20:11:17	778.9/	4

```
Computed constellations:     14035                ("/" means ascending node)
Tested planet. passages:      1041
Detected transits      :        44
Centr./grazing transits:    0 /  2       CPU-time  0: 0: 1.544 -- end of run.
```

More details about the output format is provided in section 3.4.6. This is one example of fixing the parameters individually. In the following, the results of some quick start options are presented.

3.4.2 Quick start option 1

Output:

PLANETS IN ALIGNMENT WITH THE PYRAMIDS OF GIZA
(Mercury at aphelion)
< option 1 >

VSOP87C, comb. search, ecliptic of date, "Sun" free 3D, C-M, FITEX
Ecl. N and S, years -13000.00 to 17000.00 (c2), tolerance F <= 1.50/ 1.00 %

con	k	year	Lm-Lv	Lm-Le	e	it	x-Sun	y-Sun	z-Sun	dr	P	F[%]
	-59934	-12435.214	9.349	13.126	0	85	-588.5	370.0	310.5	5.2	*	0.434
	-59768	-12395.233	13.603	20.167	0	87	-570.7	227.1	397.3	8.1		0.716
	-55865	-11455.188	12.728	18.358	0	89	-552.5	298.7	394.0	8.0	*	0.697
	-35839	-6631.888	-10.931	-15.051	0	63	-411.5	557.6	-273.4	10.5		0.889
	-31770	-5651.861	-7.629	-9.869	0	82	-482.4	571.5	-169.5	8.1		0.668
	-27867	-4711.714	-8.557	-11.707	0	75	-457.9	570.4	-209.9	4.6		0.379
	-23798	-3731.708	-5.250	-6.516	0	65	-526.9	564.6	-99.3	8.6		0.697
	-712	1828.517	-19.311	-31.682	0	114	-606.7	-146.1	-182.7	8.1	*	0.803
	450	2108.387	10.444	17.871	0	76	-676.8	153.5	282.4	7.9		0.677
	3191	2768.562	-20.247	-33.537	0	116	-596.5	-184.6	-143.6	6.5	*	0.662
12	4519	3088.413	13.779	23.191	0	46	-667.5	21.3	272.4	0.8		0.069
	7260	3748.588	-16.937	-28.312	0	129	-648.7	-110.6	-136.6	6.7	*	0.642
	11163	4688.633	-17.875	-30.163	0	115	-636.4	-148.3	-107.7	5.5	*	0.540
	12491	5008.485	16.189	26.748	0	72	-660.7	-70.6	221.9	9.3		0.866

Computed constellations: 124559 (P: polarity, * view from ecl. south)
Detected constellations: 14 CPU-time 0: 0: 2.412 -- end of run.

After "tolerance F" the two given values belong to the VSOP87 short and full version. After the limiting year dates, the parameter (c2) means that both the Julian and the Gregorian calendar as well as the corresponding decimal years are used within their time periods; (c1) means that the Gregorian calendar is applied for all times. The parameters in the header line just above the table are as follows:

con	number of constellation 1–14 or arrow "->," if date is not far from a known constellation
k	number of Mercury aphelion (or perihelion) passage (see section 3.3.15)
year	decimal year (astron. counting, which means that the year 0 exists; see section 4.9.1)
Lm-Lv	difference of heliocentric longitudes of Mercury and Venus
Lm-Le	difference of heliocentric longitudes of Mercury and Earth
e	error code from FITEX, "0" means "no error," more information in the source code (app.)
it	number of iterations when using FITEX for calculating the "Sun position" in 3D
x-Sun	x-coordinate of the "Sun position" in meters at the Giza plateau (y-Sun, z-Sun analog)
dr	accuracy of "Sun position" in the pyramid area in meters (see section 4.9.2)
P	polarity: the star "*" represents the view from the ecliptic south ("no star": ecliptic north)
F[%]	relative accuracy of comparing pyramid (chamber) positions with planetary positions

In the subroutine "FITEX," the error code "e" is named "KE." In references [5] and [13], the relative deviation F is also named F_{pos}, F'_{pos}, or F''_{pos}, depending on the way of calculation. These quantities are always the relative error when comparing the pyramid positions with the planetary positions. The options "1" and "500" are identical (compare with [5, Table 50]).

3.4.3 Quick start option 3

Output:

```
                 PLANETS IN ALIGNMENT WITH THE PYRAMIDS OF GIZA
                             (Mercury at aphelion)
                                 < option   3 >

 VSOP87C (2005) full ver.,  ecliptic of date,        "Sun" free 3D, C-M, FITEX
 Ecl. N and S,  constellation 12,   JDE =  2849079.76330, year =  3088.41 (c2)
                         date (Gregor.,TT) = 31. May  3088,  6:19: 9,  Thursday

===============================================================================
con   k      year    Lm-Lv   Lm-Le   e  it  x-Sun  y-Sun  z-Sun   dr  P   F[%]
-------------------------------------------------------------------------------
  Lm       Bm       Rm       Lv       Bv       Rv       Le       Be       Re
   xm       ym      zm        xv       yv      zv        xe       ye      ze
 xv-xm    xe-xm    yv-ym    ye-ym    zv-zm    ze-zm              rel. deviation
===============================================================================
12   4519  3088.413  13.779  23.191  0  46 -667.5   21.3  272.4   0.8    0.069
-------------------------------------------------------------------------------
 274.2350 -3.8355 0.466784 260.4560  0.3611 0.725141 251.0441  0.0001 1.010140
 0.465739 0.000000-0.031224 .704258-.172709 .004569 0.928519-0.397789 0.000001
 0.238520 0.462780-0.172709-0.397789 0.035794 0.031226            0.06939575 %
-------------------------------------------------------------------------------
 ascending node (M/V/E/Ma):    61.262371   86.535685      ---      57.966374
 inclination i  (M/V/E/Ma):     7.022736    3.405473    0.000000    1.844689
 perihelion pi  (M/V/E/Ma):    94.431801  146.691325  121.707611  356.114290
 transl. X1, X2, X3; del-t:    -0.465804   -0.000042    0.031160    0.000 days
 Euler angl. X4, X5, X6; M:   -45.993868   24.468218   43.897290    97644154.
===============================================================================

===============================================================================
pla.   x[AU]     y[AU]     z[AU]       L       B        r[AU]     Lm-L     dev.
-------------------------------------------------------------------------------
Mer  0.034394 -0.464467 -0.031224 274.2350 -3.8355  0.466784   0.0000   0.0000
Ven -0.120230 -0.715090  0.004569 260.4560  0.3611  0.725141  13.7790   1.5890
Ear -0.328133 -0.955359  0.000001 251.0441  0.0001  1.010140  23.1909   1.7809
Mar -0.742601 -1.384620 -0.003376 241.7944 -0.1231  1.571191  32.4406    ---
Jup  3.659951 -3.600495 -0.044977 315.4692 -0.5019  5.134280 -41.2342  -6.4502
Sat  6.950958  6.246050 -0.394537  41.9425 -2.4175  9.353321-127.7075  87.2925
Ura 14.561780-13.283778 -0.228606 317.6278 -0.6645 19.711836 -43.3928    ---
Nep-30.177931  0.905335  0.497313 178.2816  0.9437 30.195604  95.9534    ---
===============================================================================
 Celestial pos. in Giza     body            x[m]       y[m]       z[m]     dr[m]
-------------------------------------------------------------------------------
 Local coordinates          Sun           -667.49      21.30     272.36     0.77
 of the "planets"           Mercury         -0.12      -0.09      16.24     0.15 <
 (pyramid positions)        Venus          385.58    -239.89      33.43     0.30 <
                            Earth          739.06    -574.51      23.93     0.14 <
                            Mars          1373.30   -1232.84      34.00     0.96
                            Jupiter       4545.10    4889.86   -3044.47     5.07
                            Saturn      -10104.25   10738.27    -928.40    10.57
                            Uranus       18521.60   19497.37  -12553.34    20.57
                            Neptune      -1354.80  -43610.38   15633.42    31.99
===============================================================================
            ("<" exact deviation dr)     CPU-time  0: 0: 0.084 -- end of run.
```

Information above the first solid line of the tables:

`VSOP87C (2005) full ver., ecliptic of date` describes the VSOP version

`"Sun" free` "Sun position" at the Giza plateau free

`3D` calculation of "Sun position" in 3 dimensions

`C-M`	vertical coordinate “z” of pyramid positions at the center of mass of each pyramid
`FITEX`	calculation of “Sun position” by coordinate transformation and fit program FITEX
`Ecl. N and S`	view on ecliptic plane not fixed because of 3-dimensional calculation

The parameters below the first solid line are identical to those in section 3.4.2. The quantities below the dashed line mean the following:

`Lm Bm Rm`	heliocentric spherical coordinates of Mercury
`Lv Bv Rv Le Be Re`	heliocentric spherical coordinates of Venus and Earth
`xm ym zm`	Cartesian coordinates of Mercury; x-axis through Sun and Mercury aphelion
`xv yv zv xe ye ze`	analog Cartesian (rectangular) coordinates for Venus and Earth
`xv-xm ...`	difference of Cartesian coordinates for comparison with pyramid positions
`rel. deviation`	the relative accuracy or error F (resp. F_{pos}, F'_{pos}, or F''_{pos})
`ascending node`	ecl. longitude when the planet moves through ecl. plane from south to north
`inclination i`	tilt angle between planes of planetary orbit and Earth's orbit
`perihelion pi`	ecl. longitude for location of shortest distance between planet and the Sun
`(M/V/E/Ma)`	Mercury, Venus, Earth, and Mars
`transl. X1, X2, X3`	translation coordinates of planetary positions in 3D when using FITEX
`del-t`	time difference between current date and next aphelion/perihelion passage
`Euler angl. X4, X5, X6`	three angles for rotation of planetary configuration when using FITEX
`M`	scale factor, calculated with M = 1 AU/ X_7 (AU = Astronomical Unit)

The next table in the output contains the Cartesian and spherical coordinates of all planets from Mercury to Neptune. “Lm–L” is the difference in ecliptic longitude between Mercury and the corresponding planet, which can be used for a comparison with the accordant angle in the pyramid area. The quantity “dev.” is the deviation of “Lm–L” in degree to the angles given by the pyramid positions. The first three angles, belonging to the positions of Mercury, Venus and Earth, are quite clear (compare with Fig. 8). The deviations for Jupiter and Saturn are based on the positions of the pyramid at Abu Rawash (Jupiter) and the pyramid area in Abusir (Saturn) [5, Fig. 70, p. 150]. These pyramid locations are very near to the (transformed) orbits of Jupiter and Saturn, after coordinate transformation of all planetary positions in the solar system with respect to the pyramids of Giza.

The last table shows the local coordinates of the Sun and all planets after coordinate transformation to the pyramid area in Giza. The origin of the coordinate system is located in the center of the base area of the Mykerinos Pyramid. The x-axis points to the north, the y-axis points to the west, and the z-axis points upward. The quantity “dr[m]” in the last column is the accuracy of the calculated “Sun position” and the “planetary positions.” For the calculation of “dr” and for the meaning of “<,” see section 4.9.2 and also [13, Table 25]. The remarkable positions of “Sun” and “Mars” are highlighted.

3.4.4 Quick start option 4

Output:

```
                 PLANETS IN ALIGNMENT WITH THE PYRAMIDS OF GIZA
                            (Mercury near aphelion)
                                 < option    4 >

  VSOP87C (2005) full ver.,  ecliptic of date,        "Sun" free 3D, C-M, FITEX
  Ecl. N and S,  constellation 12,   JDE =  2849079.76330, year =  3088.41 (c2)
  Special search (interval), step number =    36,   step width =  1.000 hour(s)

 con    k      year    Lm-Lv    Lm-Le    e  it  x-Sun  y-Sun  z-Sun    dr  P    F[%]
    (  JDE        dt[h]    X5    M/10^7  h-Sun    "      "      "       "   "     "  )
 ================================================================================
```

```
 12   4519  3088.413  13.779  23.191  0  46 -667.5   21.3  272.4   0.8     0.069
  2849079.01330  -18.0  25.67  9.649  20.77 -674.6   50.3  272.1   1.6     0.145
  2849079.05497  -17.0  25.61  9.656  20.78 -674.3   48.7  272.2   1.5     0.136
  2849079.09664  -16.0  25.55  9.662  20.80 -673.9   47.0  272.3   1.4     0.126
  2849079.13830  -15.0  25.48  9.668  20.82 -673.5   45.4  272.3   1.3     0.117
  2849079.17997  -14.0  25.42  9.674  20.83 -673.1   43.8  272.4   1.2     0.108
  2849079.22164  -13.0  25.35  9.681  20.85 -672.7   42.2  272.5   1.1     0.098
  2849079.26330  -12.0  25.29  9.687  20.86 -672.3   40.6  272.5   1.0     0.089
  2849079.30497  -11.0  25.22  9.693  20.88 -671.9   39.0  272.5   0.9     0.081
  2849079.34664  -10.0  25.16  9.700  20.89 -671.6   37.4  272.6   0.8     0.072
  2849079.38830   -9.0  25.09  9.706  20.90 -671.2   35.8  272.6   0.7     0.064
  2849079.42997   -8.0  25.02  9.712  20.91 -670.8   34.1  272.6   0.6     0.057
  2849079.47164   -7.0  24.96  9.719  20.92 -670.4   32.5  272.6   0.6     0.051
  2849079.51330   -6.0  24.89  9.725  20.94 -670.0   30.9  272.6   0.5     0.047
  2849079.55497   -5.0  24.82  9.732  20.95 -669.5   29.3  272.6   0.5     0.045
  2849079.59664   -4.0  24.75  9.738  20.96 -669.1   27.7  272.5   0.5     0.046
  2849079.63830   -3.0  24.68  9.745  20.96 -668.7   26.1  272.5   0.5     0.049
  2849079.67997   -2.0  24.61  9.751  20.97 -668.3   24.5  272.5   0.6     0.054
  2849079.72164   -1.0  24.54  9.758  20.98 -667.9   22.9  272.4   0.7     0.061
  2849079.76330    0.0  24.47  9.764  20.99 -667.5   21.3  272.4   0.8     0.069
  2849079.80497    1.0  24.40  9.771  21.00 -667.1   19.7  272.3   0.9     0.078
  2849079.84664    2.0  24.33  9.778  21.00 -666.7   18.1  272.2   1.0     0.088
  2849079.88830    3.0  24.25  9.784  21.01 -666.2   16.5  272.1   1.1     0.098
  2849079.92997    4.0  24.18  9.791  21.01 -665.8   14.9  272.0   1.2     0.108
  2849079.97164    5.0  24.11  9.798  21.02 -665.4   13.3  271.9   1.3     0.119
  2849080.01330    6.0  24.03  9.804  21.02 -664.9   11.7  271.8   1.4     0.130
  2849080.05497    7.0  23.96  9.811  21.02 -664.5   10.1  271.7   1.6     0.141
  2849080.09664    8.0  23.89  9.818  21.03 -664.1    8.5  271.6   1.7     0.152
  2849080.13830    9.0  23.81  9.824  21.03 -663.6    6.9  271.4   1.8     0.164
  2849080.17997   10.0  23.74  9.831  21.03 -663.2    5.3  271.3   1.9     0.175
  2849080.22164   11.0  23.66  9.838  21.03 -662.8    3.7  271.1   2.1     0.187
  2849080.26330   12.0  23.58  9.845  21.03 -662.3    2.2  271.0   2.2     0.199
  2849080.30497   13.0  23.51  9.852  21.03 -661.9    0.6  270.8   2.3     0.211
  2849080.34664   14.0  23.43  9.859  21.03 -661.4   -1.0  270.6   2.4     0.223
  2849080.38830   15.0  23.35  9.865  21.03 -661.0   -2.6  270.4   2.6     0.235
  2849080.42997   16.0  23.28  9.872  21.03 -660.5   -4.2  270.2   2.7     0.247
  2849080.47164   17.0  23.20  9.879  21.03 -660.1   -5.8  270.0   2.8     0.259
  2849080.51330   18.0  23.12  9.886  21.02 -659.6   -7.3  269.8   3.0     0.272
 ================================================================================
                                                CPU-time  0: 0: 0.076 -- end of run.
```

This is a time scan around the aphelion passage of Mercury [13, Tab. 24] in the year 3088 AD. Theoretical and (almost) ideal values are highlighted (see explanations in [13]). The new parameters are described as follows:

`Special search (interval)`	search with Mercury near to the aphelion position
`step number`	number of time steps in the time interval for each aphelion passage
`step width`	width of time steps in hours

The first of the two rows just above the solid line at the beginning of the table is identical to that in section 3.4.2. It belongs to the very first row of numbers in the table, which gives some quantities for the date of the aphelion passage. The second of the two header rows belongs to all other rows in the table. It contains some new parameters:

`dt[h]`	time difference in hours to the middle of the time interval (aphelion passage)
`X5`	tilt angle X_5 between Earth's surface and the transformed plane of the Earth's orbit
`M/10^7`	scale factor between positions of planets and pyramids (divided by 10^7)
`h-Sun`	height of the transf. “Sun position” above the southern horizon in degree, as seen from the “Mercury position” (see Figs. 2, 18, and [5, Fig. 151], and [13, Fig. 13])

3.4.5 Quick start option 7

Output:

```
        PLANETS IN ALIGNMENT WITH THE CHAMBERS OF THE CHEOPS PYRAMID
                          (Mercury at perihelion)
                               < option   7 >

  "Keplers equation",        ecliptic of date, E-V-M, "Sun" south of sub. cham.
  Ecl. N and S, years -13000.00 to  17000.00 (c2)    angular range:  1.8500 deg

 con   k          JDE            year        Lm      Lm-Lv   Lm-Le     del1    del2  P
===============================================================================
    -56861 -2550434.93231 -11694.956  230.093  98.696 114.324   1.736  -1.186 *
    -54691 -2359541.44361 -11172.307  237.794 -98.170-115.408  -1.210   0.102
    -51783 -2103726.57489 -10471.910  248.141  95.639 116.330  -1.321   0.820 *
    -42816 -1314905.41676  -8312.191  280.237 -96.743-113.723   0.217   1.787
  1 -38913  -971561.04515  -7372.146  294.297 -97.721-115.558  -0.761  -0.048
    -36005  -715746.17643  -6671.749  304.808  95.922 116.021  -1.038   0.511 *
    -27038    73074.98170  -4511.932  337.411 -96.272-113.791   0.688   1.719
  2 -23135   416419.35331  -3571.906  351.692 -97.244-115.611  -0.284  -0.101
    -20227   672234.22203  -2871.523    2.367  96.199 115.701  -0.761   0.191 *
    -16324  1015578.59364  -1931.497   16.743  95.214 113.840  -1.746  -1.670 *
    -11260  1461055.38017   -711.848   35.477 -95.807-113.768   1.153   1.742
  7  -8352  1716870.24889    -11.465   46.276  97.477 117.232   0.517   1.722 *
  3  -7357  1804399.75177    228.177   49.978 -96.781-115.572   0.179  -0.062
     -4449  2060214.62050    928.560   60.818  96.494 115.370  -0.466  -0.140 *
 12   4518  2849035.77863   3088.293   94.434 -95.370-113.661   1.590   1.849
  8   7426  3104850.64735   3788.690  105.397  97.792 116.894   0.832   1.384 *
  4   8421  3192380.15023   4028.338  109.155 -96.348-115.453   0.612   0.057
     11329  3448195.01896   4728.735  120.159  96.809 115.031  -0.151  -0.479 *
     12324  3535724.52184   4968.383  123.931 -97.326-117.242  -0.366  -1.732
  9  23204  4492831.04581   7588.851  165.409  98.118 116.555   1.158   1.045 *
  5  24199  4580360.54870   7828.499  169.223 -95.949-115.263   1.011   0.247
     27107  4836175.41742   8528.896  180.391  97.132 114.689   0.172  -0.821 *
     28102  4923704.92030   8768.544  184.219 -96.933-117.041   0.027  -1.531
 10  38982  5880811.44428  11389.013  226.310  98.444 116.222   1.484   0.712 *
     39977  5968340.94716  11628.660  230.180 -95.586-115.017   1.374   0.493
     42885  6224155.81588  12329.058  241.512  97.455 114.354   0.495  -1.156 *
     43880  6311685.31876  12568.705  245.396 -96.577-116.788   0.383  -1.278
     54760  7268791.84274  15189.174  288.099  98.766 115.911   1.806   0.401 *
     55755  7356321.34562  15428.822  292.026 -95.262-114.737   1.698   0.773
     58663  7612136.21434  16129.219  303.521  97.771 114.045   0.811  -1.465 *
     59658  7699665.71723  16368.866  307.461 -96.255-116.505   0.705  -0.995
===============================================================================
  Computed constellations:     124558      (P: polarity, resp. view on ecliptic)
  Detected constellations:         31       CPU-time  0: 0: 0.148 -- end of run.
```

New terms and parameters:

`"Keplers equation"` calculation with orbital elements and solving Kepler's equation

`E-V-M` mapping of Earth, Venus, Mercury to King's, Queen's, and subterranean chamber

`"Sun" south of sub. cham.` “Sun position” fixed, south of subterranean chamber

`angular range` limit for angular deviations in degree when comparing the positions

`del1 del2` angular deviations δ_1 and δ_2 in degree between angles of planetary positions ($L_m - L_v$ and $L_m - L_e$) and of chamber positions in Cheops Pyramid

This run uses the orbital elements given as polynomials of third degree and solving Kepler's equation numerically. It needs less than 1 second to check more than 124,000 constellations. The results are not as precise as calculated with the short and full versions of VSOP87, but the computation is much faster and all important constellations are found.

3.4.6 Quick start option 11

Output:

```
                              TRANSITS OF MERCURY
                 (geocentric transit phases, terrestrial time TT)
                                < option  11 >

  VSOP87C, comb. search,     ecliptic of date,            all Mercury transits
       Period   (years) from   2950.00  to   3200.00,  Jul./Greg. calendar

 co/p    date/  time:   I         II       nearest      III       IV    sep["]a  S
===============================================================================
     18. Nov. 2953  12:58:21  13: 0:38  15: 2:35  17: 4:34  17: 6:52 -646.7/ 17
     19. May  2963   2:31: 0   2:34:14   6: 6:17   9:38:13   9:41:27  386.9  18
     21. Nov. 2966   6: 4:14   6: 5:57   8:48:19  11:30:45  11:32:28 -141.4/ 16
     21. May  2976   9:57:19  10: 1:20  12:54:45  15:48: 6  15:52: 6 -634.7  15
     24. Nov. 2979  23:58:20   0: 0: 9   2:31:53   5: 3:40   5: 5:30  359.5/ 14
     25. Nov. 2992  18:52:55  18:56:30  20:11:23  21:26:17  21:29:53  855.5/ 12
     19. Nov. 2999  17:34:26  17:36:30  19:52:29  22: 8:31  22:10:35 -542.5/ 17
     20. May  3009   8:48:36   8:51:37  12:39:28  16:27:11  16:30:12  190.9  18
     22. Nov. 3012  10:51:51  10:53:33  13:37:29  16:21:31  16:23:13  -36.8/ 16
     23. May  3022  17:33:33  17:39:50  19:28:51  21:17:50  21:24: 7 -836.4  15
     25. Nov. 3025   4:55:32   4:57:27   7:20:44   9:44: 4   9:46: 0  463.5/ 14
     18. Nov. 3032   6:15:14   6:23:43   6:53: 7   7:22:32   7:31: 1 -945.6/ 19
     28. Nov. 3038   0:31: 3   0:43:41   0:59:11   1:14:41   1:27:19  958.8/ 12
     21. Nov. 3045  22:13: 3  22:14:57   0:41:37   3: 8:21   3:10:15 -436.5/ 17
     21. May  3055  15:25:25  15:28:23  19:21:16  23:13:59  23:16:56  -14.3  18
     23. Nov. 3058  15:42: 5  15:43:47  18:27:21  21:11: 1  21:12:43   66.7/ 16
     26. Nov. 3071   9:56:42   9:58:47  12:10:45  14:22:47  14:24:52  567.6/ 14
     19. Nov. 3078  10:18:37  10:22: 5  11:42:28  13: 2:52  13: 6:20 -839.5/ 19
 12  18. May  3088  17:10:47  17:16: 8  19:20:59  21:25:48  21:31: 8  796.5  20
     22. Nov. 3091   2:54:52   2:56:41   5:31: 3   8: 5:30   8: 7:19 -332.3/ 17
     23. May  3101  22: 4:47  22: 7:50   1:54:59   5:41:59   5:45: 1 -212.9  18
     24. Nov. 3104  20:34:48  20:36:32  23:17:46   1:59: 5   2: 0:48  172.4/ 16
     27. Nov. 3117  14:59:23  15: 1:43  16:58:56  18:56:12  18:58:32  670.4/ 14
     20. Nov. 3124  14:42:24  14:45: 3  16:31:23  18:17:46  18:20:24 -735.1/ 19
     21. May  3134  22:49:57  22:53:44   1:52:34   4:51:19   4:55: 6  601.2  20
     23. Nov. 3137   7:38:45   7:40:30  10:20:16  13: 0: 8  13: 1:53 -226.8/ 17
     24. May  3147   4:59: 9   5: 2:27   8:32:13  12: 1:53  12: 5:11 -414.1  18
     26. Nov. 3150   1:28:23   1:30: 9   4: 7: 4   6:44: 4   6:45:50  276.0/ 16
     28. Nov. 3163  20: 6:36  20: 9:25  21:46:36  23:23:49  23:26:37  773.6/ 14
     21. Nov. 3170  19:14:30  19:16:46  21:21:19  23:25:55  23:28:10 -630.3/ 19
     21. May  3180   4:52: 1   4:55:16   8:24:42  11:54: 1  11:57:15  405.9  20
     24. Nov. 3183  12:26:11  12:27:54  15:10:53  17:53:57  17:55:40 -123.0/ 17
     24. May  3193  12: 2:16  12: 6:10  15: 4: 4  18: 1:52  18: 5:45 -612.6  18
     26. Nov. 3196   6:22:28   6:24:19   8:54:45  11:25:16  11:27: 6  379.7/ 16
===============================================================================
  Computed constellations:      10687                ("/" means ascending node)
  Tested planet. passages:        788
  Detected transits       :        34
  Centr./grazing transits:     0 /  0       CPU-time  0: 0: 1.192 -- end of run.
```

In the header, the expression “`comb. search`” means “combination search”: The search starts for each transit with the VSOP87C short version and continues with the full version. Nonetheless, no central nor grazing transit appears during this time period. The constellation number (12) at the beginning of the line is automatically generated by the program (subroutine “konst”). This program run is similar to the book option 310 [13, Tab. 31].

The parameters in the last header line are as follows:

`co`	number of constellation (such as 12), "`->`" means "near to a known constellation"
`p`	partial transit: "`m`" Mercury, "`v`" Venus; "`c`" or "`C`" central transit (not given here)
`date`	calendar date of middle of the transit, more precisely: minimum separation
`I II III IV`	times of inner and outer contact points, transit phases (see Fig. 7)
`nearest`	moment of nearest approach (min. sep.) between planet and center of the Sun
`sep["]`	minimum separation between planet and the Sun in arc seconds
`a`	ascending node: "slash" (descending node: "no slash")
`S`	serial number of transit

In our epoch, the passage of Mercury through the ascending node always takes place in November, the passage through the descending node in May. However, the ascending and descending node (a) for Mercury and Venus are not determined by the given months but independently on the basis of geometrical considerations.

Each transit of one series is labeled with the same number. In contrast, the absolute value of the serial numbers are arbitrary. Jean Meeus, for example, did not name each series with a number but with a letter A, B, C, ... [22, pp. 42 ff.]. Here, we take the numbers used on the NASA/Goddard Space Flight Center website, webmaster Fred Espenak:

http://eclipse.gsfc.nasa.gov/transit/catalog/MercuryCatalog.html

and http://eclipse.gsfc.nasa.gov/transit/catalog/VenusCatalog.html

(Note: In these links, the transit phases are given in universal time UT.) In order to always get the same serial numbers *S*, independently from the starting date of the chosen time period, the first numbers are taken from the file "inserie.t". Thus, this file is used only at the beginning of each run. All other serial numbers are determined during runtime of the program.

3.4.7 Quick start option 14

Output:

```
                              PLANETS IN A LINE (SYZYGY)
               (angular range of eclipt. longitudes dL minimized, JDE)
                                    < option  14 >

  VSOP87C, comb. search,     ecliptic of date,       linear c. Mercury to Mars
  Period   (years) -13000.00  to  17000.00 (c1),   angular r.:  6.00/  5.00 deg

 co tr    k          JDE            year   dt[days]  Lm-Lv    Lm-Le  Lm-Lma    dLmin
===============================================================================
       -62144 -3015259.12387 -12967.601 -38.133   1.757   0.0     0.105   1.757
       -61116 -2924752.04882 -12719.801  36.451  -2.558  -3.025   0.0     3.025
       -56018 -2476304.86414 -11491.995  15.891  -1.871  -3.100   0.0     3.100
       -55699 -2448270.17542 -11415.238 -11.643   2.577   3.490   0.0     3.490
       -54830 -2371781.88140 -11205.820  31.286   1.469   0.0     0.900   1.469
       -51975 -2120688.28715 -10518.350 -27.612  -1.048  -0.384   0.0     1.048
    M  -50946 -2030182.60335 -10270.553 -42.389  -0.500  -0.988   0.0     0.988
       -48544 -1818813.77667  -9691.845  24.059   4.074   0.0     1.955   4.074
       -48225 -1790778.08803  -9615.086  -2.474   3.782   3.234   0.0     3.782
       -44501 -1463199.29860  -8718.206 -21.543   2.831  -0.189   2.831   3.019
     V -40777 -1135613.28929  -7821.306 -33.392  -2.288  -2.121   0.0     2.288
       -39749 -1045106.26361  -7573.506  41.143  -4.656  -3.657   0.0     4.656
       -33463  -492135.45751  -6059.523  36.617  -0.732  -0.497   0.0     0.732
       -29579  -150537.15060  -5124.259 -38.030  -3.780  -2.775   0.0     3.780
       -28046   -15654.33423  -4754.962 -12.227   4.108   3.948   0.0     4.108
       -27177    60834.13961  -4545.544  30.882   1.499   0.0     0.676   1.499
       -24322   311928.64773  -3858.071 -27.103   2.726   3.541   0.0     3.541
       -23293   402434.40305  -3610.274 -41.808   3.463   3.984   0.0     3.984
```

```
      -20891   613802.18908  -3031.569  23.600   4.661   0.0     3.870   4.661
      -20572   641837.19908  -2954.812  -3.613   4.509   2.495   0.0     4.509
      -19384   746366.77815  -2668.619  18.379   0.226   4.109   0.0     4.109
      -16848   969416.60704  -2057.930 -22.063   4.094   0.469   0.0     4.094
      -13124  1297003.80471  -1161.026 -32.723   2.174   3.048   0.0     3.048
   M  -12096  1387510.46824   -913.228  41.449  -2.172  -0.337   0.0     2.172
       -5810  1940480.90518    600.754  36.554   0.079   0.707   0.0     0.707
       -5650  1954490.74930    639.112 -28.698   4.457   0.0     0.403   4.457
       -4621  2044997.03231    886.909 -42.875   4.093   0.0     1.425   4.093
       -2955  2191576.37670   1288.230 -20.467  -4.154   0.0    -0.771   4.154
       -1926  2282079.86037   1536.020 -37.445   0.453   2.400   0.0     2.400
         476  2493450.15754   2114.732  30.475   1.507  -0.232   1.507   1.739
         795  2521489.41998   2191.501   7.515  -3.907   0.0    -1.484   3.907
12 M    4519  2849066.01327   3088.376 -13.750  -3.397  -2.605   0.0     3.397
        5548  2939566.30702   3336.157 -33.917   3.882   0.0     0.569   3.882
        8243  3176650.26922   3985.271 -27.352  -3.312   0.0    -0.379   3.312
        8269  3178981.57686   3991.654  16.752  -0.800   1.467  -0.800   2.267
        9272  3267156.02956   4233.067 -42.053  -3.820  -0.574   0.0     3.820
       15557  3820126.65275   5747.049  41.208  -2.124   0.0    -1.952   2.124
       15717  3834138.13217   5785.411 -22.408  -2.064  -3.418   0.0     3.419
       15743  3836472.36564   5791.802  24.622   4.659   4.389   0.0     4.659
       16746  3924642.67733   6033.204 -38.324   2.520   0.0     2.339   2.520
       19441  4161725.56802   6682.315 -32.830  -3.781   0.0    -0.412   3.781
       20974  4296612.30826   7051.623  -3.103  -4.188   0.0    -3.368   4.188
   M   21843  4373097.00488   7261.031  36.229  -0.135   0.380  -0.135   0.515
       24698  4624193.65495   7948.510 -19.615  -0.632   4.214  -0.632   4.846
       26915  4819212.51626   8482.453 -28.801  -1.437  -3.049   0.0     3.049
       28129  4926065.39990   8775.007  29.292  -0.821  -3.563   0.0     3.563
       28448  4954105.08050   8851.777   6.750  -2.902   0.0    -1.249   2.902
       32172  5281682.80510   9748.654 -13.384  -0.969   0.0    -0.897   0.969
       35922  5611596.79305  10651.928  15.543  -0.971   0.043  -0.971   1.013
       36925  5699772.17611  10893.344 -42.331  -3.639   0.0    -2.630   3.639
       38113  5804287.97146  11179.498 -34.123  -2.641  -3.546   0.0     3.546
       39646  5939173.53429  11548.803  -5.573  -1.199  -3.007   0.0     3.007
       43210  6252743.27610  12407.327  41.406   0.134   2.572   0.0     2.572
   M   43370  6266755.55351  12445.692 -21.412   1.706   1.352   0.0     1.706
       43396  6269087.74498  12452.077  23.576   3.799   2.388   0.0     3.799
       44399  6357258.98958  12693.482 -38.437   3.641   1.774   0.0     3.641
       49496  6805712.91680  13921.307  35.715  -0.904  -0.924   0.0     0.924
       50844  6924245.65229  14245.838 -14.233   3.579  -0.162   3.579   3.742
       54568  7251829.78723  15142.733 -27.956   2.517   2.077   0.0     2.517
       56101  7386721.26013  15512.054   6.504  -1.307   1.289  -1.307   2.596
==============================================================================
 Computed constellations:    150628
 Number of syzygies     :        60       CPU-time  0: 0: 7.920 -- end of run.
```

New expressions and parameters:

`linear c.`	linear constellation, syzygy
`angular r.`	max. angular range, first value: short version, second value: full version of VSOP87
`co`	number of constellation
`tr`	transit, "`M`," "`V`": full transit, "`m`," "`v`": grazing transit (within a few hours or days)
`dLmin`	minimum angular range dL_{min} of ecliptic longitudes of all participating planets

The moment of minimum angular range for the ecliptic longitudes of all participating planets does not need to happen within the period of the planetary transit, but can happen shortly before or after the transit. So, the time difference between the moment of minimum angular range and transit can be a few hours or days. The angular range (`angular r.`) of 6° and 5° in the head lines belong to the short and the full version of VSOP87. The first number should be larger than the second one (see also [13, Table 29]). Otherwise, one or a few constellations can be lost.

3.4.8 Book option 250

The table [13, Tab. 25] represents all important data when the planets stand in a constellation according to the chamber arrangement in the Cheops Pyramid. This book option is identical to option 8. Mercury is placed in its perihelion 44 days before the “pyramid constellation” (option 3). Two significant locations in the Cheops Pyramid – secret chambers (?) – are highlighted.

Output:

```
          PLANETS IN ALIGNMENT WITH THE CHAMBERS OF THE CHEOPS PYRAMID
                             (Mercury at perihelion)
                                < option 250 >

 VSOP87C (2005) full ver.,  ecliptic of date, E-V-M, "Sun" free 3D mid., FITEX
 Ecl. N and S,  constellation 12,   JDE =  2849035.77863, year =  3088.29 (c2)
                       date (Gregor.,TT) = 17. Apr. 3088,   6:41:13,   Tuesday

================================================================================
con   k      year     Lm-Lv    Lm-Le    e  it  x-Sun  y-Sun  z-Sun   dr  P   F[%]
--------------------------------------------------------------------------------
  Lm       Bm       Rm       Lv       Bv       Rv       Le       Be       Re
   xm       ym      zm        xv       yv      zv       xe       ye       ze
 xv-xm    xe-xm    yv-ym    ye-ym    zv-zm    ze-zm            rel. deviation
================================================================================
12   4518  3088.293 -95.595-113.868  0 105 -21.78 -17.38  -8.76  0.20    0.570
--------------------------------------------------------------------------------
  94.2332  3.8355 0.307417 189.8280  3.3144 0.720012 208.1012 -0.0001 0.998755
 0.306728 0.000000 0.020564-.070079 .715384 .041627-0.404127 0.913342-0.000002
-0.376807-0.710856 0.715384 0.913342 0.021064-0.020566            0.56966279 %
--------------------------------------------------------------------------------
 ascending node (M/V/E/Ma):    61.260938   86.534589      ---      57.965443
 inclination i  (M/V/E/Ma):     7.022735    3.405472    0.000000    1.844689
 perihelion pi  (M/V/E/Ma):    94.429919  146.689667  121.705528  356.112069
 transl. X1, X2, X3; del-t:    -0.305872    0.001023   -0.020245    0.000 days
 Euler angl. X4, X5, X6; M:   -16.990371   -4.182910   50.631409  2316903300.
================================================================================

================================================================================
pla.   x[AU]     y[AU]     z[AU]       L        B        r[AU]     Lm-L     dev.
--------------------------------------------------------------------------------
Mer -0.022641  0.305891  0.020564  94.2332  3.8355  0.307417   0.0000   0.0000
Ven -0.708259 -0.122694  0.041627 189.8280  3.3144  0.720012 -95.5948   1.3652
Ear -0.881019 -0.470443 -0.000002 208.1012 -0.0001  0.998755-113.8680   1.6420
Mar -1.223696 -1.059317  0.015315 220.8818  0.5421  1.618585-126.6486    ---
Jup  3.428235 -3.843492 -0.038355 311.7316 -0.4267  5.150408 142.5015 177.2855
Sat  7.124162  6.065578 -0.396527  40.4114 -2.4267  9.364943  53.8218 -91.1782
Ura 14.442448-13.403693 -0.227280 317.1363 -0.6609 19.705201 137.0968    ---
Nep-30.172472  1.043277  0.493994 178.0197  0.9374 30.194544 -83.7865    ---
================================================================================
 Celestial pos. in Giza    body             x[m]       y[m]       z[m]     dr[m]
--------------------------------------------------------------------------------
 Local coordinates         Sun             -21.78     -17.38      -8.76     0.20
 of the "planets"          Mercury          -5.38     -28.43      -7.01     0.09 <
 (chamber positions)       Venus            -0.20      23.38      -2.95     0.26 <
                           Earth           -10.93      46.09      -5.20     0.18 <
                           Mars            -27.45      86.83      -3.25     0.44
                           Jupiter        -352.87     -39.18     -30.97     2.01
                           Saturn            7.15    -619.13     -60.71     3.62
                           Uranus        -1274.34    -219.61    -103.67     7.37
                           Neptune        1221.69    1474.36     162.73    10.91
================================================================================
             ("<" exact deviation dr)     CPU-time  0: 0: 0.092 -- end of run.
```

For the description of the parameters, see the table of the “pyramid constellation” (option 3, section 3.4.3). The “Sun position” is now located in the Great Pyramid, or more precisely, below the Great Pyramid, and the “Mars position” can be found about 40 m above the King's chamber (see Figs. 5, 19, and 20). The local coordinates are displayed at the bottom of the output. In the fourth written line of the output, “mid.” means that the relevant position is the spatial middle of each chamber.

3.4.9 Book option 381

Here, we get additional “planetary positions” at the Giza plateau of the four main dates in 3088 AD (see Fig. 12). The calculation of the coordinates is described in sections 4.6.3 and 4.6.4. Analog positions inside the Cheops Pyramid (Figs. 19, 20) can be computed with option 380.

Output:

```
               PLANETS IN ALIGNMENT WITH THE PYRAMIDS OF GIZA
              (more positions - coordinate system of pyramids)
                              < option 381 >

 VSOP87A (2005) full ver.,  standard J2000.0,        "Sun" free 3D, C-M, FITEX
 Ecl. N and S,  constellation 12,   JDE =  2849079.76330, year =  3088.41 (c2)
                       date (Gregor.,TT) = 31. May  3088,  6:19: 9,  Thursday

================================================================================
con   k      year      X5    M/10^7  h-Sun  x-Sun  y-Sun  z-Sun    dr  P   F[%]
--------------------------------------------------------------------------------
12   4519  3088.413   24.58   9.764  20.99 -667.5   21.3  272.4    0.8    0.069
================================================================================

                            Celestial positions in Giza
================================================================================
 body          x[m]       y[m]     z[m]   dr[m]      latitude N    longitude E
--------------------------------------------------------------------------------
   date of chambers: JDE = 2849035.77863
--------------------------------------------------------------------------------
 Sun         -667.49      21.30   272.36   0.77      29  57.9905     31  7.6813
 Mercury    -1107.01      35.38   441.03   1.09      29  57.7526     31  7.6726
 Venus       -459.18   -1006.75   613.80   0.87      29  58.1033     31  8.3204
 Earth         35.24   -1320.33   490.56   0.83      29  58.3709     31  8.5154
 Mars         910.94   -1885.15   425.24   1.21      29  58.8449     31  8.8666
--------------------------------------------------------------------------------
   date of syzygy:   JDE = 2849066.01327
--------------------------------------------------------------------------------
 Mercury     -130.71    -387.90   246.95   0.39      29  58.2810     31  7.9357
 Venus        251.70    -595.71   218.68   0.28      29  58.4880     31  8.0649
 Earth        589.42    -867.01   167.06   0.45      29  58.6708     31  8.2336
 Mars        1257.17   -1460.49   155.25   1.03      29  59.0323     31  8.6026
--------------------------------------------------------------------------------
   date of transit:  JDE = 2849067.30624
--------------------------------------------------------------------------------
 Mercury     -102.82    -358.64   222.36   0.36      29  58.2961     31  7.9175
 Venus        270.45    -565.35   200.49   0.25      29  58.4982     31  8.0461
 Earth        606.56    -841.32   153.23   0.44      29  58.6801     31  8.2176
 Mars        1269.28   -1439.92   143.75   1.02      29  59.0388     31  8.5898
--------------------------------------------------------------------------------
   date of pyramids: JDE = 2849079.76330
--------------------------------------------------------------------------------
 Mercury       -0.12      -0.09    16.24   0.32      29  58.3517     31  7.6946
 Venus        385.58    -239.89    33.43   0.11      29  58.5605     31  7.8437
 Earth        739.06    -574.51    23.93   0.33      29  58.7518     31  8.0518
 Mars        1373.30   -1232.84    34.00   0.96      29  59.0951     31  8.4611
================================================================================
                                           CPU-time  0: 0: 0.044 -- end of run.
```

3.4.10 Book option 511

In this run the date is completely free when comparing the positions of pyramids and planets. The position of Mercury is not restricted to the aphelion or perihelion, but can be everywhere on the orbit. Originally, the search for such events took place with constant time steps around each aphelion passage. As shown in [5], Mercury must always be located near the aphelion. Otherwise, no solution exists. In P3, the short version of VSOP87 was used for this search, which can be reproduced in P4 with the quick start option (5). If a constellation was found, the exact date was originally optimized by minimizing the relative error *F* “by hand” with the VSOP87 full version. The results are listed in Table 51 in book 1 [5].

Output:

```
              PLANETS IN ALIGNMENT WITH THE PYRAMIDS OF GIZA
                    (time not restricted, F minimized)
                              < option 511 >

 VSOP87C, comb. search,     ecliptic of date,        "Sun" free 3D, C-M, FITEX
 Ecl. N and S, years -13000.00 to  17000.00 (c2), tolerance F <=  0.50/ 0.10 %

con   k       year   dt[days]    X5      M/10^7   x-Sun  y-Sun  z-Sun  P   F[%]
===============================================================================
   -55865 -11455.192 -1.594   85.176   9.3060  -572.8  339.8  352.1  *  0.059
   -39921  -7615.072 -9.846   99.961   9.0800  -605.0  402.1  184.3  *  0.081
   -23632  -3691.743 -5.645   45.801   9.0661  -483.1  559.7 -168.4     0.092
     3191   2768.567  1.918  168.702  10.5076  -627.8 -134.4 -152.6  *  0.083
12   4519   3088.413 -0.198   24.801   9.7334  -669.4   28.9  272.6     0.045
    11163   4688.662 10.522  174.347   9.3062  -670.6  232.1 -136.7  *  0.065
    19301   6648.673 -4.921  176.513  10.3385  -654.2 -124.6  -65.2  *  0.015
    31176   9508.827  9.112 -167.460   9.0020  -617.6  406.0 -147.7  *  0.055
13  39314  11468.834 -7.615  167.389   9.6831  -709.1   -1.4    4.6  *  0.060
    55258  15308.976 -8.037 -159.439   9.2144  -731.5  122.0   -1.2  *  0.025
===============================================================================
 Computed constellations:    313708      (P: polarity, * view from ecl. south)
 Detected constellations:        10       CPU-time  0: 0: 6.996 -- end of run.
```

Now, it is possible to find all such optimized constellations, calculated with the VSOP87 full version, within one program run. In the provided program output, the middle section of Table 51 [5] is reproduced with the book option “511,” where the centers of mass in each pyramid form the basis for the calculations. This option is a good test for the correctness of Table 51.

When comparing this output with the middle columns of Table 51 in [5, p. 347], there are sometimes slight differences in the last digit of dt, X_5, and M. These are small deviations because of the manual fit procedure used in [5]. Actually, this option for reproducing Table 51 required a major programming effort because the algorithm should be fast but no planetary constellation should be lost. The automatic minimization of F [%] with this quick start option is normally more precise than the data in Table 51; however, the numerical differences are negligibly small. Also, the (decimal) year exhibits some small differences to the data in [5]. Generally, the reasons for such deviations are described in section 4.9.

This approach – center of mass, position free in 3 dimensions, point of time free, and full VSOP87 version – can also be investigated with another special search routine by doing an equidistant scan (in time) around each aphelion passage. The respective quick start option is 518, as mentioned in section 3.2.1. Analog calculations with the vertical coordinate being the top and the base of the pyramid can be performed with the options 517 and 519, respectively.

3.4.11 List of quick start options

Table 2: *Summary of all quick start options for the P4 program with a brief description (key words) of each. More detailed information is provided in the header lines of each output after running the program. The book options allow for reproducing the results in the book tables and provide even a few supplementary tables, not given in the books.*

option	brief description
	Quick start options
1	Pyramid positions, Mercury at aphelion, 13000 BC–17000 AD, 3D-calc., F_{pos} < 1 %
2	Pyramid positions, Mercury at aphelion, 13000 BC–17000 AD, 2D-calc., F_{pos} < 1.5 %
3	Pyramid positions, constellation 12 (May 31, 3088) with relevant information, “celestial positions” in Giza
4	Pyramid positions, time scan around Mercury passage through aphelion, constellation 12 (May 31, 3088), time span 1.5 days
5	Pyramid positions, special search around aphelion passage of Mercury, 13000 BC–17000 AD, VSOP87 short version, 3D-calc., F_{pos} (at aphelion) < 3 %, F_{pos} (beyond aphelion) < 0.2 %
6	Chamber positions in Great Pyramid, Mercury at perihelion, 13000 BC–17000 AD, 3D-calc., F_{pos} < 0.88 %
7	Chamber positions, Mercury at perihelion, 13000 BC–17000 AD, calculation by solving Kepler's equation, maximum angular deviation 1.85°
8	Chamber positions, constellation 12 (April 17, 3088) with relevant information, celestial positions in Great Pyramid
9	Chamber positions, time scan around Mercury passage through perihelion, constellation 12 (April 17, 3088), time span 1.5 days
10	Chamber positions, time not restricted, F_{pos} minimized, 3500 BC–6500 AD, F_{pos} < 0.05 %
11	Mercury transits in front of the Sun, geocentric phases, 2950 AD–3200 AD (includes constellation 12)
12	Venus transits in front of the Sun, geocentric phases, 1500 AD–4000 AD
13	Triple conjunction, planets Mercury, Venus, and Earth in a line, 2900 AD–3300 AD, “equal” longitudes, dL < 5° (includes constellation 12)
14	Fourfold conjunction, planets Mercury, Venus, Earth, and Mars in a line, 13000 BC–17000 AD, “equal” longitudes, dL < 5°
15	TYMT-test (test of program performance), Mercury transits, 3000 BC–7000 AD
	Special options
111	General information: authors, copyrights, and basis of calculations
–803	calculates start numbers of the transit series and creates “inser-2.t” in order to replace “inserie.t”
	Quick start options for book 2 [13]
170	Chambers positions, Mercury at aphelion, 13000 BC–17000 AD, 3D-calc., F_{pos} < 1 %
180	Chambers positions, Mercury at perihelion, 13000 BC–17000 AD, 3D-calc., identical to option **6**, except: F_{pos} < 1 % (→ years of “pyramid constellations”)
190	Chamber positions, Mercury at aphelion, mapping of planets: E-M-V, 3D-calc., F_{pos} < 1 %
191	“ “ , Mercury at perihelion, “ “ “ : V-M-E, “ “
192	“ “ , Mercury at aphelion, “ “ “ : V-M-E, “ “
200	identical to option **10**, except time period: 3500 BC–10700 AD and only “Gregorian years”
210	identical to option **9**, except: time span 24 days, time step 12 hours
220	identical to option **9**, except: time span 2 days, time step 1 hour
230	identical to option **4**, except: time span 24 days, time step 12 hours
240	identical to option **4**, except: time span 2 days, time step 1 hour

Table 2: *– continue –*

250	identical to option **8**
260	identical to option **3**
270	identical to option **13**, except time period: 2800 AD–3300 AD
280	identical to option **13**, except time period: 11000 AD–11700 AD (includes constell. 10 and 13 [5])
290	identical to option **14**
300	Mercury transits, geocentric phases, 1900 AD–2300 AD
310	Mercury transits, geocentric phases, 2900 AD–3300 AD (includes constellation 12)
320	Venus transits, geocentric phases, 4000 BC–0 AD
330	Venus transits, geocentric phases, 0 AD–4000 AD
350–351	Syzygy, four planets in a line, May 17, 3088; Mercury transit, min. separation, May 18, 3088
360–361	Venus transit, minimum separation, Dec. 18, 3089; syzygy, three planets in a line, Dec. 23, 3089
370	Search for "shadow-constellations," time not restricted, 0 AD–5000 AD, F_{pos} < 2 %
371	Preceding "shadow-constellation" at constell. 12, May 22, 3088, ecliptic of date, not a book table
372	" " " " " " " , J2000.0, not a book table
380	Special output, chamber positions, constell. 12, additional positions in Great Pyr. (Figs. 5, 19, 20)
381	Special output, pyramid positions, constellation 12, additional positions in Giza (Fig. 12)
385	Special output, elements of all planetary orbits for the year 0 AD, not a book table
Quick start options for book 1 [5]	
390–392	Pyramid positions, Mercury at aphelion, comparison of angles, view from ecl. north, 10000 BC–10000 AD, max. angular deviation 1.2°... 1.4° , VSOP87A; VSOP87C; "Kepler's equation"
400–402	similar to **390**–**392**, except: view from ecliptic south, 5000 BC–15000 AD
410–419	Pyramid positions, constellations: 2–4 and 8–14, VSOP98C, spherical heliocentric coordinates
420–429	identical to **410**–**419**, except: rectangular heliocentric coordinates
430–432	Pyramid positions, constellations 3, 9, and 13, planets Mercury to Neptune, VSOP87A (J2000.0)
440–442	identical to **430**–**432**, except: VSOP87C (ecliptic of date)
450	identical to option **2**
460–461	Pyramid positions, reference Mercury orbit, parameters ω, i, and τ, constellations 1–5 and 6–10
470–471	identical to **460**–**461**, rectangular heliocentric coordinates in Table 47 [5].
480–481	Pyramid positions, reference Venus orbit, only F_{pos} is given for constellations 1–5 and 6–10
490–492	identical to **417**–**419**, Pyramid positions, only the parameter X_1–X_7 for the coordinate transformation are given for the constellations 12, 13, and 14 in Table 49 [5]
500	identical to option **1**
501	identical to option **1**, except: positions are center of pyramid base instead of center of mass
502	identical to option **1**, except: pyramid positions are top of pyramids instead of center of mass, not a book table
510	Pyramid positions, time not restricted, F_{pos} minimized, 13000 BC–17000 AD, positions are top of pyramids, F_{pos} < 0.1 %
511	identical to option 510, except: positions are center of mass of the pyramids
512	identical to option 510, except: positions are center of pyramid base
517	like option **5**, except: VSOP87C full version, top of pyramids, F_{pos} (at aphelion) < 3.8 %, F_{pos} (out of aphelion) < 0.1 %, not a book table
518	like option **517**, except: center of mass of pyramids, F_{pos} (at aphelion) < 3.0 %, not a book table
519	like option **517**, except: center of pyramid base, F_{pos} (at aphelion) < 2.1 %, not a book table
Program start with input file	
999	program start with parameters from the input file "inedit.t," which can be edited manually

4. Technical and theoretical basis

This chapter is a brief description of the archaeological, geometrical, and astronomical basis of the P4 program. For the details and further background, see the corresponding references and the source code in the appendix. When proceeding from the previous program version P3 to P4, some details of geometrical aspects and time systems are slightly changed and compiled in section 4.9.

4.1 Positions at the Giza plateau

The exact coordinates of pyramid positions and chamber positions in the Cheops Pyramid are necessary for an accurate comparison with the planetary positions and are summarized here. For more details see Refs. [5, 13].

4.1.1 Positions of pyramids

The positions of the pyramids in Giza were measured very precisely within a geodetic net by Sir W. M. F. Petrie [6, 6a]. The main distances are provided on the right half of Fig. 8. The "Sun position" was determined here by graphically comparing the pyramid positions with the planetary orbits of Mercury, Venus, and Earth [5, pp. 121 ff.]. From the distance between the "Sun position" and the Mykerinos Pyramid we obtain the angles δ_1 and δ_2. The first search for planetary constellation was done by comparing these angles with the difference in ecliptic longitude of Mercury and Venus, as well as of Mercury and Earth. Later on, a 2-dimensional search was performed without a predefined "Sun position." For the 3-dimensional search, the pyramid positions in height were needed. The relative levels of the pyramid base for the Cheops Pyramid is 0.0 m, for the Chefren Pyramid 10.11 m, and for the Mykerinos Pyramid 12.68 m [9, part IV, map 1]. To obtain the coordinates of the pyramid positions, a coordinate system is defined with its origin in the center of the base area of the Mykerinos Pyramid. As said before, the x-axis is pointing to the north, the y-axis points to the west, and the z-axis points upward (see Fig. 2).

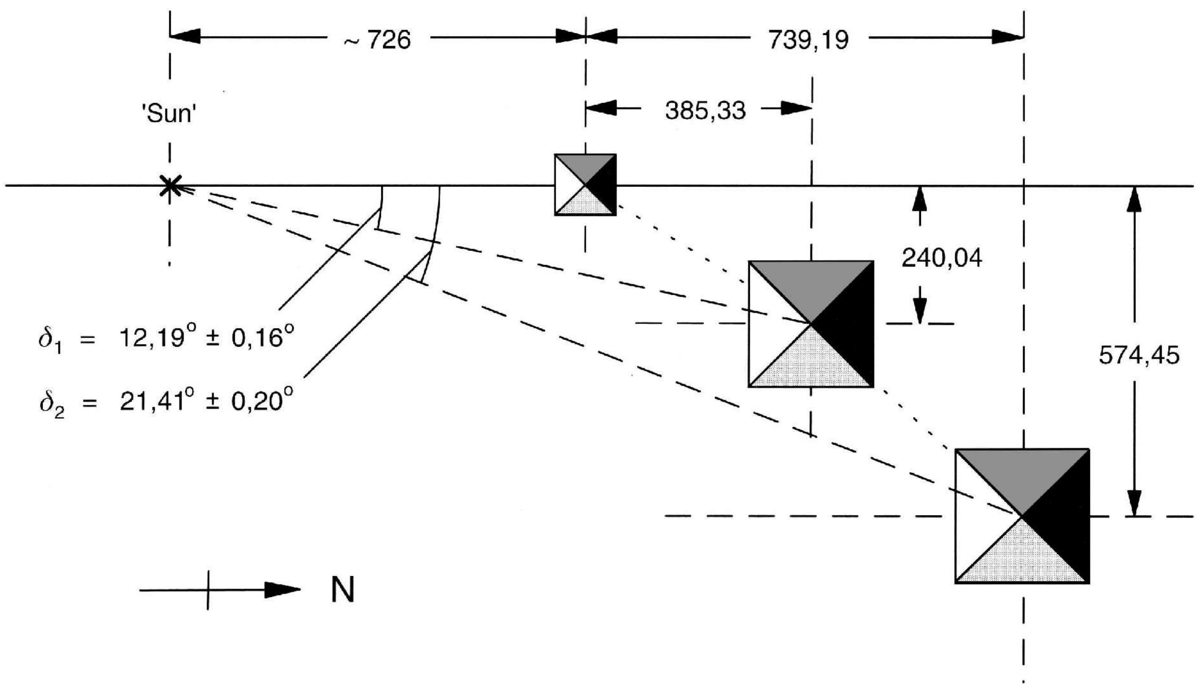


Figure 8: *Geometric relations on the pyramid plateau in Giza. The distances between the pyramids, given in meters, were measured by Petrie [6, p. 125; 6a] (see also [5, pp. 130 ff.]). For the angular errors, see [5, p. 128].*

For the height positions of the pyramids, three different levels were tested, when comparing with the planetary positions: the ground base, the center of mass, and the top of each pyramid. It can be shown mathematically that the center of mass of a pyramid is placed at a quarter of the pyramid height [5, p. 314]. With the coordinate system defined previously, the three pyramid positions have the coordinates listed in Table 3. The original heights of the pyramids are 146.59 m (Cheops Pyramid), 143.70 m (Chefren Pyramid), and 65.14 m (Mykerinos Pyramid), calculated with the base lengths and the pyramid angles from [6] (see also [5, p. 257]).

Table 3: *Coordinates of the centers of the three pyramids in Giza (in meters) [6] according to the defined coordinate system. For the z-component three options are given: the level of the pyramid base, the center of mass, and the top of the pyramid.*

pyramids	***x* [m]**	***y* [m]**	z_b **[m]** (base)	z_{cm} **[m]** (c-m)	z_t **[m]** (top)
Cheops Pyramid	739.19	–574.45	–12.68	23.968	133.91
Chefren Pyramid	385.33	–240.04	–2.57	33.355	141.13
Mykerinos Pyramid	0	0	0	16.285	65.14

4.1.2 Positions of chambers

The exact positions of the chambers in the Cheops Pyramid were taken from the drawings of V. Maragioglio and C. Rinaldi [9]. They used length and angular data from measurements and publications of Piazzi Smyth [25, 26], John and Morton Edgar [27], Howard Vyse [28], J. S. Perring [29], and W. M. F. Petrie [6, 6a].[2] The coordinates of the chamber positions, derived from the given data, are summarized in Table 4. The origin of the coordinate system is placed at the middle axis of the east wall of the Queen's chamber on the ground level of the pyramid (see Fig. 9). The x-axis points to the north, the y-axis points upward, and the z-axis points to the east, which is out of the drawing plane.

Table 4: *Coordinates of the three chambers in the Cheops Pyramid (in meters, taken from [9]) according to the coordinate system in Fig. 9. Concerning the z-component, three options are given: the middle of the east wall, the spatial middle, and the middle of the west wall of each chamber.*

chambers	***x* [m]**	***y* [m]**	z_E **[m]** (east wall)	z_M **[m]** (middle)	z_W **[m]** (west wall)
King's chamber	–11.05	45.95	0	–5.24	–10.48
Queen's chamber	0	23.54	0	–2.88	–5.76
Subterranean cham.	–5.46	–28.45	0	–7.035	–14.07

The essential data and more are visualized in Fig. 9. The numbers in italic letters are calculated here from the given data. The numbers are not always consistent, because in the charts of Maragioglio and Rinaldi, the data are based on various measurements performed by different researchers. If different data exist, the deviations are in the range of one or a few centimeters and are not relevant for the astronomical comparison. In one case the number has not been taken from the reference: the horizontal distance from the north baseline of the pyramid to the end of the descending corridor is given as 107.39 m [9, Part IV, Map 3]. This distance could not be measured directly, but has to be calculated from other distances and angles. In Fig. 9 we obtain this length by

2 Full texts of several references are available on the Internet. Some links are provided in the reference list. The references [25–29] are examples for each author. It has not been checked, whether all the data in Fig. 9 can be found therein.

the sum of 94.27 m + 13.33 m = 107.60 m. So, we rely primarily on data that could be measured directly, like for example the length of the descending corridor, and avoid possible calculation errors from the reference. At this point, it has to be said that the work of Maragioglio and Rinaldi shows outstanding quality because their encyclopedic drawings provide a huge number of measured data and valuable details.

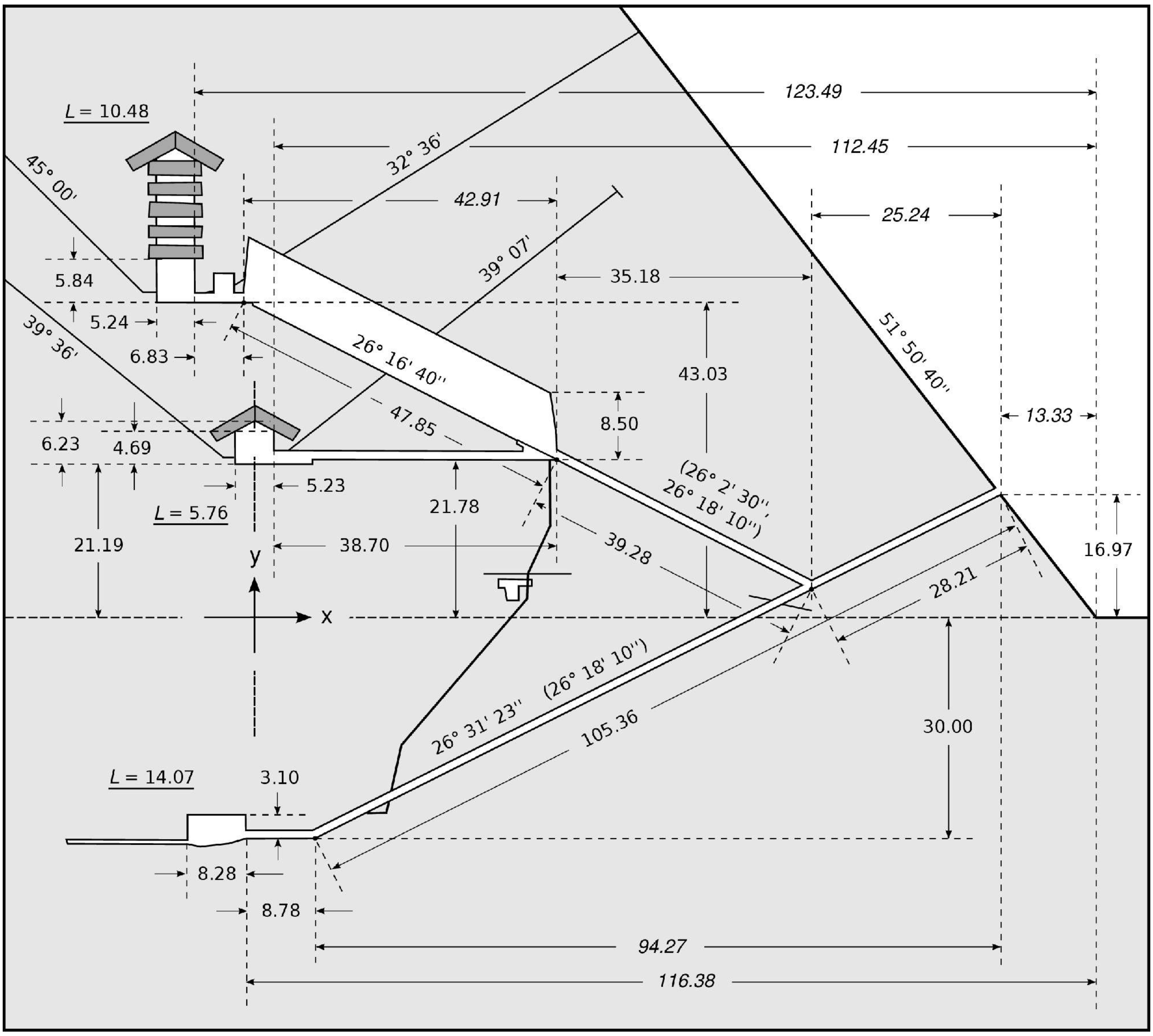


Figure 9: *Inner construction of the Cheops Pyramid as seen from the east, with the linear measurement data given in meters. The technical data are taken from detailed drawings of V. Maragioglio and C. Rinaldi [9, part IV, maps 3–7]. The numbers in italic are additionally calculated from the other data. The underlined quantities "L" are the lengths of the chambers, extending vertically from the east walls of the corridors into the depth of the drawing.*

4.2 VSOP87 – planetary positions

The VSOP87 planetary theory was developed by P. Bretagnon and G. Francou (Bureau des Longitudes, Paris, today: IMCCE, Institut de mécanique céleste et de calcul des éphémérides) [1, 2]. VSOP means "Varations Séculaires des Orbites Planétaires" and 87 is the year (1987) of publication. As said in the introduction, the files for the VSOP87 theory can be downloaded from the FTP server on the IMCCE homepage.

VSOP87 allows for calculating the positions of the planets of our solar system (Mercury to Neptune) with very high precision as a function of time. The theory includes all gravitational perturbations between the planets and relativistic effects. It is valid for a time period ranging several thousand years into the past and into the future. Although the theory was further improved (VSOP-2000, VSOP2002, and VSOP2002b (A. Fienga, J.-L. Simon) [30]), the accuracy of VSOP87 is by far sufficient for the purpose of the comparison with the Giza pyramids. The available six VSOP87 versions, differing in the kind of the used coordinate system, are compiled in the following table.

Table 5: *The six VSOP87 versions. Here, the full versions VSOP87A and VSOP87C and a short version of VSOP87D are used.*

version	kind of coordinates	coordinate system
VSOP87	Heliocentric ecliptic orbital elements (elliptical coord.)	equinox J2000.0
VSOP87A	Heliocentric ecliptic rectangular coordinates	equinox J2000.0
VSOP87B	Heliocentric ecliptic spherical coordinates	equinox J2000.0
VSOP87C	Heliocentric ecliptic rectangular coordinates	equinox of the date
VSOP87D	Heliocentric ecliptic spherical coordinates	equinox of the date
VSOP87E	Barycentric ecliptic rectangular coordinates	equinox J2000.0

4.2.1 VSOP87 full version

For technical information a brief summary is quoted from Ref. [1]: *"The VSOP82 solution is made of the perturbations developed up to the third order of the masses for all the planets. Perturbations up to the sixth order obtained by an iterative method complete the theory of the four outer planets. It also contains the perturbations of the Moon onto the Earth-Moon barycenter and the relativistic perturbations expressed in isotropic and standard coordinates. The integration constants are determined by adjustment to the numerical integration DE200."* The Planetary and Lunar Ephemerides DE200 are based on numerical integration and interpolation (JPL, Jet Propulsion Laboratory, E. M. Standish et al. [31–33]).

The next VSOP-version VSOP87 was improved in such a way that the planetary positions no longer have to be calculated from the orbital elements, based on elliptical coordinates. Instead, the positions are given directly in rectangular variables X, Y, Z, and in spherical variables L, B, r, being the longitude, latitude, and distance of a planet to the Sun ('r' = radius). Two different kinds of coordinate systems are used: the standard equinox J2000.0 and the dynamical equinox (equinox of the date). The reference J2000.0 is a fixed system and is directly linked to the reference frame of DE200. The conversion from the standard system J2000.0 to the equinox of the date is performed with a precession matrix as a function of time. This matrix, taking into account the precession of the Earth's axis, is valid for several thousand years in the past and in the future. Although further developments lead to the new versions DE405 and DE406, we guess that the modifications to DE200 are slight. All three versions are available on CD-ROM from the website of Willmann-Bell, Inc.: http://www.willbell.com/software/jpl.htm [33].

Here, the full versions VSOP87A and VSOP87C are applied using rectangular coordinates. All theoretical input of the VSOP87 theory is finally expressed in analytical expressions of rectangular coordinates in terms of periodic series and Poisson series. These sums contain up to several thousand parameters $A_{\alpha n}$, $B_{\alpha n}$, and $C_{\alpha n}$, where the index α runs from 0 to maximal 5 and n from 1 to maximal 2047 (for Saturn). As an example the analytical expression for the X-coordinate of a planet is given as a function of the time τ with $\tau = (JDE - 2\,451\,545.0)/\,365\,250.0$:

$$X(\tau) \;=\; \sum_{\alpha=0}^{\alpha(max)} \sum_{n=1}^{N(\alpha)} \tau^{\alpha} \cdot A_{\alpha n} \cdot \cos(B_{\alpha n} + C_{\alpha n}\, \tau) \tag{4}$$

N becomes smaller as α increases. The expressions for the *Y*- and *Z*-variable are analog. The conversion to appropriate spherical coordinates and other rectangular coordinates is done separately in the P4 program. From Eq. (4) and the corresponding equations for *Y* and *Z,* it is easy to get the current velocity of the planet by calculating the derivatives with respect to time (τ). So, we obtain, for example, the x-component of the velocity by:

$$v_X(\tau) \;=\; \sum_{\alpha=0}^{\alpha(max)} \sum_{n=1}^{N(\alpha)} \left(\alpha\, \tau^{\alpha-1} \cdot A_{\alpha n} \cdot \cos(B_{\alpha n} + C_{\alpha n}\, \tau) - \tau^{\alpha} \cdot A_{\alpha n} C_{\alpha n} \sin(B_{\alpha n} + C_{\alpha n}\, \tau)\right) \tag{5}$$

Nevertheless, the velocity is not needed in P4; thus Eq. (5) is provided here, because the calculation is quite simple. (The relevant program lines in P4 were converted to comment lines.) In addition to other parameters, the coefficients $A_{\alpha n}$, $B_{\alpha n}$, and $C_{\alpha n}$ are stored for each planet in one file, which is, for instance, “VSOP87A.mer” for Mercury and the standard equinox J2000.0. To improve the rapidity of computation, these coefficients are read only once from the file (from hard disc or solid-state drive) and are stored for all coordinates *X*, *Y*, *Z*, and for all planets in a single five-dimensional array for direct access. The subroutine VSOP87 has been adapted accordingly and renamed VSOP87X. Details about application of the gravitational theory and the derivation of the analytical results, respectively, can be found in Refs. [1, 2]. More technical information is provided in the files README, vsop87.doc (Table 1), and in the source code in the appendix.

4.2.2 VSOP87 short version

In “Astronomical Algorithms” of Jean Meeus, the most important periodic terms of the VSOP87D-version (spherical coordinates) are compiled in table form [17, app. II, pp. 381–422). The tables contain different coefficients *A*, *B*, and *C*, also belonging to Eqs. (4) and (5), but the series are shortened by more than 95 % and also the number of decimals is reduced. After appropriate conversion to rectangular coordinates, the results of the VSOP87D short version can be compared directly with the VSOP87C full version because both alternatives are based on the mean equinox of the date. The accuracy of the short version is not much lower than that of the VSOP87C full version. For the year 3088, the difference in the ecliptic longitudes and latitudes between the short and the full version of VSOP87 is approximately 0.0001°, which is less than 1 arc second.

4.2.3 Orbital elements and Kepler's equation

In an alternative method, we use the orbital elements listed in [17] as polynomials of third degree as a function of time in the following form:

$$a_0 + a_1 T + a_2 T^2 + a_3 T^3 \tag{6}$$

They were derived from VSOP82 [1]. The coefficients a_0 to a_3 are given for the mean equinox of the date and also for the standard equinox J2000.0 ([17, pp. 200–204] and “invsop3.t” in Table 1). The time *T* is measured in Julian centuries:

$$T \;=\; \frac{JDE - 2451545.0}{36525} \tag{7}$$

The six orbital elements for each planet are [17, pp. 197 ff.]:

L_c = mean longitude of the planet
a = semimajor axis of the orbit
e = eccentricity of the orbit
i = inclination on the plane of the ecliptic
Ω = longitude of the ascending node
π = longitude of the perihelion

The mean longitude L_c is the longitude of a body if its orbit would be circular; therefore, we use the subscript “*c*.” To get the real position of the planet, we have to solve the equation of Kepler [17, pp. 184 ff.] with respect to the eccentric anomaly E:

$$E = M + e \cdot \sin E \tag{8}$$

The mean anomaly M is given by $M = L_c - \pi$. Because Kepler's equation is a transcendental equation, it can only be solved numerically. Different iterative methods exist to find the roots of a function f, as follows:

$$f(E) = M + e \cdot \sin E - E = 0 \tag{9}$$

The following three methods are realized in the P4 program: method of Newton/Raphson, 'fixed point' method, and secant method. Only the first method is actually used. (The other methods can be activated by changing the value of the parameter “meth” in the P4 source code.) The method of Newton/Raphson is quite fast and is appropriate because the derivative $f'(E) = \partial f(E)/\partial E$ can be determined analytically. In our case the corresponding equation is

$$E_{n+1} = E_n - \frac{f(E_n)}{f'(E_n)} = E_n + \frac{M + e \cdot \sin E_n - E_n}{1 - e \cdot \cos E_n} \tag{10}$$

Iterative application of Eq. (10) yields the solution E, satisfying Eq. (8). The index “n” means the n^{th} iteration. By using

$$\tan\frac{\nu}{2} = \sqrt{\frac{1+e}{1-e}} \cdot \tan\frac{E}{2} \tag{11}$$

[34, p. 36] we get the true anomaly ν. The inclinations of Mercury and Venus orbits are only a few degrees and thus the true anomaly of a planet is nearly identical to its ecliptic longitude. The first search for planetary constellations was performed by comparing differences in ecliptic longitude of Mercury, Venus, and Earth with the corresponding angles δ_1 and δ_2 in the pyramid area (see Fig. 8). Therefore, it was a good test of the results to compare differences of the true anomalies with the same angles. Using Kepler's equation, all of the main constellations are found, although a 3-dimensional search is not done here. The ecliptic latitude B and radius r are not calculated for this search option. The computation time is much shorter than with the other VSOP87 versions.

4.2.4 Accuracy of the theory

The accuracy of VSOP87 for Mercury, Venus, Earth-Moon-barycenter, and Mars is better than 1” (1 arc second) within the time period of 2000 BC to 6000 AD and about 1” at both ends of this time span [2]. For Jupiter and Saturn, the same precision is valid for the years 0 to 4000 AD and for Uranus and Neptune from 4000 BC to 8000 AD [2]. For our purpose, which is a comparison with the positions in Giza, 1 arc second is much better than necessary; and for the important year 3088 AD, the precision is even better than that. The question is: What precision do we have for years further in the past or future, like for instance for the year 15,000 AD?

An accurate answer to this question is not easy as no more information was found in Refs. [1, 2]. The deviation of the theory is not increasing linearly with time, but stronger than that. A vague possibility is comparing the full VSOP87 version with the short VSOP87 version [17]. If we calculate the planetary positions for the beginning of the years 2000 BC and 6000 AD, the differences in ecliptic longitudes and latitudes between both theory versions are also around 1 arc second for the planets Mercury to Mars. This is the same value as the accuracy of the full version alone. If we do the same for the years 13,000 BC and 17,000 AD for angles like ($L_M - L_E$), the corresponding differences of the theory versions have a magnitude of 0.1° or 0.2°. So, even for these deviations, the precision is good enough for comparison with the pyramid positions, the chamber positions, and also for checking the planetary conjunctions ($dL \leq 5°$.) On the other hand, a deviation of 0.2° or even 0.1° is not precise enough to determine the exact transit data of Mercury and Venus. In this case, the errors of the corresponding position angles have an order of magnitude of 45°. Therefore, the transit calculations are valid primarily from 2000 BC to 6000 AD, and because the year 3088 AD is well in this range, the relevant calculations are without any problem. Another possibility (not concerning the precision) is comparing VSOP87A and VSOP87C. The angular difference $L_M - L$ should be nearly the same and the distances r should be identical. (The results are very similar to the test before.) Nevertheless, the allowed time span for applying VSOP87 is limited to the range 13,000 BC to 17,000 AD. In addition, when computing planetary transits more than 4000 years in the past or in the future, the user should be aware of this increasing uncertainty.

In order to check the results, the calculations were also performed by using the orbital elements of the planets, given by Meeus [17, pp. 200−204] and by solving Kepler's equation (subroutine “vsop3”; see, e.g., quick start option 7). In the present age, its accuracy is a bit lower than that of the other VSOP87 subroutines. However, when using years further in the past or future, the deviations do not increase as strongly as with the other routines, especially if the differences of ecliptic longitudes are computed. Therefore, this routine can be used within a longer time period, and the time limits are set to 30,000 BC and 30,000 AD (see Fig. 10 and last paragraph of section 4.9.1).

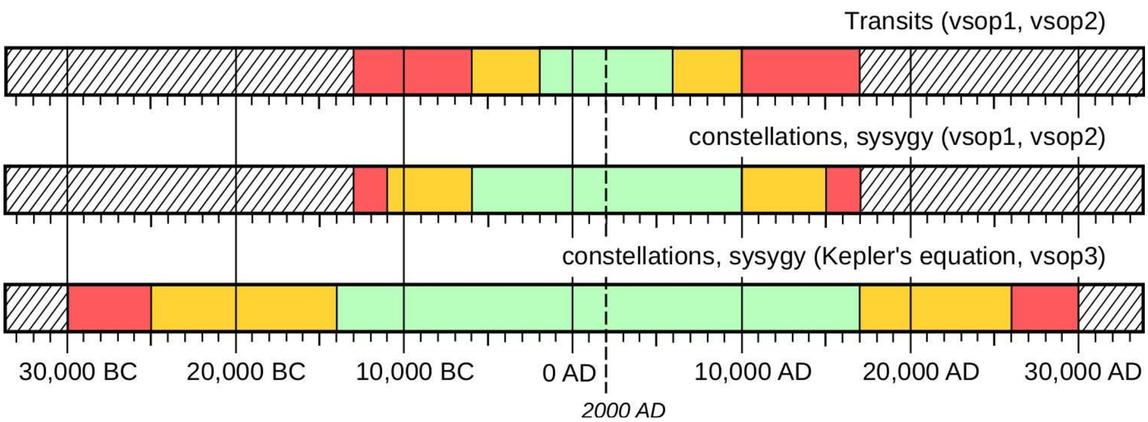


Figure 10: *Estimated time periods with different precision of the astronomical calculations. The colors have the following meaning;* ***light green****: relatively high up to very high precision,* ***yellow****: precision acceptable, care has to be taken,* ***red****: larger deviations and errors possible,* ***hatched area****: years are out of range, error message. Three different subroutines exist in the P4 program, which are based on the VSOP theory:* ***vsop1*** *(VSOP87 short version),* ***vsop2*** *(VSOP87 full version), and* ***vsop3*** *(orbital elements according to VSOP82, Kepler's equation).*

4.3 Relation between pyramid and planet positions

How can we compare the positions of the pyramids with those of the planets? As a first attempt, the arrangement of the planetary orbits were drawn to scale on a large sheet of paper, like in Fig. 11.

Then the arrangement of the pyramids was added in such a way that the Cheops Pyramid was placed on the Earth's orbit, the Chefren Pyramid on Venus' orbit, and – if possible – the Mykerinos Pyramid on Mercury's orbit. The allocation of the pyramids to the planets was done according to Eqs. (1), (2), and (3). In Fig. 11, five arrangements A to E of the pyramid positions are provided, where Earth and Venus (Cheops and Chefren Pyramid) are always located exactly on their orbits. In most cases, the planet Mercury would not reach its orbit. Only for the case that Mercury is placed at or near to the aphelion, all three planets are arranged correctly (see arrangement A in Fig. 11). (The aphelion is the place on the orbit of maximum distance to the Sun.) So, in the following, Mercury has to stand at the aphelion or near to it; otherwise, no solution exists. Interestingly, in Eq. (3), defining the relation between Mercury and Mykerinos Pyramid, the aphelion distance of Mercury is used – a remarkable coincidence! In Fig. 11, the north−south alignment of the pyramids also correlates with the main symmetry axis of Mercury's orbit. In principle, this is not necessary in this geometric test, but for the final planetary constellation this is almost the case. The distance from Mercury to the Sun in arrangement A, which is the aphelion distance, can be transferred to the Giza plateau. With this simple geometric (1-dimensional) approach, the distance was determined to be 726 m (compare with Fig. 8).

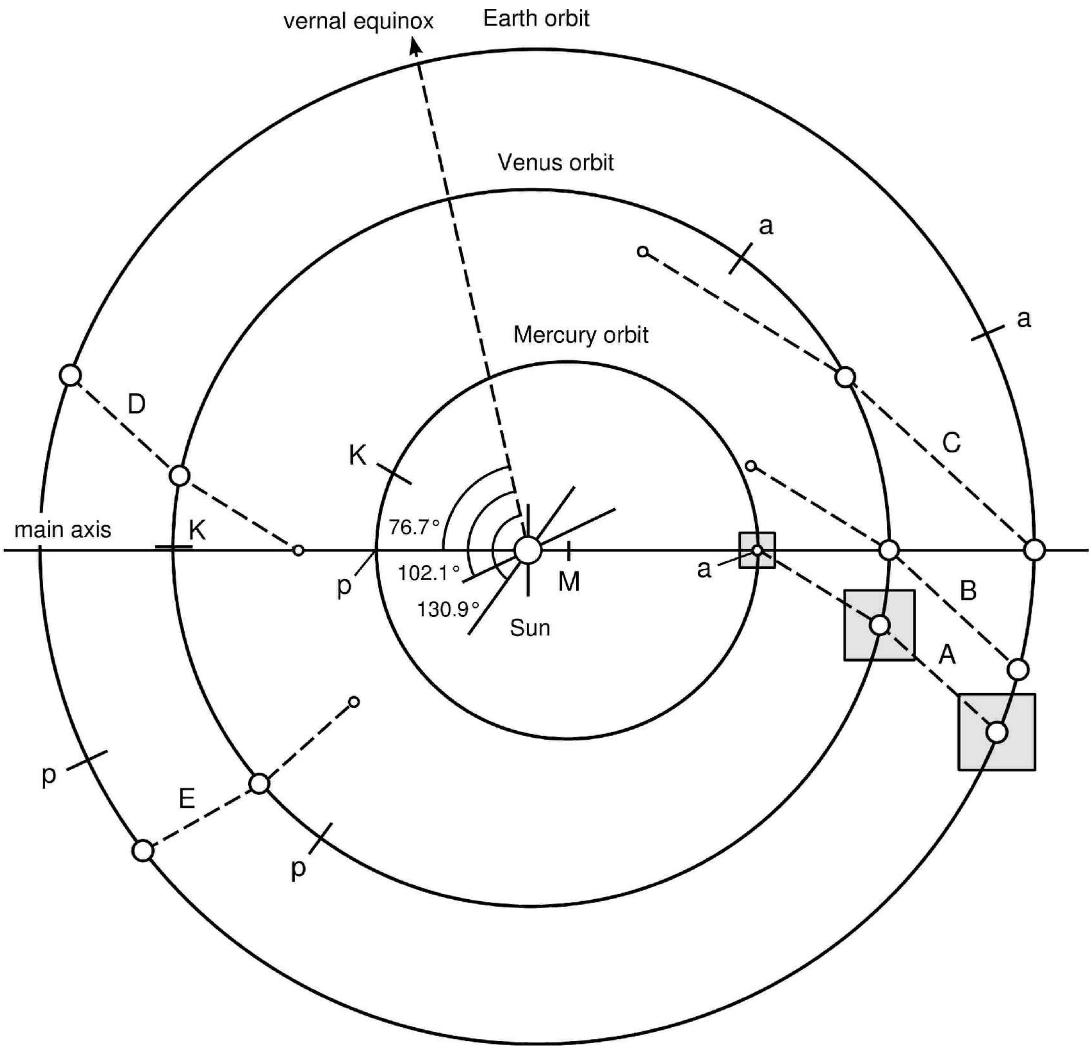


Figure 11: Approximate to-scale representation of the planetary orbits of Mercury, Venus, and Earth. "K" means ascending node, "p" perihelion, "a" aphelion, and "M" is the middle of the Mercury orbit. The polygons "A" to "E" represent each the arrangements of the three great pyramids in Giza. The constellation "A" fits almost perfectly to the pyramid positions. A solution exists only when Mercury is placed at or near to the aphelion. For better visibility, the planets were magnified by a factor 500 and the Sun by a factor 6.

The next question is: Does this situation ever happen? And if yes, when? At first, we start with the assumption that Mercury is placed exactly at its aphelion. What we need are the dates when this actually happens. Fortunately, Jean Meeus derived a formula with the VSOP87 theory for all moments when Mercury stands at the aphelion [17, p. 253]. The Julian date, used here, is:

$$JDE = 2451\,590.257 + 87.969\,349\,63\,(k - 0.5) \tag{12}$$

where k is an integer number. For $k = 0$, we obtain the first aphelion passage of Mercury after the beginning of the year 2000. Replacing $(k - 0.5)$ with k yields the perihelion passages, respectively. Now, with these aphelion dates we can start to compare pyramid and planetary positions.

4.3.1 1-dimensional comparison

The most simple approach is to compare the angles δ_1 and δ_2 at the Giza plateau (Fig. 8) with the corresponding differences of ecliptic longitudes (L). More precisely, if the index "M" stands for Mercury, "V" for Venus, and "E" for Earth, the following equations must be valid: $L_M - L_V = \delta_1$ and $L_M - L_E = \delta_2$ within a given tolerance. In this case, the "Sun position" is placed exactly south of the Mykerinos Pyramid. Another option is to locate the "Sun position" south of the Chefren Pyramid with the angles δ_1 and δ_2 being adapted accordingly (section 3.3.10). In this "1-dimensional" comparison, only one parameter, the ecliptic longitude L, is considered. For the calculation of the relative error F_{pos}, being another measure to evaluate a detected constellation, see [5, pp. 133 ff.].

4.3.2 2- and 3-dimensional comparison

The cases of 2 and 3 dimensions are treated in a similar way. Therefore, we start with 3 dimensions. Let $\boldsymbol{a}$ be the vector from the Mykerinos Pyramid to the Chefren Pyramid, $\boldsymbol{b}$ the vector from the Mykerinos Pyramid to the Cheops Pyramid, and accordingly, $\boldsymbol{a'}$ the vector from Mercury to Venus and $\boldsymbol{b'}$ the vector from Mercury to Earth. The vectors $\boldsymbol{a'}$ and $\boldsymbol{b'}$ are derived from the planetary positions, which are calculated before with VSOP87. For example, for the "pyramid date" of constellation 12 [5, 13], $JDE = 2849079.76330$, we find:

$$\boldsymbol{a} = (385.33, -240.04, 17.07)^T \tag{13}$$

$$\text{and} \qquad \boldsymbol{a'} = (0.238520, -0.172709, 0.035794)^T \tag{14}$$

The numbers in Eq. (13) are given in meters, calculated from Table 3 (compare coordinate system in Fig. 2) and those in Eq. (14) in "AU." The "Astronomical Unit" (149,597,870.61 km) is the mean distance between Earth and the Sun. The superscript "T" means "transposed." In this example, the vector $\boldsymbol{a}$ connects the barycenters of the pyramids. With a and b being the lengths of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, let $p = b/a$ be the ratio of the distances between the pyramids and $q = b'/a'$ accordingly the ratio for the corresponding planets. Furthermore, let δ_p be the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$, which can be calculated with the inner product $\boldsymbol{a} \cdot \boldsymbol{b}$ by $\delta_p = \arccos(\boldsymbol{a} \cdot \boldsymbol{b}/(a \cdot b))$, and δ_q the corresponding angle between $\boldsymbol{a'}$ and $\boldsymbol{b'}$. Now, the conditions $p = q$ and $\delta_p = \delta_q$ imply that the alignments of pyramids and planets are identical. Because these equations are (probably) never exactly valid, we define the following relative deviation F''_{pos} in percent (δ_p and δ_q in radians) [5, pp. 326, 339]:

$$F''_{pos} = 100 \cdot \sqrt{\frac{1}{2}\left(\left(\frac{q-p}{p}\right)^2 + (\delta_q - \delta_p)^2\right)} \;[\%] \tag{15}$$

For the 2-dimensional calculation, the positions of the pyramids are projected onto the surface of the Earth and the positions of the planets onto the ecliptic plane. Thus, the z-component of each vector is set to zero and Eq. (15) can also be used. The analog name in [5] for the 2-dimensional case is F'_{pos}. Concerning the pyramids and the 3-dimensional approach, the relative deviation of the main constellation 12 is only 0.07 %. In the case of the chambers in the Cheops Pyramid, the calculations are analog, with the only exception that the 2-dimensional calculation is not realized. Finally, the calculation in 3 dimensions appears to be the most reasonable one. The "Sun position" is not predefined and the date is not necessarily restricted to aphelion passages. Note: The factor 1/2 in Eq. (15) causes F'_{pos} and F''_{pos} to be rather "relative error per coordinate" than "relative error," which simplifies the comparison of 1-, 2-, and 3-dimensional calculations (not mentioned in [5].)

4.4 Two fit programs

Different programs for iterative fitting and computing of data are used in P4. Two of them are described in this section in more detail. The first one (FITEX) is more complex and was written by G. W. Schweimer. I kindly got it from the KfK (Kernforschungszentrum Karlsruhe, today KIT) where I did my PhD. The second one (ringfit) was created to improve the processing speed of P4. The improvement is only little, but the used equation seems interesting. Therefore, it is used to calculate the transit phases. Other fit algorithms, applied in P4, are described in their astronomical context in sections 4.2.3, 4.7.1, and 4.7.2.

4.4.1 FITEX

The description here is written on the basis of the program description, given by Dr. G. W. Schweimer (KfK, Cyclotron Laboratory, today KIT) [15, 16, and 5]. Originally, the code was written in FORTRAN IV, but has been adapted now to the new compilers GNU gfortran and Intel-Fortran. The program consists of four subroutines (the last four subroutines in P4) and allows us to solve the nonlinear least squares problem. It uses a least squares interpolation between variables and functions or the exact gradient of the functions.

Very often in scientific measurements the problem exists of finding some parameters of a mathematical model, so that the measured data are reproduced by the model in terms of a least squares fit. The mathematical problem is solved if the minimum of the Euclidean norm of the vector $\boldsymbol{F}$ is found by variation of the parameter vector $\boldsymbol{X}$:

$$|\boldsymbol{F}(\boldsymbol{X})| = \text{minimum} \tag{16}$$

The components of the vector $\boldsymbol{F}$ are the differences between measured values $\boldsymbol{Y}$ and model values $\boldsymbol{Z}(\boldsymbol{X})$ in terms of the measuring errors $\Delta\boldsymbol{Y}$:

$$F_\mu = \frac{Y_\mu - Z_\mu(X)}{\Delta Y_\mu} \quad , \quad \mu = 1 \ldots m , \tag{17}$$

where μ is counting the single data points. For the most part, the solution of Eq. (16) can be found only numerically. Therefore, an optimum procedure does not exist. In the given method the following information about the vector $\boldsymbol{F}$ is used:

1. The vector $\boldsymbol{F}$ has at least as many components as the vector $\boldsymbol{X}$.
2. The solution vector $\boldsymbol{X}$ is known approximately, i.e., the range for each component X_i, $i = 1 \ldots n$, is known.

The functions F_μ are calculated in the main program (P4) by the user. The subroutines of the search program are embedded in the main program and are connected through a question−answer relationship. The search program calculates the expected best vector of parameters and asks the main program for the values of the functions. The main program answers with the function values. The search program stores these values in the memory and asks again, if necessary.

The minimum of the Euclidean norm of the vector ***F*** can be found with an iterative procedure. The estimated best vector of parameters $\boldsymbol{X}_{new}$ is obtained by the linear approximation of the functions ***F***. The linear approximation is

$$\boldsymbol{F}_{lin}(\boldsymbol{X}) = \boldsymbol{H} + \boldsymbol{G}\cdot(\boldsymbol{X}-\boldsymbol{C}) \qquad \text{with} \qquad \boldsymbol{X} \neq \boldsymbol{C} \tag{18}$$

Here, the vector ***H*** and the matrix ***G*** are the approximations of the function values and of the derivatives at ***X*** = ***C***. The well-known problem of the linear least-squares fit, implemented in the search program as another subroutine, yields a stable procedure to find the vector $\boldsymbol{X}_{new}$, so that the linear approximation $\boldsymbol{F}_{lin}(\boldsymbol{X})$ becomes a minimum. The matrix ***G*** of the derivatives can be calculated analytically for simple functions. For complicated functions, it is more convenient and more effective to determine the derivatives numerically from the function vectors, calculated during the earlier iteration process. The latter procedure is used in the P4 program.

Different problems that may show up during the search are fixed by the program. Under certain conditions, it may happen that the new point $\boldsymbol{X}_{new}$ is worse, meaning that it has a larger $\boldsymbol{F}_{lin}$ than the previous best point $\boldsymbol{X}_{old}$. In this case the program would switch to a 1-dimensional search with step-size control along the straight line, connecting $\boldsymbol{X}_{new}$ and $\boldsymbol{X}_{old}$. When calculating the derivatives numerically, another difficulty might be that the rank of one of the used matrices becomes smaller than *n* (number of the components of ***X***), so that the system of supporting points collapses into a subspace. This is fixed by creating a random point in the neighborhood of the previous best point $\boldsymbol{X}_{old}$.

The program terminates the search if $|X_{neu}(i)-X_{min}(i)| < |E(i)|$ for $i = 1 \ldots n$, where $E(i)$ are the search accuracies. An estimate of the accuracy of the result follows. If the program does not terminate correctly, an error analysis is carried out. More information is available in Refs. [15, 16] and in the source code “p4.f95” within the last four subroutines (appendix).

4.4.2 Ringfit

A common method to find the roots of a function ($y(x)=0$) is the secant method. It can be used universally because the analytical derivative of the function is not needed, in contrast to the method of Newton and Raphson. Two points are fitted by a straight line and this line is extrapolated or interpolated to zero. Normally, when calculating the roots of a function, the function is not linear. The idea of the new method is to make the algorithm faster by also taking into account the curvature of the function. Instead of a linear extrapolation, a constant curvature is assumed which means a circle. Thus, instead of two points three points are fitted to a circle and the intersections with the x-axis are calculated. Therefore, this algorithm is named “ringfit.” Iterative application generates the roots of any function if it is continuously differentiable. Because “ringfit” works well and is slightly faster than the secant method, it is briefly described in a general form.

The equation of a circle is

$$r^2 = (x-x_0)^2 + (y-y_0)^2 \qquad \text{or} \qquad y(x) = y_0 \pm \sqrt{r^2-(x-x_0)^2} \tag{19a,b}$$

with r being the radius and (x_0, y_0) being the coordinates of the center of the circle. If three points (x_1, y_1), (x_2, y_2), and (x_3, y_3) of an arbitrary function (near to the x-axis) are placed on the circumference of a circle, we get the x-coordinate, x_0, of the center of this circle by:

$$x_0 = \frac{(x_1^2+y_1^2)(y_3-y_2) + (x_2^2+y_2^2)(y_1-y_3) + (x_3^2+y_3^2)(y_2-y_1)}{2\cdot\left(x_1(y_3-y_2) + x_2(y_1-y_3) + x_3(y_2-y_1)\right)} \tag{20}$$

The y-coordinate y_0 is calculated with the same equation by interchanging all x and y and leaving all of the indices unchanged. The reader can verify this result (x_0) straightforward by starting with three equations like Eq. (19a), corresponding to the three initial points, and eliminating r and y_0. To get the radius r, we insert x_0 and y_0 as well as the coordinates of one of the initial points, like e. g. x_1 and y_1, into Eq. (19a). The desired intersections of the circle with the x-axis ($y = 0$) are found by setting y in Eq. (19a) to zero and solving the equation for x. This gives

$$x_{1,2}^{(s)} = x_0 \pm \sqrt{r^2 - y_0^2} \tag{21}$$

The superscript (s) means “solution.” In most cases two solutions exist. The nearest one replaces the “worst” of the previous three points, and iterative use of this procedure yields the final solution which is the root of the original function. However, some aspects have to be considered.

1. We have to find the nearest solution to the previous three points, meaning that we have to decide whether the plus or the minus sign in Eq. (21) applies.
2. In principle, it might happen that the three points are located on a straight line. In this case, the method doesn't work because the denominator in Eq. (20) becomes zero and r becomes infinite. We overcome this situation by checking whether the denominator of Eq. (20) is zero, and if this is given we switch from “ringfit” to the secant method.
3. The term under the root in Eq. (21) can be negative. This implies that an intersection between circle and x-axis does not exist. In that case, either the three points of the initial function have not been chosen properly, or this function does not have any roots.
4. Here, “ringfit” is used to compute the transit phases. So, the x-values, representing Julian Ephemeris Days, have a lot of digits. If such numbers are squared, in many cases, numerical noise prevents correct results, also called numerical instability. Therefore, at the beginning the three initial x-values are shifted to the origin by a constant time interval to reduce their size. The shift can be, for example, x_2 which even simplifies Eq. (20). At the end of the calculation the three (new) x-values are shifted back to the old region by the same interval. This should be done always if the differences of the x-values at the beginning are much smaller than the x-values themselves. An example of how the algorithm can be implemented is given in the source code of P4 (subroutine “ringfit”).

Whereas the secant method extrapolates with straight lines, “ringfit” extrapolates with circles. The latter routine probably has not much practical relevance because here the speed gain is about 0 to 3 %. (In other applications, the improvement can be larger.) Nevertheless, it is slightly faster than the secant method and the basic idea and its equations also have an aesthetic aspect. Therefore, the routine is used here.

4.5 Coordinate transformation of planetary orbits

The 2-dimensional comparison of pyramid and planetary positions means that the height level of the pyramids above the Earth's surface and the planetary positions out of the ecliptic plane are neglected. In other words, the positions are projected perpendicularly to the Earth's surface and to the ecliptic plane, respectively, just by ignoring the z-coordinate. (The x- and y-axis are placed in the ecliptic plane.) Now, the question is: Why should we use the ecliptic plane for projecting the positions? The ecliptic plane is the plane of the Earth's orbit, the third planet. Would it be better to take Mercury's orbit, since Mercury is the first planet? In principle, it makes sense to take the plane of the Mercury or the Venus orbit as the reference plane – with the new x- and y-axis on it. In order to check this approach, a coordinate transformation from the heliocentric ecliptic coordinate system to the heliocentric coordinate system of the Mercury or Venus orbit is necessary.

The main equations of the transformation from the ecliptic to the Mercury orbit coordinate system are given without further explanation. For details and drawings of planetary orbits and their orientation see [5, app. A15]. The x-axis in the ecliptic system is defined in such a way that the Mercury aphelion is placed perpendicularly above the x-axis. In the "Mercury system" the Mercury aphelion is placed directly on the new x-axis. Concerning Mercury (index "*M*"), let Ω_M be the ecliptic longitude of the ascending node, L_M the ecliptic longitude of the aphelion, and i the inclination of the Mercury orbit. Then we define $\omega = \Omega_M - L_M$ and

$$\tau = \arcsin\left(\frac{\sin\omega}{\sqrt{1-(\sin i\ \cos\omega)^2}}\right) + \omega - \pi \qquad (22)$$

as well as $\xi = \tau - \omega$. Here, π is Ludolph's number and not the longitude of perihelion. For the derivation of Eq. (22) see [5, pp. 331–333]. Now, the transformation can be performed with the rotational matrix $\boldsymbol{R}$:

$$\boldsymbol{R}(\omega, i, \xi) = \begin{pmatrix} \cos\omega\cos\xi - \sin\omega\cos i\ \sin\xi & \sin\omega\cos\xi + \cos\omega\cos i\ \sin\xi & \sin i\ \sin\xi \\ -\cos\omega\sin\xi - \sin\omega\cos i\ \cos\xi & -\sin\omega\sin\xi + \cos\omega\cos i\ \cos\xi & \sin i\ \cos\xi \\ \sin\omega\sin i & -\cos\omega\sin i & \cos i \end{pmatrix} \qquad (23)$$

The angles ω, i, and ξ are the Euler angles. The calculation for the Venus orbit is similar. With this transformation it is possible to conduct the 2-dimensional comparison between pyramids and planets with three different reference planes: the ecliptic plane, the plane of Mercury orbit, and the plane of Venus orbit. Many more details and calculated examples are provided in [5, app. A15].

4.6 "Celestial positions" at the Giza plateau

Let us assume that the three planets Mercury, Venus, and Earth stand in a constellation identical to the arrangement of the pyramids of Giza with the following correlation: Mercury ↔ Mykerinos Pyramid, Venus ↔ Chefren Pyramid, and Earth ↔ Cheops Pyramid. This means that the three planets build a triangle in space and the pyramid positions form a triangle at the Giza plateau. If these triangles are mathematically "similar," which means that they have the same shape (not the same size), then the previous assumption is true. The question is: How can the real Sun position with respect to the planetary positions be transferred to the Giza plateau, when taking into account the pyramid positions? In the following, two ways of calculating the "Sun position" at the Giza plateau are explained by considering 3 dimensions. (For the geometrically predefined "Sun position" at Giza and for the "Sun positions" being free on the Earth's surface in 2 dimensions, see Ref. [5]).

4.6.1 "Sun position" by system of linear equations

Here again, the vectors ***a*** and ***b***, pointing from one to another pyramid as explained in section 4.3.2, define the arrangement of the three pyramids in Giza. The corresponding vectors for the planets are ***a'*** and ***b'***. The vectors ***a*** and ***b*** are always constant (because the pyramids do not move), whereas the vectors ***a'*** and ***b'*** change continuously with time. In order to obtain a vector basis for the 3-dimensional space, we create a vector ***d***, perpendicular to ***a*** and ***b***. With the vector product $\boldsymbol{a}\times\boldsymbol{b}$ and $a = |\boldsymbol{a}|$ being the absolute value (as before), we have

$$\boldsymbol{d} = -(\boldsymbol{a}\times\boldsymbol{b})\frac{a+b}{2\cdot|\boldsymbol{a}\times\boldsymbol{b}|} \quad \text{with} \quad \boldsymbol{a}\times\boldsymbol{b} = (a_y b_z - a_z b_y,\ a_z b_x - a_x b_z,\ a_x b_y - a_y b_x) \qquad (24)$$

Analogously, we get a vector ***d'*** for the planets. (The letter "***d*** " is used instead of "***c***" to be consistent with [5], because in that book ***c*** was already defined as a vector from the Chefren Pyramid to the Cheops Pyramid.) Note that the basis ***a***, ***b***, and ***d*** as well as ***a'***, ***b'***, and ***d'*** are not orthogonal, which is not necessary here. Now, we get the solution – the (transferred) "Sun position" – in two steps.

The three vectors ***a'***, ***b'***, and ***d'*** represent a basis of the 3-dimensional space. So, first we expand the vector ***s'***, which is the vector from Mercury to the real Sun, with respect to the basis ***a'***, ***b'***, and ***d'***. This means that the following system of inhomogeneous linear equations (SLE) has to be solved:

$$\boldsymbol{a'}\,x_1 + \boldsymbol{b'}\,x_2 + \boldsymbol{d'}\,x_3 \;=\; \boldsymbol{s'} \qquad (25)$$

After solving the SLE (25) [5, p. 341], we build a linear combination of the basis ***a***, ***b***, and ***d*** in the pyramid area with the solution x_1, x_2, and x_3 and get the "Sun position" ***s*** on the Giza plateau by

$$\boldsymbol{s} \;=\; \boldsymbol{a}\,x_1 + \boldsymbol{b}\,x_2 + \boldsymbol{d}\,x_3 \qquad (26)$$

One more aspect has to be considered. All pyramid vectors start at the position of the Mykerinos Pyramid. If we use the center of mass as the pyramid positions, the position of the Mykerinos Pyramid is not the origin of our coordinate system, but a quarter of the pyramid height above that. So, we have to add a quarter of the height, which is 16.285 m, to the z-component of the result ***s***. Thus, the coordinates of the "Sun position" in the pyramid area finally are (details in [5, app. A16]):

$$s_x \;=\; -665.1\text{ m}, \quad s_y \;=\; 22.8\text{ m}, \quad \text{and} \quad s_z \;=\; 273.1\text{ m} \qquad (27)$$

4.6.2 "Sun position" by coordinate transformation and FITEX

Another possible way to obtain the "Sun position" is to transform the planetary positions (coordinates) to the pyramid positions by translation, rotation, and change in the size by a "scale factor." In this case, the position of the Sun can also be transfered to the Giza plateau. At first, the problem of calculating the corresponding parameters and especially the rotation angles seems difficult, but it becomes easy if we also include FITEX. So, the solution is found by the search program. All components needed are still present in P4. For the rotation in space we take the rotational matrix ***R*** of Eq. (23).

At first, a point in time is calculated by P4 (VSOP87) when the planetary constellation and the arrangement of the pyramids match each other (F''_{pos} being minimized). This means that the arrangements – both forming a triangle – are mathematically "similar." Then the positions of the planets are adapted to those of the pyramids by translation, rotation, and "downsizing" in 3-dimensional space. For the translation, three parameters X_1, X_2, and X_3 are needed; the rotation in space means another three parameters X_4, X_5, and X_6, and change in size is given by one parameter X_7. The calculation is an iterative process. At the beginning, X_1 to X_7 are chosen more or less arbitrarily. Then the program FITEX optimizes these seven parameters by iteratively minimizing the Euclidean distances between the transformed positions of the planets and the corresponding pyramid positions. If x, y, z and x', y', z' are the coordinates of a planet before and after the transformation, the full transformation is given by

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \;=\; X_7 \cdot \boldsymbol{R}(X_4, X_5, X_6) \cdot \begin{pmatrix} x + X_1 \\ y + X_2 \\ z + X_3 \end{pmatrix} \qquad (28)$$

The search program FITEX works efficiently. The number of iterations necessary to find the solution X_1 to X_7, is approximately 50 to 150 for each constellation within ±15,000 years from present time, although seven parameters have to be optimized simultaneously. But how do we get the "Sun position"? In the heliocentric coordinate system, the Sun is placed in the origin. So, we apply the transformation of Eq. (28) to the zero vector $(0, 0, 0)^T$. As in section 4.6.1, for the date of constellation 12 we get the following coordinates of the "Sun":

$$s_x = -667.5 \text{ m}, \quad s_y = 21.3 \text{ m}, \quad \text{and} \quad s_z = 272.4 \text{ m} \tag{29}$$

The differences in the results between Eqs. (27) and (29) are about 1 and 2 m, which seems reasonable. The transformation of Eq. (28) is also used for the chamber positions in the Cheops Pyramid and – once the parameters have been found – for transforming the positions of the outer planets Mars to Neptune to the pyramid area. This second procedure is preferred because the positions do not match exactly 100 % and the small deviations are balanced by minimizing the distances with FITEX. Some examples of other "Sun positions" and "planetary positions" in the Giza area, calculated with this second method, are listed in sections 3.4.2–3.4.4 and 3.4.8–3.4.10.

4.6.3 Additional "planetary positions"

The previous section describes two methods of calculating the transformed "Sun position" inside the Cheops Pyramid (see Fig. 5). On the left half of Fig. 5 there are some more positions inside the pyramid. They belong to the transformed planets at the "pyramid's date" and at the "conjunction (syzygy) date." These positions do not seem as important as those defined by the "chambers date." However, for the sake of completeness, we describe how they can be computed. To make it more clear, these positions refer to the coordinate system of the Cheops Pyramid, and not to the "date of the chambers," but instead to the "dates of syzygy and pyramids."

The calculation is straight forward. For the "date of the chambers," the positions of the planets are adapted to the chamber positions by coordinate transformation and the fit-program FITEX. The corresponding seven parameters, X_1 to X_7, are kept for later use. Next, the planetary positions are calculated for the associated "pyramid's date," being 44 days later with Mercury at aphelion. Finally, we repeat the coordinate transformation with these new data by using the previous seven parameters, X_1 to X_7, and get the "pyramid positions" inside the Cheops Pyramid (Fig. 5). Because we need a fixed coordinate system, we use VSOP87A (J2000.0), although the results, when calculated with VSOP87C, are nearly identical. The tools in the program already exist. The trick is that we have two different points of time and must know how to use them correctly. We can do the same for the date of the planetary conjunction (syzygy) as well as for the middle of the Mercury transit. Concerning the latter date, the planetary positions are not shown in Fig. 5 because they are not much different than those of the conjunction.

What about the coordinate system of the pyramids? Here, the origin is placed at the center of the Mykerinos Pyramid. Similarly, we can also calculate the transformed planetary positions for all dates – given before – on the Giza plateau and in the urban area of Giza, respectively. The region is shown in Fig. 12, with the planetary orbits plotted accordingly. In this case, the procedure of applying the dates is reversed. From the transformation of the planetary to the pyramids positions we obtain seven new parameters, X_1 to X_7, for the "pyramid's date." Then the same coordinate transformation is done for the other points of time. Note that the "planetary orbits" in Fig. 12 are tilted against the Earth's surface by about 24.5° (see Fig. 2) so that the visible shape of the orbits becomes slightly elliptical. Some numbers are provided further down in Table 6. The tilted small rectangle at the "Sun position" (Fig. 12) is a concrete platform of 25 m wide by 50 m long, aligned to the center of the Chefren Pyramid and still existent in 2003. Today, the shape of the platform has been changed.

If the reader is interested in where these positions can be found in Giza, it would be more convenient to have the exact geographical latitude and longitude instead of the Cartesian coordinates in meters (GPS coordinates in section 3.4.9). In this case, it is easy to find the locations with a GPS receiver. The corresponding calculation is described in the following section.

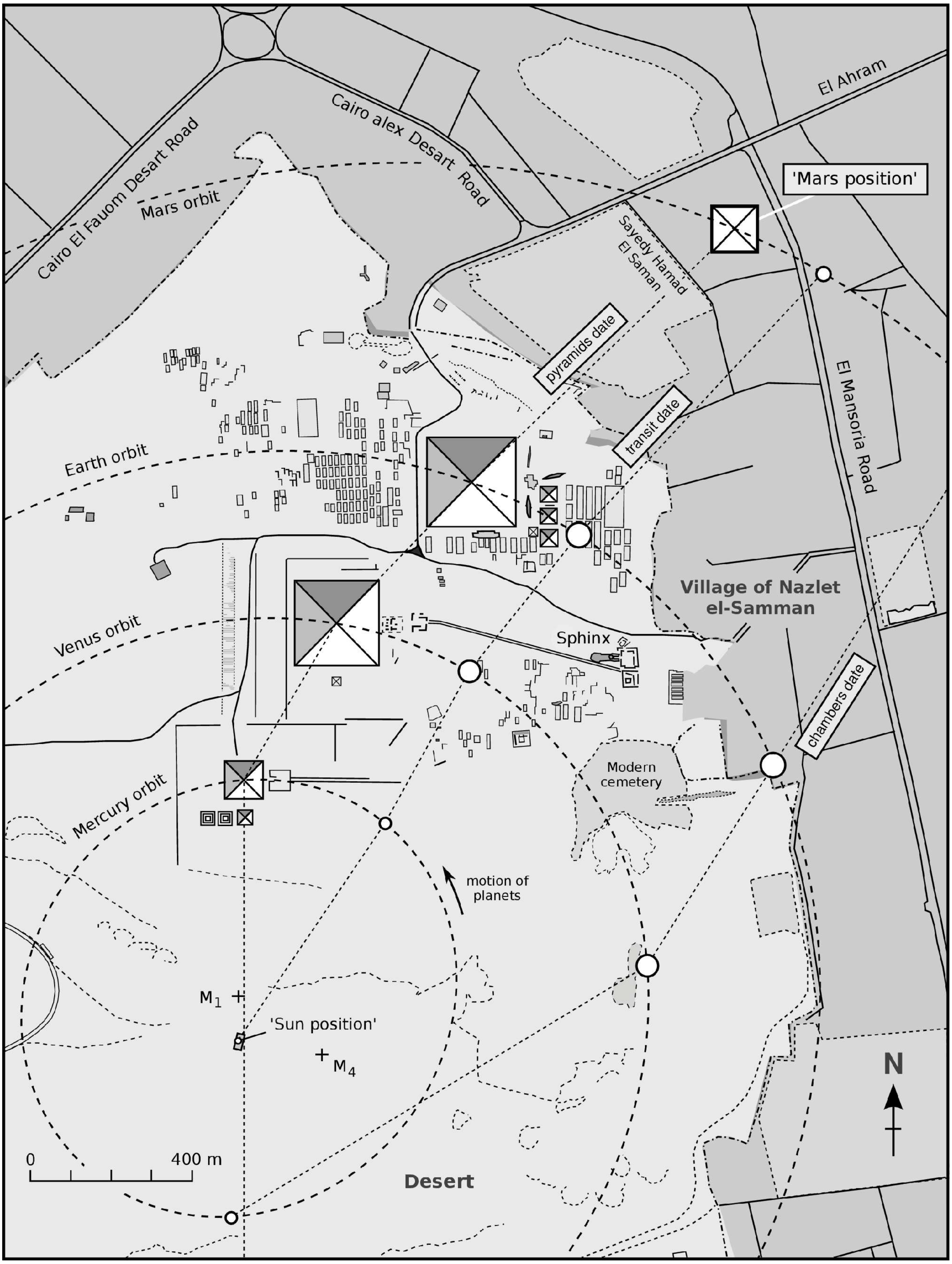


__Figure 12:__ Pyramids plateau of Giza and the neighboring village. The transferred planetary orbits are projected vertically to the Earth's surface. The sizes of the planets are magnified with respect to the orbits. The Mars position, which belongs to the pyramids, is represented by a white pyramid (29° 59.095' N, 31° 8.461' E). Its size is adapted roughly in proportion to Mars. The points M_1 and M_4 are the orbital centers for Mercury and Mars. Other planetary positions belong to the date of the Mercury transit and to the "chamber's date" (3088 AD). The GPS coordinates can be calculated with option 381 and are provided in section 3.4.9. Background created on the basis of Google Maps; © 2015 Google, ORION-ME.

4.6.4 Geographical coordinates

The conversion to latitude and longitude is not trivial if done properly. One reason is that we have to match a flat area, given in Cartesian (rectangular) coordinates, to the surface of a sphere; and another reason is that the Earth is not even an exact sphere, but an ellipsoid or spheroid. To get accurate results, we have to consider the mathematical definition of the geographical latitude. The cross section of the Earth along the rotational axis is an ellipse. So, we begin with some basic equations.

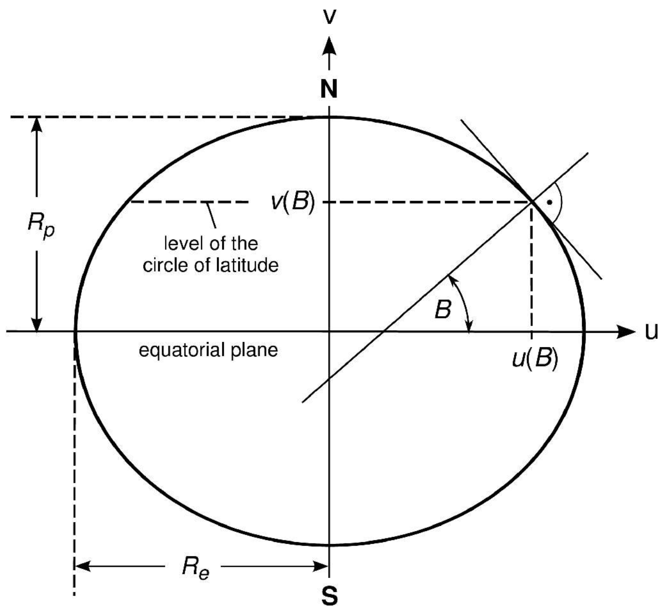


Figure 13: *Schematic elliptical-shaped cross section of the Earth. The geographical latitude B is the cutting angle between the tangent normal and the equatorial plane.*

If the Earth would be a sphere with radius *R,* then the coordinates in Fig. 13 would be $u = R \cdot \cos(B)$ and $v = R \cdot \sin(B)$. (Here, the common identifiers *x* and *y* are replaced by *u* and *v* in order to not be confused with the rectangular coordinates *x, y,* and *z* at the Giza plateau.) If considering the elliptic shape of the Earth's cross section, the calculation becomes a bit more complicated. With R_e and R_p being the Earth's equatorial and polar radius (see Table 7), the equation for the Earth ellipse in Fig. 13 is

$$\left(\frac{u}{R_e}\right)^2 + \left(\frac{v}{R_p}\right)^2 = 1 \qquad (30)$$

It follows

$$v(u) = \pm R_p \sqrt{1 - \left(\frac{u}{R_e}\right)^2} \qquad (31)$$

As shown in Fig. 13, the geographical latitude is the angle *B* of the intersection between the tangent normal and the equatorial plane. The tangent normal per definition is aligned perpendicularly to the tangent, whose slope is the derivative of the elliptical-shaped function with respect to *u.* More precisely, the derivative is the tangent of $B - \pi/2$. From Eq. (31) we obtain

$$\frac{dv}{du} = \mp \frac{R_p u}{R_e \sqrt{R_e^2 - u^2}} = \pm \tan\left(B - \frac{\pi}{2}\right) \qquad (32)$$

In order to get *u* as a function of *B,* we solve Eq. (32) for *u*. The positive solution is

$$u(B) = R_e \left[1 + \left(\frac{R_p}{R_e} \tan B\right)^2\right]^{-\frac{1}{2}} \qquad (33)$$

Using the procedure described in section 4.6.3, the P4 program calculates the rectangular coordinates *x, y,* and *z* of any planetary position, which is transferred from space to the pyramid area. As mentioned previously, the x-axis points to the North, the y-axis points to the west, and the z-axis points upward. The origin of the coordinate system is placed in the center of the base area of the Mykerinos Pyramid. In this case, the z-coordinate, being more or less the position above or under ground, is not relevant. Only *x* and *y* are converted to the geographical coordinates *B* and *L,* enabling the use of GPS. (Note that the coordinate system of the chambers is different.)

The center of the Mykerinos Pyramid is located at a latitude of B_0 = 29° 58.3518' N and a longitude of L_0 = 31° 7.6946' E (or B_0 = 29.972530°N, L_0 = 31.128243°E, measured in Giza by averaging the GPS coordinates of the 4 pyramid corners). It defines the origin of the coordinate system. Let x and y be the rectangular coordinates of a calculated “planetary position”, measured from the Mykerinos Pyramid, then we calculate the corresponding geographical coordinates B and L. The differences $\Delta B = B - B_0$ and $\Delta L = L - L_0$ are related to x and $-y$ in the “pyramids system” and to x and z in the “chambers system.” In the following, the subscript “0” always refers to the point of origin.

The latitude is calculated in two steps. First, we determine an approximate value of the difference in latitude by $\Delta B_a = x \cdot 360°/U$, with U = 40 008 km being the circumference of the Earth, measured across the poles. (Here, the subscript “a” always means “approximate.”) So, an approximate value of the desired latitude is given by $B_a = B_0 + \Delta B_a$. Next, we determine the exact latitude B. For the Mykerinos Pyramid we get the geocentric coordinates u_0 and v_0 by inserting B_0 into Eq. (33) and then applying Eq. (31). Similarly, we obtain approximate values u_a and v_a with Eqs. (33) and (31) by inserting B_a. The Euclidean distance x_a between two points having the latitudes B_0 and B_a and the same longitude, is

$$x_a = \sqrt{(u_a - u_0)^2 + (v_a - v_0)^2} \tag{34}$$

Next, we correct the value ΔB_a by calculating the difference $\Delta B = \Delta B_a \cdot |x/x_a|$. On the one hand, distances like x and y mean straight lines, and on the other hand the surface of the Earth is not flat but slightly curved. Concerning the pyramids, the distances are in the range of 1 or 2 km, meaning that the differences in latitude and longitude are less than 0.02°. For such small angles α, we get a very good approximation: $\sin\alpha \approx 2\sin(\alpha/2) \approx \alpha \approx \tan\alpha$, where α is given in radians. The reader might verify that the term $2\sin(\alpha/2)$ is the Euclidean distance between two points on a sphere with radius 1. (This distance would be measured in a straight line through the Earth.) Thus, we neglect the curved nature of the Earth's surface and get the final latitude $B = B_0 + \Delta B$.

When calculating the longitude, in principle there is another problem. If we have the exact coordinates x and y, then it makes a small difference whether we go at first x meters to the north and then y meters to the east (at the latitude B), or if we go y meters to the east (at the latitude B_0) and then x meters to the north. The reason why is simple. If we move a constant “west−east” distance $|y|$ from the equator “upward” toward the North Pole, the corresponding difference in geographic longitude ΔL becomes continuously larger. Although this effect is quite small for distances of a few kilometers, we balance the result by using the arithmetic mean of the latitudes: $B_m = (B_0 + B)/2$. This means that we first go $x/2$ meters to the north, then y meters to the east, and again $x/2$ meters to the north.

Once more we use Eq. (33) and calculate $u(B_m)$. With $2\pi u(B_m)$ being the circumference of the circle of latitude B_m, we obtain

$$\frac{\Delta L}{360^o} = \frac{y}{2\pi u(B_m)} \tag{35}$$

which yields ΔL. Note that y means –y for the pyramids and z for the chambers. Finally, we get the geographical longitude by $L = L_0 + \Delta L$.

Note: The perimeter U of a circle with radius R is given by $U = 2\pi R$. Surprisingly, the perimeter of an ellipse U_{ell} can be calculated only numerically. However, about 100 years ago the Indian mathematician Srinivasa Ramanujan found the following analytical approximation for the circumference of an ellipse. By using the Earth radii this is

$$U_{ell} \approx \pi(R_e + R_p)\left(1 + \frac{3\lambda^2}{10 + \sqrt{4 - 3\lambda^2}}\right) \quad \text{with} \quad \lambda = \frac{R_e - R_p}{R_e + R_p} \tag{36}$$

This formula is very interesting because for low and medium eccentricities it is extremely precise and probably nobody (on Earth) knows whether it can be deduced mathematically and how Ramanujan found it. An example of “planetary positions” in the Giza area is provided in Table 6 by using the date of the Mercury transit (calculation with option 381 or 0, section 3.4.9; see Fig. 12).

Table 6: *Geographical positions at the date of the Mercury transit (May 18, 3088, 19:20:59, TT).*

corresponding planet	Mercury	Venus	Earth	Mars
Latitude (North)	29° 58.2961'	29° 58.4982'	29° 58.6801'	29° 59.0388'
Longitude (East)	31° 7.9175'	31° 8.0461'	31° 8.2176'	31° 8.5898'

For the chambers system (Fig. 5), such calculations do not make much sense because the chambers are separated by only a few meters and there is no GPS reception inside the Cheops Pyramid. Nevertheless, in P4 the geographical coordinates are also calculated for the chambers.

4.7 Syzygy

4.7.1 Planetary conjunctions

The condition for planetary conjunctions is that the ecliptic longitudes L of all participating planets are similar within a given angle dL_0 , e.g., $dL_0 = 5°$. The ecliptic latitudes, which describe the positions out of the ecliptic plane, are neglected. The two main options are “3 planets in conjunction” (Mercury, Venus, and Earth) and “4 planets in conjunction” (Mercury, Venus, Earth, and Mars). In order to save computation time, the chronological search happens mostly in “large steps” with a special search after each step. This “large step” is (mostly) the synodical period of Venus and Earth of approximately 584 days. We start with a conjunction and after each step, when Venus and Earth stand again in conjunction, the overall range dL, including all participating planets, is minimized as a function of time. If the minimized angle dL_{min} is smaller than the limit dL_0 , a new syzygy is found.

For the minimization of dL, being an iterative process, the difference in L for all planets has to be checked pairwise. Now, three planets mean three differences and four planets mean six differences. So, after the minimization procedure, the condition is that the maximum of all differences must be lower than the given limit dL_0. The minimization algorithm uses three points of time with equal time intervals. Let the angular ranges dL_1, dL_2, and dL_3 be the associated function values. If the corresponding three points of time are in ascending order, then the algorithm to minimize dL goes like this: At the beginning both time differences are 5 days. For $dL_1 \leq dL_2 \leq dL_3$ the three points of time are shifted to the left (to earlier times) by one interval, for $dL_1 > dL_2 > dL_3$ they are shifted to the right; and for $dL_1 > dL_2 \leq dL_3$ they move closer together by the (optimized) factor 5. If the difference between two times is lower than the search minimum ε or if $dL_1 \leq dL_2 > dL_3$, meaning “numerical noise,” the procedure is terminated and the solution is found (subroutine “fitmin,” 1. method). In P4 a check automatically follows to determine whether a simultaneous Mercury or Venus transit happens. Therefore, we continue with the transits in front of the Sun.

4.7.2 Transit phases

Three different ways to determine these Mercury and Venus transits are provided in P4. The first two options are quite simple. In the case that, e.g. , Mercury has the same ecliptic longitude as Earth, it is checked by plane geometry whether the ecliptic latitude of Mercury B_M is small enough that the planet stands in front of the solar disk. In the second option, the condition of “identical ecliptic longitudes” is replaced by “minimum separation” between planet and the Sun. These two options are not very precise because the finite speed of light is neglected. Therefore, only the third option is explained in more detail.

This option includes the calculation of geocentric phases and minimum separation of a transit (see Figs. 7 and 14). Here, the term “geocentric” has the meaning “as seen from the center of the Earth.” For the calculation of the geocentric phases we need the diameters of Mercury, Venus, and Sun, which are summarized in the following table. The radii of Mercury and Venus are taken from “Transits” [22, p. 16], and the solar radius is taken from a recent measurement of Brown/Christensen-Alsgaard [35]. The given radius of Venus includes the opaque atmosphere of nearly 50 km height.

Table 7: *Optical size of the celestial bodies, Earth radii: IERS (2003)*

	radius [arc sec]	radius [km]
Sun	958.97	695508
Mercury	3.3629	2439.0
Venus	8.4100	6099.5
Earth, equatorial radius	8.7941	6378.1366
Earth, polar radius	8.7647	6356.7519

If taking the solar radius of 695,990 km, used by Meeus [22], the results are identical to those of Meeus in almost all cases, apart from some rounding effects. (If desired, the solar radius can be adapted easily in the source code p4.f95.) As an example, we now take the data of Mercury, but the arguments are analogous for Venus. The letters L, B, and r characterize the heliocentric spherical coordinates of a planet, and the subscripts “E” and “M” mean Earth and Mercury. Let α be the separation between Mercury and the Sun as seen from the center of the Earth, then we find

$$\alpha = \arctan\left(\frac{r_M \sqrt{1 - \cos^2\beta}}{r_E - r_M \cos\beta}\right) \qquad (37)$$

with

$$\cos\beta = \sin B_E \sin B_M + \cos B_E \cos B_M \cos(L_E - L_M) \qquad (38)$$

The angle β is the separation of Mercury and Earth as seen from the center of the Sun. Eq. (37) is deduced by plane geometry from the astronomical triangle Earth-Mercury-Sun, and Eq. (38) is in principle the spherical law of cosines (trigonometry in 3 dimensions). For $B_E = 0$, which is the case when using the ecliptic of date (VSOP87C), Eq. (38) reduces to

$$\cos\beta = \cos B_M \cos(L_E - L_M) \qquad (39)$$

At the beginning, Eq. (38) was used in P4 because it is universally valid. In contrast to this approach of spherical trigonometry, another possibility is provided on the basis of vector analysis. If $\boldsymbol{r}$ is a vector from the Sun to a planet in rectangular coordinates, and by applying the inner product of two vectors and the absolute value (length) of a vector $|\boldsymbol{r}|$, we get the separation by

$$\alpha = \arccos\left(\frac{-\boldsymbol{r}_E \cdot (\boldsymbol{r}_M - \boldsymbol{r}_E)}{|\boldsymbol{r}_E| \cdot |\boldsymbol{r}_M - \boldsymbol{r}_E|}\right) \qquad (40)$$

Both Eqs. (37) and (40) yield the same results, but finally, Eq. (40) is used because the calculation is slightly faster. Considering the transit of Venus, Eqs. (37) to (40) can be taken by replacing the indices “M” by “V.” But how do we get the exact geocentric transit phases? If s and s' are the angular radii (semidiameters) of Sun and Mercury or Venus as seen from the Earth, then we obtain the outer contact points 1 and 4 with

$$\alpha = s + s' \tag{41}$$

and the inner contact points 2 and 3 with

$$\alpha = s - s' \tag{42}$$

(compare Figs. 7, 14). But we have to be careful. If, for example, Eq. (41) is fulfilled and we have calculated the planetary positions for one point of time, it does not mean that we see the planet in contact with the Sun. If the light is coming from the “contact point” on Mercury's surface to the Earth, it needs approximately 5 or 6 minutes. During this time, the Earth has moved away from the point, where we wanted to make the observation. In short, we have to consider the finite speed of light.

Let us assume that the light from the Sun's circumference passes Mercury at the time t_M and reaches Earth at the time t_E. The difference $\Delta t = t_E - t_M$ is the travel time of the light. If t_M and the position of Mercury is given, we need the position of Earth to calculate Δt; on the other hand, we need Δt to calculate the position of Earth. So, it seems as if we have a problem. Fortunately, this can be solved iteratively. In the following, c is the speed of light and ε is the search accuracy, such as ε = 0.1 s. The time t_M is given and the problem now is to determine the exact time t_E, when the light, starting from Mercury at t_M, reaches the Earth. The problem can be solved with the following “fixed point” algorithm:

Step 1: Calculate the position of Mercury $\boldsymbol{r}_M$ with VSOP87 at the time t_M and set initial travel time of light (arbitrarily) to Δt = 320 s.

Step 2: Calculate the position of Earth $\boldsymbol{r}_E$ with VSOP87 at the time $t_E = t_M + \Delta t$.

Step 3: Calculate the optical path length between Mercury and Earth by $\Delta r = |\boldsymbol{r}_E(t_E) - \boldsymbol{r}_M(t_M)|$ and the travel time of light by $\Delta t_{new} = \Delta r / c$.

Step 4: As long as $|\Delta t_{new} - \Delta t| > \varepsilon$, replace Δt with Δt_{new} and continue with Step 2; otherwise, stop this routine and the solution is $t_E = t_M + \Delta t_{new}$.

Furthermore, the minimum separation is found using a procedure with three points (separations α) as a function of time t, which are α_1, α_2, and α_3 at times t_1, t_2, and t_3. These points define a kind of hyperbolic function of the following form:

$$\alpha(t) = a \cdot \sqrt{(t-b)^2 + c^2} \quad \text{with} \quad b = t_2 + \frac{1}{2} \cdot \frac{(\alpha_2^2 - \alpha_1^2)(t_3 - t_2)^2 + (\alpha_3^2 - \alpha_2^2)(t_1 - t_2)^2}{(\alpha_2^2 - \alpha_1^2)(t_3 - t_2) + (\alpha_3^2 - \alpha_2^2)(t_1 - t_2)} \tag{43a, b}$$

The parameters a and c need not be calculated. Because t is given as large number *JDE*, the addition and subtraction of t_2 in Eq. (43b) is a trick to avoid numerical instability (like in “ringfit,” section 4.4.2). Next, the minimum at $t = b$ replaces the worst of the three previous points, which iteratively yields the nearest approach (subroutine “fitmin,” 2. method). Note that the transit calculations are partly performed in a different way than by J. Meeus [22]. Nevertheless, if the solar radius of 695,990 km, applied by Meeus, is also used in P4, the results are identical in almost all cases.

4.7.3 Position angles of transit

The position angles refer to the transiting planet, when it is in contact with the Sun's limb. The angles are measured from the y-axis (Fig. 14), which points to the celestial north pole. They correlate also with the apparent motion of the Sun due to the Earth's rotation. Jean Meeus provides a procedure for calculating the apparent positions of Mercury and Venus on the solar disk during the transit [22, pp. 14 ff.]. Unfortunately, for Mercury the method is available only for the years between 1600 and 2300 AD. Because we are interested in the year 3088 AD, another way must be found.

Let us assume an Earth reference system that is not rotating and independent from the orientation of the Earth axis (CRS, Celestial Reference System), and another system that is fixed to the Earth (TRS, Terrestrial Reference System). If $\boldsymbol{x}_{CRS}$ and $\boldsymbol{x}_{TRS}$ are two position vectors belonging to the same local point, but to the two different systems, the transformation between both vectors at a time t is given by [36, 37]

$$\boldsymbol{x}_{CRS} = \boldsymbol{P}(t)\cdot\boldsymbol{N}(t)\cdot\boldsymbol{U}(t)\cdot\boldsymbol{X}(t)\cdot\boldsymbol{Y}(t)\cdot\boldsymbol{x}_{TRS} \tag{44}$$

The matrices $\boldsymbol{P}$ and $\boldsymbol{N}$ take into account precession and nutation, $\boldsymbol{U}$ the rotation, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ the polar motion of the Earth. Although in our case the matrices $\boldsymbol{N}$, $\boldsymbol{U}$, $\boldsymbol{X}$, and $\boldsymbol{Y}$ can be neglected, the calculation is still not easy. However, instead of using explicitly the precession matrix $\boldsymbol{P}(t)$, the position angles are calculated with the following four steps:

1. The positions of the planets Mercury (or Venus) and Earth in Cartesian coordinates – calculated with VSOP87C (ecliptic of epoch) – are rotated around the x-axis by an angle, which is the obliquity of the ecliptic of that epoch. We have to take the x-axis because it connects the solar center with the Earth's position at the beginning of spring (vernal equinox). By this rotation, the x-y-plane becomes parallel to the plane of the Earth's equator. The obliquity of the ecliptic ε_e, which varies slightly in time, is taken from Axel D. Wittmann [38, p. 203]:

$$\varepsilon_e = 23.4458042^\circ - 0.856033^\circ \cdot \sin\left(0.015306\cdot\left(T + 0.50747\right)\right) \tag{45}$$

   The time T is measured in Julian centuries as in Eq. (7) and the argument of the sine function is given in radians. Other equations for ε_e with polynomials exist, but Eq. (45) has the advantage of having no “runaway effect” for large T [38]. Mathematically, the transformation is performed by using the rotational matrix of Eq. (73) in section 4.9.3.

2. Now, the new positions of Mercury or Venus and of the Sun are translated by the (negative) coordinates of the new Earth position. This means that the origin of the heliocentric coordinate system is shifted to the Earth's center and so becomes geocentric.

3. The new rectangular coordinates of Mercury (Venus) and the Sun are transformed into spherical coordinates.

4. From these geocentric coordinates, the position angle of Mercury − or accordingly Venus − with respect to the solar center is calculated by equations taken from André Danjon [39, p. 36] and Jean Meeus [22, p. 15], respectively. In the following, α_S and δ_S are the apparent right ascension and declination of the center of the Sun, and α_P and δ_P are the corresponding angles for the planet Mercury or Venus. With $\Delta\alpha = \alpha_P - \alpha_S$, $\Delta\delta = \delta_P - \delta_S$, and K being an auxiliary quantity we get

$$K = \frac{206264.8062}{1 + \sin^2\delta_S\cdot\tan\Delta\alpha\cdot\tan\left(\Delta\alpha/2\right)} \tag{46}$$

$$x = -K\cdot\left(1 - \tan\delta_S\cdot\sin\Delta\delta\right)\cdot\cos\delta_S\cdot\tan\Delta\alpha \tag{47}$$

$$y = K\cdot\left(\sin\Delta\delta + \sin\delta_S\cdot\cos\delta_S\cdot\tan\Delta\alpha\cdot\tan\left(\Delta\alpha/2\right)\right) \tag{48}$$

   The constant 206264.8062 is the number of arc seconds in one radian. The zero position of right ascension is not relevant because declination and differences of right ascension are unaffected. The quantities x and y are the rectangular coordinates of the planet given in arc seconds. Finally, the position angle P, measured from the y-axis (Fig. 14), is

$$P = \arctan\left(\frac{-x}{y}\right) \tag{49}$$

With $\cos P$ having the same sign as y, we get P in the correct quadrant [22]. In the P4 program this is realized as follows: If we have $y \cdot \cos P < 0$, then P is replaced by $P + 180°$. The transit in 3088 AD (Fig. 14) is not a central one, but it is the first one of a new transit series. Due to the convention, taken from the "NASA Eclipse Web Site," this series has the number 20. It comprises nine transits and will last from 3088 to the year 3456.

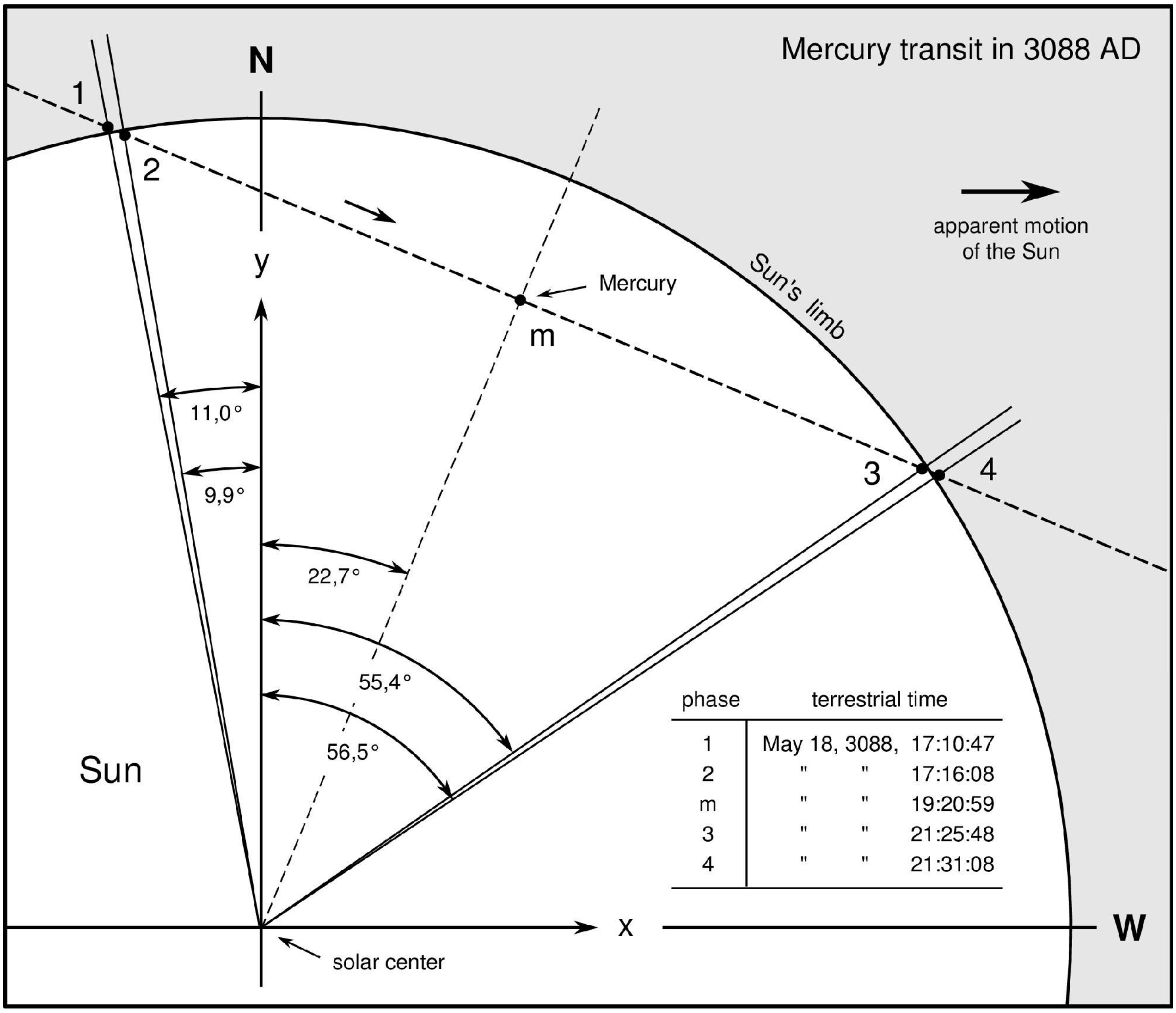


Figure 14: *True-to-scale representation of the Mercury transit in 3088 AD with position angles. If calculating the data with the previously used solar radius 695,990 km [22] and not with the current value of (695,508 ± 26) km [35], the deviations are maximal 0.06° for the position angles and 18 s for the times of day. These differences are rather small.*

Several iterative search algorithms are used in the P4 program. The contact points 1 to 4 are determined with the subroutines "ringfit" and "secant" in combination with the "fixed point" algorithm (section 4.7.2, subroutines "vsop1tr" and "vsop2tr"), whereas the nearest approach is calculated by a special minimum search (subroutine "fitmin," 2. method). Other methods are those of Newton and Raphson, used to solve Kepler's equation (subroutine "vsop3"), the procedure to minimize the angular range of a planetary conjunction (subroutine "fitmin," 1. method), and FITEX, the multi-parameter fit program used, e.g., to determine the "Sun position" (last four subroutines in P4).

A few remarks should be done about the characteristics of grazing transits. In principle, there are three different kinds of geocentric grazing transits. All calculations in the P4 program are done with the assumption that the observer is placed at the center of the Earth. This yields the geocentric transit phases (times in the tables). In case of a geocentric grazing transit, only three transit phases are provided: the two outer contact points and the minimum separation because the planet never gets completely on the solar disk. If this transit can be seen as a full transit from other parts of the Earth then it is also named a "partial transit." The second possibility is that from the geocentric position, the planet does not touch the Sun's limb, but from other points of the Earth, we have a grazing transit. In this case, only one transit phase can be calculated, which is the nearest separation between planet and Sun. The third possibility means that it looks like a full transit from the geocentric position but from other parts of the Earth it is a grazing transit. In this case, we have five transit phases, like a full transit, but actually from parts of the Earth, it is a grazing (partial) transit. In the computed tables, these three cases – marked with "m" for Mercury and "v" for Venus – can be distinguished easily. They have three, one, or five transit phases, respectively.

4.7.4 Transit series

The transits of Mercury or Venus in front of the Sun can be combined in the so-called transit series. Different ways of combinations are possible and are described in detail in [22, pp. 7–13]. The main patterns are successive transits of Mercury every 46 years and of Venus every 243 years. Each series has a number and different series are serially numbered according to their first appearance. The series of Mercury, starting at May 18, 3088 has the number 20 (see last column in the table of section 3.4.6). The corresponding paths of Mercury along the Sun's disk are provided in Fig. 15. The figure is drawn on the basis of the position angles, calculated with P4 (VSOP87).

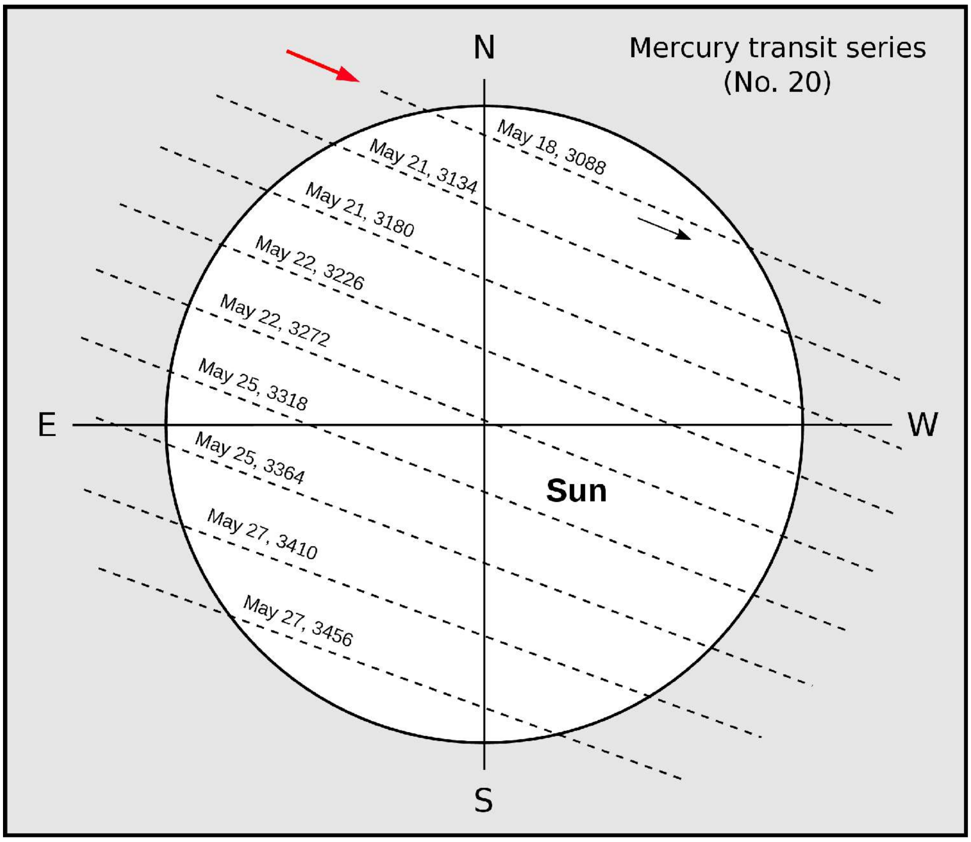


Figure 15: *The complete transit series of Mercury, having the number 20 (according to the serial numbers on the website of the NASA/Goddard Space Flight Center – see links in section 3.4.6). The transit series begins in May 18, 3088.*

Now, if a new transit is found by the P4 program, the question is: How do we get the corresponding serial number? For Mercury, the time differences between transits in the same series are multiples of 46 years. In the P4 program the date (JDE) of each first transit of a new series is stored. Therefore, the date of the new transit is compared with these transit dates, already stored. If the time difference between the new and one of the stored transit dates is a multiple of 46 years the new transit has the same serial number as the stored transit date. If there is no connection to a preceding transit, the new one gets a new serial number. To check for the multiple of 46 years, the function "modulus" (mod) can be used. Let J_{prev} and J_{new} be the decimal years of one of the stored older transits and the new transit. Then the corresponding relation for Mercury is:

$$(J_{new} - J_{prev}) \bmod 46 < \varepsilon \qquad (50)$$

Here, ε is a small time period like, for example, 0.03 years. This description is a little bit simplified. Instead of using years in the P4 program, the periods of 46 and 243 years are provided in Julian Days. For Mercury, the averaged time interval is Δt_M = 16802.200 Julian Days and for Venus it is Δt_V = 88756.137 Julian Days. Actually, after a longer time period some of the already finished transit series can show up again. Due to the convention, these transits get a new serial number.

Another example is given here independently of the pyramids. The recent Venus transit in the year 2012 has the number 5. If we follow this series No. 5 into the future, we find that approximately in the years between 5000 and 7200 a fantastic sequence of about 8 successive central Venus transits will occur within this series. The corresponding Venus passages are represented in Fig. 16. The transits are shown from the beginning only up to the central transits. During later transits of this series, Venus will pass also in front of the northern half of the Sun so that Fig. 16 would become confusing. Additionally, the precision of VSOP87 decreases slightly in this remote future.

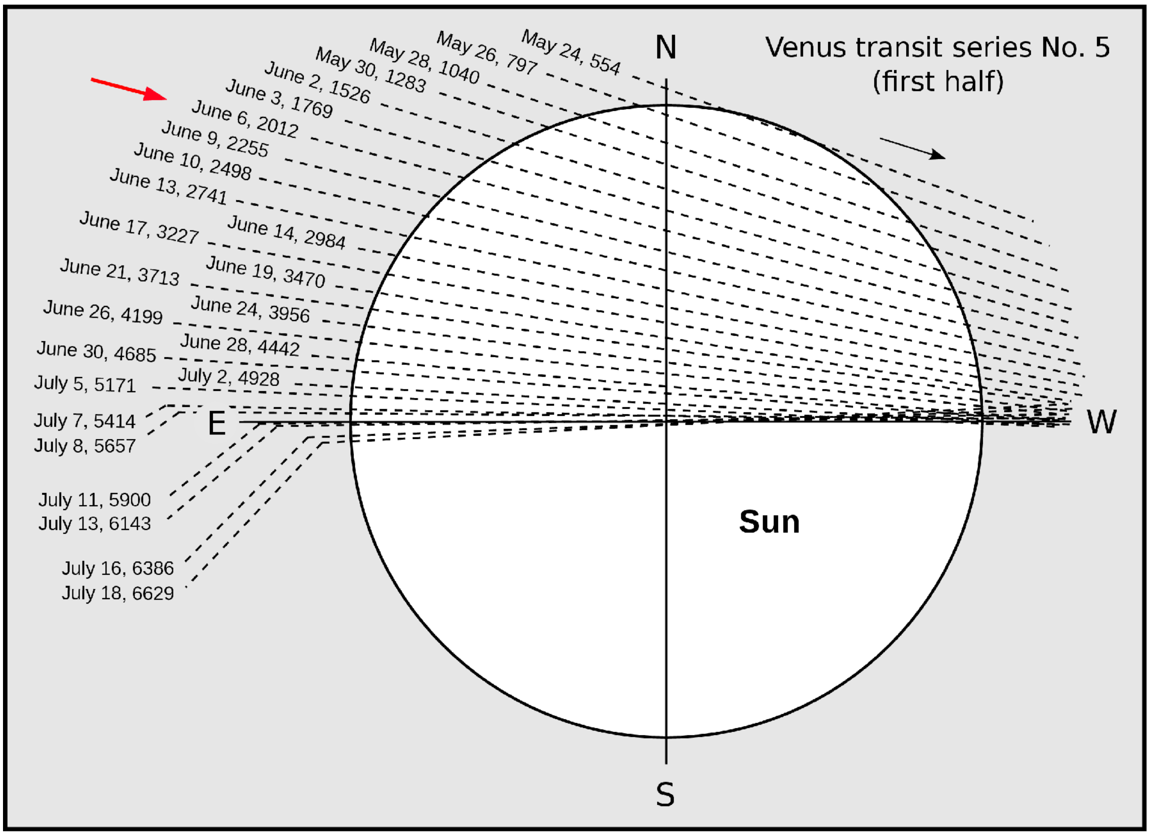


Figure 16: *Venus transits, first half of series No. 5 (number convention like in Fig 15, only Gregorian calendar), from the beginning in the year 554 up to the central transits between the years 5000 and 7200.*

More details could be discussed, like for instance special circumstances that show up when computing conjunctions or methods to improve the processing speed, but here only the main points are given. In principle, all details can be found in the source code (appendix).

4.8 Universal time

The terrestrial (dynamical) time (TT) is a linear time scale with constant lengths of the days. It is appropriate for astronomical purposes in order to handle large time spans accurately. The universal time (UT) takes into account the slowing down of the Earth due to tide friction. It means that the length of a day is increasing very slowly. Because the "second" is the basic time constant, a leap second is introduced occasionally in order to keep the time of day in phase with the Earth's rotation. These leap seconds are applied in UTC (Coordinated Universal Time), which means that the difference of UTC to TT changes always discontinuously by 1 second. The difference between UT and TT changes continuously but not uniformly because the deceleration of the Earth varies slightly from time to time.

Because the Earth's deceleration for times far in the past or in the future is not precisely known, UT (and also UTC) includes some uncertainties in these times. For such time periods, UT can only be extrapolated. TT is by definition a precise measure of time and is used in astronomy for long time spans when the Earth's rotation is not relevant. The terrestrial time TT is equivalent to JDE and is used in the VSOP87 theory. If the time of day is important, like for example, for historical events on Earth, UT or UTC should be used. The P4 program allows for a conversion from TT to UT. The equations, provided further down, are used to calculate the time difference:

$$\Delta T = TT - UT \tag{51}$$

These equations are taken from the "NASA Eclipse Web Site," Polynomial Expressions for Delta-T, and are reproduced here because they seem to be available only on the Internet. The polynomials up to 7th degree were created by Fred Espenak and Jean Meeus, based on the works of Morrison/ Stephenson [40] and Stephenson/Houlden [41].

To apply the equations, a decimal year J is required. Espenak and Meeus provide the following equation: J = year + (month – 0.5)/12. For consistency with references [5] and [13], "J" (Jahr) instead of "y" (year) is used. The maximum error would be 0.5 months and the average error about 8 days, being sufficiently small. However, we use the decimal year, given by Eqs. (68) and (69) in section 4.9.1. The average error of 0.5 days is even smaller and the application is easier because in these equations the year is given directly as a function of *JDE.*

Before the year -500 (astronomical counting), which is 501 BC, and from now into the future, ΔT has to be extrapolated based on the reasonable assumption that the Earth's rotation decelerates more or less constantly. The polynomials are valid only within the corresponding time periods. The result ΔT is given in seconds (Fred Espenak, Jean Meeus [23]):

$J \le -500$:
$$\Delta T = -20 + 32\,u^2 \quad \text{and} \quad u = \frac{J-1820}{100} \tag{52}$$

$-500 < J \le 500$: $u = \frac{J}{100}$
$$\Delta T = 10583.6 - 1014.41\,u + 33.78311\,u^2 - 5.952053\,u^3 - 0.1798452\,u^4 + 0.022174192\,u^5 + 0.0090316521\,u^6 \tag{53}$$

$500 < J \le 1600$: $u = \frac{J-1000}{100}$
$$\Delta T = 1574.2 - 556.01\,u + 71.23472\,u^2 + 0.319781\,u^3 - 0.8503463\,u^4 - 0.005050998\,u^5 + 0.0083572073\,u^6 \tag{54}$$

$1600 < J \le 1700$: $t = J - 1600$
$$\Delta T = 120 - 0.9808\,t - 0.01532\,t^2 + \frac{t^3}{7129} \tag{55}$$

$1700 < J \leq 1800:$ $t = J - 1700$	$\Delta T = 8.83 + 0.1603\,t - 0.0059285\,t^2 + 0.00013336\,t^3 - \dfrac{t^4}{1174000}$	(56)
$1800 < J \leq 1860:$ $t = J - 1800$	$\Delta T = 13.72 - 0.332447\,t + 0.0068612\,t^2 + 0.0041116\,t^3 - 0.00037436\,t^4 + 0.0000121272\,t^5 - 0.0000001699\,t^6 + 0.000000000875\,t^7$	(57)
$1860 < J \leq 1900:$ $t = J - 1860$	$\Delta T = 7.62 + 0.5737\,t - 0.251754\,t^2 + 0.01680668\,t^3 - 0.0004473624\,t^4 + \dfrac{t^5}{233174}$	(58)
$1900 < J \leq 1920:$ $t = J - 1900$	$\Delta T = -2.79 + 1.494119\,t - 0.0598939\,t^2 + 0.0061966\,t^3 - 0.000197\,t^4$	(59)
$1920 < J \leq 1941:$ $t = J - 1920$	$\Delta T = 21.20 + 0.84493\,t - 0.076100\,t^2 + 0.0020936\,t^3$	(60)
$1941 < J \leq 1961:$ $t = J - 1950$	$\Delta T = 29.07 + 0.407\,t - \dfrac{t^2}{233} + \dfrac{t^3}{2547}$	(61)
$1961 < J \leq 1986:$ $t = J - 1975$	$\Delta T = 45.45 + 1.067\,t - \dfrac{t^2}{260} - \dfrac{t^3}{718}$	(62)
$1986 < J \leq 2005:$ $t = J - 2000$	$\Delta T = 63.86 + 0.3345\,t - 0.060374\,t^2 + 0.0017275\,\mathrm{t}^3 + 0.000651814\,t^4 + 0.00002373599\,t^5$	(63)
$2005 < J \leq 2050:$ $t = J - 2000$	$\Delta T = 62.92 + 0.32217\,t + 0.005589\,t^2$	(64)
$2050 < J \leq 2150:$	$\Delta T = -20 + 32 \cdot \left(\dfrac{J-1820}{100}\right)^2 - 0.5628 \cdot (2150 - J)$	(65)
$J > 2150:$	$\Delta T = -20 + 32\,u^2$ and $u = \dfrac{J-1820}{100}$	(66)

The universal time is now obtained by UT = TT – ΔT. Note that Eqs. (52) and (66) are identical. All equations are implemented in the calendar program DATUM-2, too. The uncertainties in ΔT are also taken from the NASA website [24] and fitted by polynomials for the use in DATUM-2 (for details see [13, app. A5]). To get an idea, some results of ΔT with errors (±) are provided in the following:

$J = -2000:$ ΔT = (778 ± 62) minutes

$J = 2000:$ ΔT = (63.9 ± 0.1) seconds

$J = 3000:$ ΔT = (74 ± 31) minutes

$J = 20\,000:$ ΔT = (294 ± 97) hours ≈ (12 ± 4) days

4.9 Computational changes from P3 to P4

When reproducing the results in the tables of book 1 [5] with the P4 program, in some cases slight numerical changes can be found. The astronomical calculations, based on the VSOP87 theory, are unchanged. This includes the dates, based on the Julian Ephemeris Day, all kinds of positions like the “Sun position” at the Giza plateau, and other astronomical quantities. However, some other calculations are improved and the changes − compared to the previous version P3 − are provided in the following. New additional options and all new features of P4 compared to P3 are listed in the program header of the P4 source code p4.f95 (appendix) and are also included in section 3.3.

4.9.1 Decimal year

In some tables the date is not given as Julian Day but as a decimal year number. This is just intended to assist the reader in knowing, for example, what *JDE* = 2456282.5 means. If the corresponding decimal year *J* = 2012.97 is given, it becomes clear that the given date is somewhere at the end of the year 2012 AD. In the first book [5, p. 315], the decimal year *J* was approximated by the following linear function of the Julian Day:

$$J = \frac{JDE}{365.248} - 4711.9986 \tag{67}$$

When comparing with the calendar date, this equation has an error of less than 12 days for the time interval 11,000 BC to 4000 AD. Before and after this period, when going further into the past or into the future, the error is increasing linearly with respect to ΔT. For the year 10,000 AD, the deviation from the calendar date is about 34 days, and for the year 100,000 AD the error is approximately 1.5 years, which can be checked easily with the DATUM-2 program. The reason for these discrepancies is the existence of two different calendars, the Julian and the Gregorian calendar, with a calendar reform in the year 1582 AD. This means that two different linear functions, being linear on a large scale, are approximated by one linear function in Eq. (67).

The simple solution of this problem is to use *two* linear functions instead of one. Thus, for the Julian calendar and the Gregorian calendar, respectively, we have

$$(0 \le JDE < 2299160.5) \qquad J = \frac{JDE}{365.25} - 4712.0 \tag{68}$$

$$(JDE < 0 \text{ or } 2299160.5 \le JDE) \qquad J = \frac{JDE - 2451545.0}{365.2425} + 2000.0 \tag{69}$$

Here, the decimal year, based on the Julian calendar, is only used for the time period $0 \le JDE < 2299160.5$, which are the years between 4712 BC and 1582 AD. The upper limit is evident because of the calendar reform. One reason for the lower limit is that the Julian calendar gets completely out of phase with the seasons before 4712 BC, and another reason is that in those years no historical events exist so that it would not make any sense to use the Julian calendar. (Note, that the Gregorian calendar is a substantial improvement, but it still needs a correction of one day about every 4000 years to stay in phase with the seasons.)

Now, the average deviation between the decimal year and the calendar date is around ± 0.5 days for all times. It does not matter how far we go into the past or future. Anyway, for our purpose the difference between Eq. (67) and the system of Eqs. (68) and (69) is relatively small, like for example, for the moment of minimum separation of Mercury transit during the planetary constellation 12. Equation (67), applied in the first book, yields *J* = 3088.365, whereas with Eq. (69) we get *J* = 3088.379. As mentioned previously, all astronomical computations with the VSOP87 theory are unaffected, since they are based on *JDE* and not on *J*.

The period of 3800 years

In this context the period of about 3800 years, described in [5, pp. 132, 136], needs an additional comment. At the end of a period of 3800 years and 1 month the planets Mercury, Venus, and Earth have nearly the same positions like at the beginning, because for all three planets this period is almost equal to an integer number of siderial orbital periods. (This helped to estimate the range of validity of vsop3 in Fig. 10.) The time interval is about 1387980.4 Julian Days. If we divide this number by the 365.25 days of a Julian year, we get 3800.0832 years, which is almost equal to 3800 years and 1 month. Division by the 365.2425 days of one Gregorian year yields 3800.1613 years, equal to 3800 years and 2 months. So, the period of 3800 years and 1 month, discussed in [5], is primarily valid for the Julian calendar. On the basis of the Gregorian calendar, after the year 1582 AD, the period lasts 3800 years and 2 months. Although the two periods differ by 1 month because of the different length of the years, the physical time period is exactly the same.

4.9.2 Position tolerance

When the planetary constellation of Mercury, Venus, and Earth is fitted to the pyramid positions (or chamber positions), an accuracy of this fit in percent is given by the relative error F_{pos}, F'_{pos}, or F''_{pos}, respectively. ("*F*" is related to the German word "Fehler," meaning error, fault or mistake.) In the following we use only F_{pos}, although the equations are also valid for F'_{pos} and F''_{pos}, which are each based on a different geometrical approach. In order to get a position error dr of the calculated "Sun position" in meters (in [5] also called Δs), the length of the position vector of the "Sun position" $r_S = |\boldsymbol{r}_S|$ is multiplied by F_{pos}, which is $\mathrm{d}r = r_S \cdot F_{pos}$ [5, e.g., Tab. 17, p. 149]. The origin of the corresponding coordinate system is placed in the center of the Mykerinos Pyramid (pyramid positions) or in the base of the Cheops Pyramid (chamber positions), which is an arbitrary choice in both cases. Although the resulting position errors are quite reasonable, it is more convenient to measure the position vectors from the common center of the three pyramids and from the common center of the three chambers. The coordinates of these two centers (position vector $\boldsymbol{r}_{CM}$) are just the arithmetic average of the corresponding rectangular position coordinates of the three pyramids or three chambers. Let r_a be the average distance of the three pyramids (chambers) from this center and r_{Sun} the distance of the "Sun position" from this center, given by

$$r_{Sun} = |\boldsymbol{r}_S - \boldsymbol{r}_{CM}| \tag{70}$$

Then, the position error of the "Sun position" (dr) in meters is calculated by the following equations:

For $r_{Sun} > r_a$:

$$\mathrm{d}r = r_{Sun} \cdot F_{pos} \tag{71}$$

for $r_{Sun} \le r_a$:

$$\mathrm{d}r = \frac{r_a}{2} \left(\left(\frac{r_{Sun}}{r_a} \right)^2 + 1 \right) \cdot F_{pos} \tag{72}$$

As F_{pos} is given in percent, it must be divided by 100 before being inserted into Eqs. (71) and (72). If the "Sun position" is located near to the common center of the three chambers, meaning that r_{Sun} is almost zero, then the relative position error dr, calculated with Eq. (71), is nearly zero, too. This would not be very reasonable and is avoided by using the parabolic function in Eq. (72).

Eqs. (70) to (72) are used for the positions of "Sun" and of all "planets" – replacing r_{Sun} accordingly – except Mercury, Venus, and Earth. For the last-mentioned planets, the deviations between transformed planetary positions and pyramid (chamber) positions can be determined exactly by calculating the corresponding Euclidean distances. These values are marked with "<" (see quick start options 3 and 250, sections 3.4.3 and 3.4.8). Finally, the previous equations are analogously used to determine the uncertainty of the Mercury aphelion position for the constellations 13 and 14.

4.9.3 Algebraic sign of X_5

One method for calculating the “Sun position” in 3 dimensions is to use coordinate transformations and the least squares fit program “FITEX,” in which the planetary positions are fitted to the pyramid or chamber positions by adjusting the seven parameters X_1 to X_7. The orientation is adapted by using the rotational matrix $\boldsymbol{R}(X_4, X_5, X_6)$ of Eq. (23). With X_4, X_5, and X_6 being the Euler angles, $\boldsymbol{R}$ is the product of three matrices $\boldsymbol{D}_z(X_6) \cdot \boldsymbol{D}_x(X_5) \cdot \boldsymbol{D}_z(X_4)$; see also [5, pp. 335 ff.]. In the following, the algebraic sign of X_5 is discussed. Let α be the rotational angle; then, for example, a rotation around the x-axis is given by the matrix

$$\boldsymbol{D}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix} \tag{73}$$

(Note: It seems that by accident all rotations in Ref. [5] are given by the transposed (inverse) matrices compared to the normal convention. Because this is only a question of agreement and because it does not change the results – except for the sign of the rotational angles – we kept this allocation here.) The angle X_5 is the tilt angle between the planes of the Earth's surface and of the transformed Earth's orbit [5, p. 345]. Both planes are shown schematically in Fig. 2. By geometric considerations it can be established that, if X_5 is a solution matching the planetary positions with the pyramid (chamber) positions, $-X_5$ is also a solution. In the latter case, the angles X_4 and X_6 have to be replaced by $X_4 \pm \pi$ and $X_6 \pm \pi$, respectively, where the plus and minus signs can be chosen arbitrarily and independently for both quantities. So, for rotations in 3 dimensions we get the following matrix identity:

$$\boldsymbol{D}_z(X_6) \cdot \boldsymbol{D}_x(X_5) \cdot \boldsymbol{D}_z(X_4) = \boldsymbol{D}_z(X_6 \pm \pi) \cdot \boldsymbol{D}_x(-X_5) \cdot \boldsymbol{D}_z(X_4 \pm \pi) \tag{74}$$

This can be shown easily with the matrix in Eq. (23). By modifying the angles correspondingly, all changes of algebraic signs of the trigonometric functions cancel each other, that the matrix remains the same. It follows that the sign of X_5 has no meaning if X_5 is given without X_4 and X_6. Therefore, most tables in the second book [13] list only the absolute values of X_5. On the other hand, the P4 program always yields the actually found algebraic sign of X_5. Eq. (74) can be demonstrated, for example, with a postcard. After defining x-, y-, and z-axis as well as X_4, X_5, and X_6, the reader gets the same result by rotating the postcard manually by using either the given angles on the left or on the right side of Eq. (74).

4.9.4 Date of constellations 13 and 14

For the constellations 13 and 14 the date is not fixed to the planetary passage through aphelion or perihelion. Instead, the exact point of time is found by manually minimizing the relative error F''_{pos}. For the results in the first book [5], when using the P3 program, the Julian days (*JDE*) were rounded three digits after the decimal point. In the P4 program, the minimization of F''_{pos} is done automatically and the *JDE* results are accurate for about five digits after the decimal point. In order to achieve consistency between different calculations within the second book, the dates are not rounded like before. Thus, when reproducing the results in the first book with the p4 program, there are slight differences in the planetary coordinates concerning the constellations 13 and 14. Nevertheless, these tiny differences are not important. They are mentioned here so that the user of the P4 program knows (if comparing the results with [5]) where these deviations come from.

4.10 Further specific features

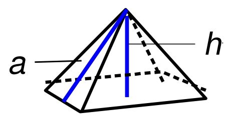
In the following, two additional discoveries with respect to constellation 12 (3088 AD) are presented in sections 4.10.2 and 4.10.3. They are not directly related to the use of the P4 program, but they do support the general findings. They are also mentioned to clarify that the overall picture of the planetary correlation is even more complex than originally expected. In sections 4.10.4 and 4.10.5, we learn more about “Sun positions” and “secret chambers,” but before these aspects are explained, some information is given concerning mathematical speculations.

4.10.1 Matching coefficients

If archaeological facts, like measured data, are mixed with mathematical speculations to show certain relations, sometimes criticism arises, saying that with such “mathematical playing around” anything can be proven. Truthfully, under special circumstances, this criticism is justified. However, it is not always true. To clarify, we will specify what “special circumstances” means in this context by using two examples.

In addition to classical archaeology, two relations that define the size of the Cheops Pyramid have been found in the literature. First, the distance from Earth to the Sun (1 AU = 149.6 million km) is said to be 1 billion times larger than the height of the Cheops Pyramid (146.59 m [10, vol. IV, p. 1227]). Secondly, the height of the triangular side of the Cheops Pyramid, being the distance from the base line to the top of the pyramid (186.43 m), multiplied by 600, should be the distance corresponding to 1° difference in longitude at the equator (111.32 km). The relative error of the first equation varies up to 3.6 % because the distance between Earth and the Sun is not constant, due to the elliptical shape of the Earth's orbit. The accuracy of the second relation is about 0.5 % which is somewhat better. The reader can verify this easily. Nevertheless, both relations have a serious disadvantage: They both contain an arbitrary factor, namely 1 billion (1,000,000,000) and 600. The problem is that with such factors, or better “matching coefficients,” just about anything can be proven. The corresponding equations are given in the following together with a small sketch. The crossing out of both equations stresses that both of them are meaningless. The reason is given in the following.

1. Distance Earth – Sun: $r = 1{,}000{,}000{,}000 \cdot h_{Cheops}$
2. Distance for 1° difference in longitude: $L(1°) = 600 \cdot a$



Let us take two arbitrary quantities (like the height of the Eiffel Tower and the distance between London and New York) and let us allow matching coefficients, consisting of one digit (1 to 9) and an arbitrary number of zeros (e.g., 100, 4000, 70, 300,000, etc.). In this case, it can be easily shown that with the corresponding “matching factor,” an average accuracy of about 10 % can be achieved. We illustrate this with an example. Let us assume that the ratio between two quantities is exactly 550. Thus, we need a factor of 550 to get an equation, which is perfectly valid. According to the previous assumption concerning matching coefficients, the nearest available factors are 500 and 600. The relative error of an equation, using 500 or 600, would be approximately 10 %. A mathematically more detailed discussion is given in [5, pp. 64 ff.]. Now, a deviation of 10 % is more than the errors in the above two equations, so we could misleadingly assume that both equations are still significant. However, this is not the case.

The given pyramid has five characteristic lengths: the height *h*, the height of the side face *a*, the base length, the diagonal in the base area, and the distance from a corner to the top of the pyramid. If we take, for example, five astronomical lengths for comparison, like the distance from Earth to the Moon, the circumference of the Earth, etc., then we have 25 different combinations with the five lengths in the pyramid. The 25 corresponding equations do not have an accuracy of 10 % each, but some of them have less and some of them have higher accuracy, meaning statistical scattering. Now, it can be shown mathematically that, on average, at least one of these combinations has an error of less than 1 %! It means that we have to look only for the smallest error of all

25 relations to obtain an accuracy of better than 1 %. It follows that all such equations containing a matching coefficient (like 600 or 1 billion) have no significance! With such factors we can prove anything. We could even easily bridge several orders of magnitude. In short, equations with such a "matching coefficient" are irrelevant! The same can be achieved if more complicated equations are used. This includes squares (like x^2, where x might be any quantity), other mathematical powers (x^3, x^4, …), square roots, constants like π, etc., or even two or more matching coefficients instead of one.

Another important criterion is that the physical units like meter, kilogram, etc. must match. For example, after combining some archaeological quantities in an equation, someone gets the number "2.99" and would claim: "Here, we have the speed of light." In actuality, the speed of light is $2.9979 \cdot 10^8$ m/s. The relative error of the pure digits is about 0.3 %, which is not bad, but the physical unit of velocity like "m/s" is missing. Furthermore, if using the m/s unit, the order of magnitude is wrong by a factor 100,000,000. The last factor is a matching coefficient, not even being mentioned. Thus, the quantity 2.99 has absolutely no meaning with respect to the velocity of light.

We keep in mind that "matching coefficients" are not allowed. However, without such factors it is almost impossible to find an equation that relates an arbitrary quantity with fundamental physical or astronomical constants. So, what about the three equations (1) to (3) of the planetary correlation? All of them are simple and of the same kind: the rule of proportion. No matching coefficients are used and the physical units are correct. Moreover, the physical quantities create an overall picture, which makes sense. So, they are not of the category "matching coefficients," explained previously. The three equations and especially Eq. (1) are analyzed in detail in Refs. [5, 13]. In the following, two other astonishing aspects are described, which support the planetary correlation.

4.10.2 Obliquity of the ecliptic

Figure 2 in the introduction shows how the transformed planetary orbits are tilted against the Earth's surface. The tilting angle between the transformed ecliptic plane (plane of the Earth's orbit) and the Earth's surface is 24.47° (= X_5; see X_5 in section 3.4.3). It just comes out by using the VSOP87 theory in the P4 program, in which X_5 is one of the parameters X_1 to X_7, characterizing the coordinate transformation from the positions of the planets to those of the pyramids.

Why didn't the master builders construct these two planes coplanar, but instead tilted them against each other? The answer is simple: the obliquity of the ecliptic is about 23.45°, which is the angle between the plane of the equator and the plane of the Earth's orbit (ecliptic plane). By the way, this obliquity is the reason why we have four seasons per year.

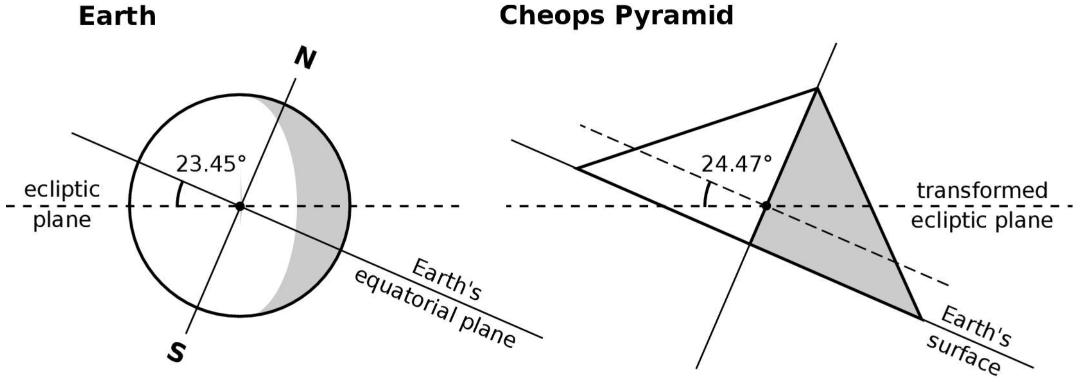


Figure 17: *Correlation between Earth and Cheops Pyramid – as seen from southwest direction – with respect to the ecliptic plane (see Fig. 2 and for 24.47° see X_5 in the table of section 3.4.3).*

The given correlation means that the angle between the plane of the Earth's equator and the ecliptic plane correlates with the angle between the Earth's surface and the ecliptic plane after coordinate transformation. The difference of 1° is small and can probably be explained (compare X_5 in the table of section 3.4.4). An illustration of the correlation is given in Fig. 17 (see also Fig. 2). This is perfectly in keeping with the correlation between pyramids and planets, and can explain why the transformed planetary orbits in Fig. 2 are considerably tilted against the Earth's surface in Giza.

4.10.3 The riddle of midwinter

Figure 18 shows that inside the Cheops Pyramid the "Sun position" is located south of the subterranean chamber. If we look from the chamber to the "Sun position," the direction is about 34.0° upward from the horizontal direction. (With Δx = 16.40 m and Δy = 11.05 m being the differences of the respective coordinates of "Mercury" and "Sun position" – see table in section 3.4.8 – the angle is calculated by $\arctan(\Delta y/\Delta x)$ = 33.97°.) The real Sun has the highest position above the horizon in the south at noontime. The question is: Is it possible that the real Sun stands in the same direction of 34° above the horizon?

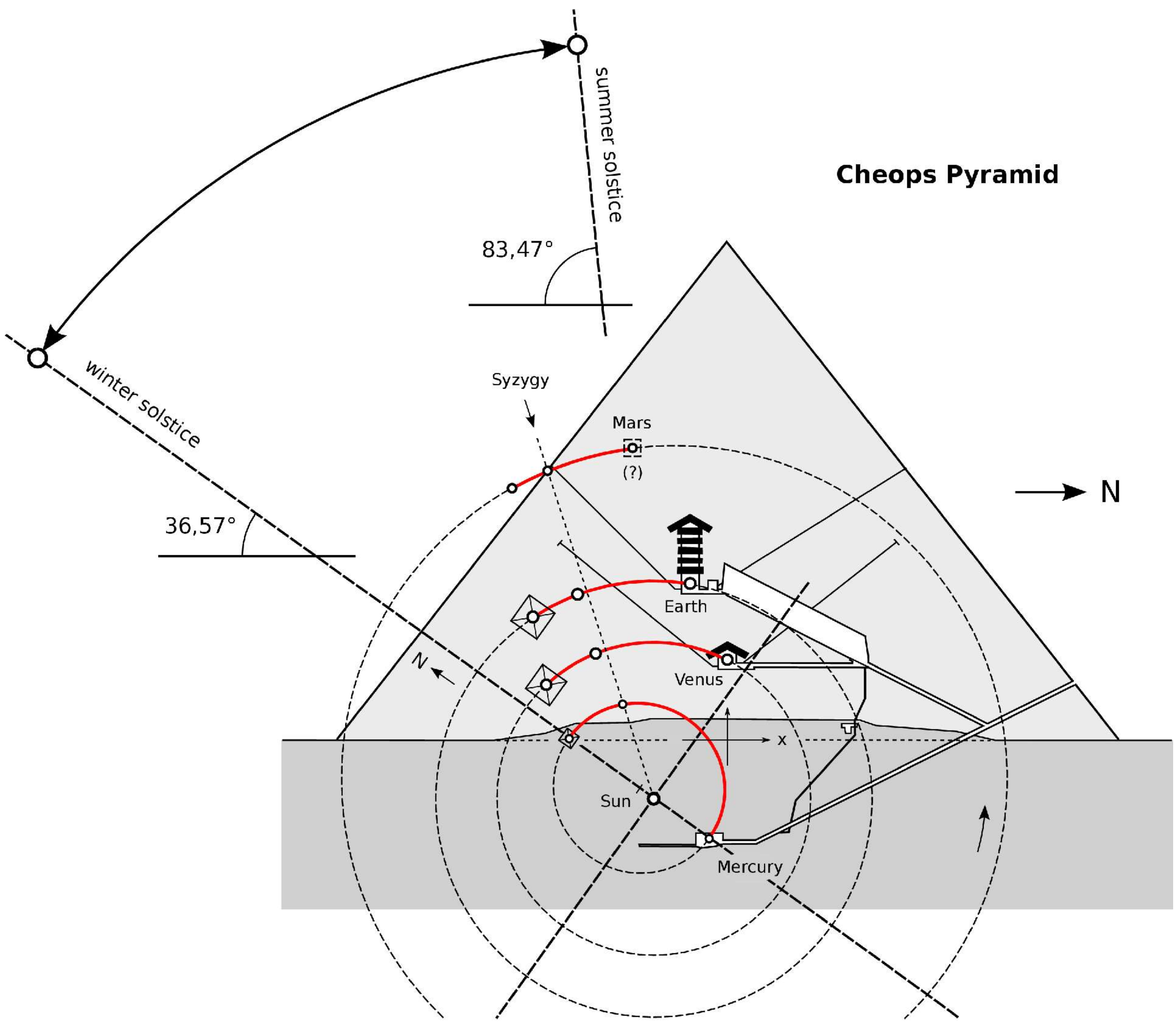


Figure 18: *Cross section of the Cheops Pyramid (3088 AD) with highest Sun position in midwinter and midsummer.*

The highest daily position of the Sun is dependent on the time of year. At summer the maximum angle above the horizon at Giza is 83.47°. The date is called midsummer or summer solstice. In winter, the lowest angle of the Sun in Giza at noon is 36.57°. Figure 18 provides the geometrical arrangement. It shows that if looking from the “Mercury position” in the subterranean chamber to the “Sun position” in midwinter, the real Sun stands almost in the same direction. The angular difference of 2.6° is small but not very small. For the astronomical comparison with the planetary positions, the chamber positions were assumed to be in the spatial middle of each chamber.

In order to find a remedy for the angular discrepancy of 2.6°, the chamber positions were slightly varied, and an interesting phenomenon was observed. If using the middle of the west walls of each chamber, the corresponding angle of the “Sun position” in the pyramid becomes 36.55°, which is almost exactly the midwinter position of the Sun above the horizon. Actually, an argument exists to use the middle of the west walls: The east walls of the three chambers are all placed in the same vertical plane (see Fig. 3), oriented in north−south direction. So, the x- and y-coordinates of the chambers are well defined with regard to this plane (Fig. 18). Now, all chambers extend to the west, but each one at a different length. This situation is similar to a histogram with three columns of different heights. Could it be that the pyramid builders exactly defined the coordinates of the planets in all 3 dimensions in this way?

Nevertheless, the “middle of west walls” option also has some disadvantages compared to the “spatial middle of chambers” option. The overall accuracy of the comparison between pyramids and planets is 2.2 % for the “west walls” option instead of 0.6 % for the “spatial middle” option (section 3.4.8). Furthermore, the angle between the vertical x-y-plane in the pyramid and the transformed ecliptic is 18.4° for the “west walls” option compared to 4.2° for the “spatial middle” option (see X_5 in section 3.4.8). So, as the “midwinter angle” becomes more accurate with the west walls, two other parameters become worse. Perhaps this problem can be answered one day, but for now it remains as open as an unsolved riddle – until further notice.

4.10.4 “Sun position” and concrete platform

In Fig. 12 we see the “Sun position,” located approximately 670 meters south from the center of the Mykerinos Pyramid. This position is defined precisely by the planetary positions in the year 3088 with an uncertainty of about 1 meter if the VSOP87 theory with 3-dimensional coordinate transformation is applied. The question is: Do we find anything special at this location? The answer is “yes”.

In the year 2003, I visited this spot and found there a platform of concrete. This stage was definitely not from ancient times, but from modern times. The center of the platform was located at latitude 29° 57.9898' north and longitude 31° 7.6842' east. The numbers are an average of a few measurements, performed with a GPS receiver (Garmin eTrex Summit). The theoretical “Sun position,” calculated with VSOP87, is latitude 29° 57.9905' north and longitude 31° 7.6813' east (section 3.4.9). At that time the platform had dimensions of about $25 \cdot 50\ m^2$ and was oriented with its long middle axis towards the center of the Chefren Pyramid. The difference between the measured and the theoretical position is only 5 meters. One year later, the whole platform was covered with half meter deposit of sand, which had been put there artificially and not by wind transportation. Today in 2014, the platform is again mostly free of sand. Meanwhile, its shape has been changed and can easily be seen in Google Earth or Google maps (satellite view).

What is the purpose of this platform? It is located in the desert, surrounded by sand and hilly ground. There is no paved vehicle access, and therefore it cannot be used as a parking area, although it looks like that. The question is: Is the remarkable coincidence of the platform position and the “Sun position” accidental or not? Of course, it can be an accident, but the probability for it seems low. The calculation of the “Sun position” yields a vertical coordinate of 272.36 m above the base level of the Mykerinos Pyramid. Thus, the “Sun position” is placed more than 250 m above the ground. So, if someone digs beneath that platform he probably will not find any treasure, except sand.

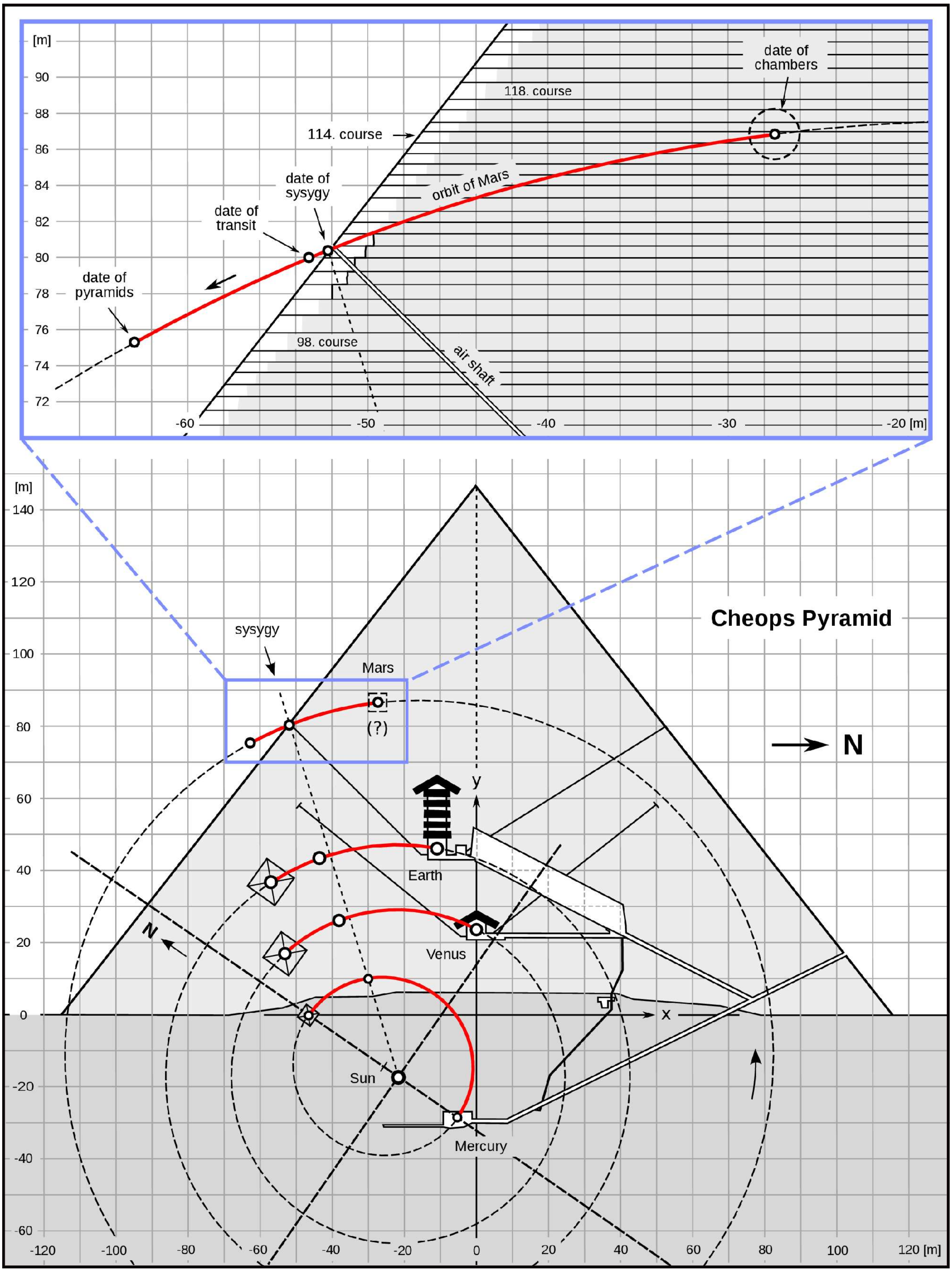


***Figure 19:** Cross section of the original state of the Great Pyramid with details of the "Mars position" and its environment (upper inset) during the astronomical events in the year 3088 AD. The levels of the courses were measured by W. M. F. Petrie [6, Map VIII]. The shape of the blocks of stone around the opening of the southern air shaft is taken from a drawing of Maragioglio and Rinaldi [9, part IV, map 2, Fig. 2]. This drawing, where we find also the numbers of the courses, was published in 1965. Today, in the year 2014 some more blocks have been removed around the mouth of the air shaft. So, the reader can compare the state of 1965 with the situation at present.*

4.10.5 “Secret chambers”

A better chance for a successful search is given in the Cheops Pyramid. Beside the “Sun position” beneath the pyramid, there is a “Mars position” about 40 m above the King's chamber. The massive volume of the Pyramid consists of more than 200 courses of stone blocks. Fortunately, Sir W. M. F. Petrie accurately measured the level of each course [6, map. VIII; 13, Tab. 15] so that it is possible to locate the course of the “Mars position.” A true-to-scale drawing of the pyramid's cross section with a grid for higher graphical precision and better visibility of the distances is provided in Fig. 19.

“Mars position”

According to the data of Petrie, the 114. course covers the height from 86.385 m to 86.957 m, measured from the pyramid base. As the vertical coordinate of the “Mars position” is 86.83 meters, the position is located within the 114. course. The horizontal distance from the original southern pyramid surface is about 19.5 meters. But how is the positioning in the east−west direction? The position of the central pyramid axis to the middle plane of the corridors (Fig. 3) is 7.20 m westward [9, part IV, map 3, Fig. 2].[3] Adding half of the corridors width (0.53 m) [9, part IV, map 6, Fig. 4], this adds up to 7.73 m to the east walls. The “Mars position” is located 3.25 m westward from the common plane of the east walls. It follows that we find the “Mars position” about 4.5 m eastward from the vertical middle plane of the pyramid (see top view in Fig. 20). The coordinates of all of the transformed planetary positions inside the pyramid can be computed with the book option 380.

Southern air shaft

An interesting fact is that the “Mars position” at the date of the Mercury transit is almost exactly placed at the original opening of the southern air shaft of the King's chamber. The orbit of Mars in the year 3088 can be calculated very precisely with the VSOP87 theory. The astronomical precision is better than 1 arc second for the heliocentric coordinates. If we transfer this precision to the “planetary positions” in the Great Pyramid, we obtain an accuracy of better than 1 mm for the transformed Mars positions. The chamber positions in the Great Pyramid have an uncertainty of approximately 10 cm, which is still very good. The largest error comes from combining chamber and planetary positions and is given by the relative error in percent (see e.g. the program output in section 3.4.8). The computed error of 0.57 % for the comparison with the chambers means a spacial uncertainty of 0.44 m for the “Mars position.” But what about the z-coordinate, fixing the position in east−west direction? The distance of the middle of the southern air shaft from the East wall of the King's chamber is 2.46 m [9, part IV, map 7, Fig. 10]. At the “chambers date” the corresponding distance of the “Mars position” is 3.25 m (book option 380). If we consider the southern air shaft being oriented exactly in north−south direction, we get an east−west distance of the “Mars position” of about 0.8 m eastward from the opening of the air-shaft, which is rather close.

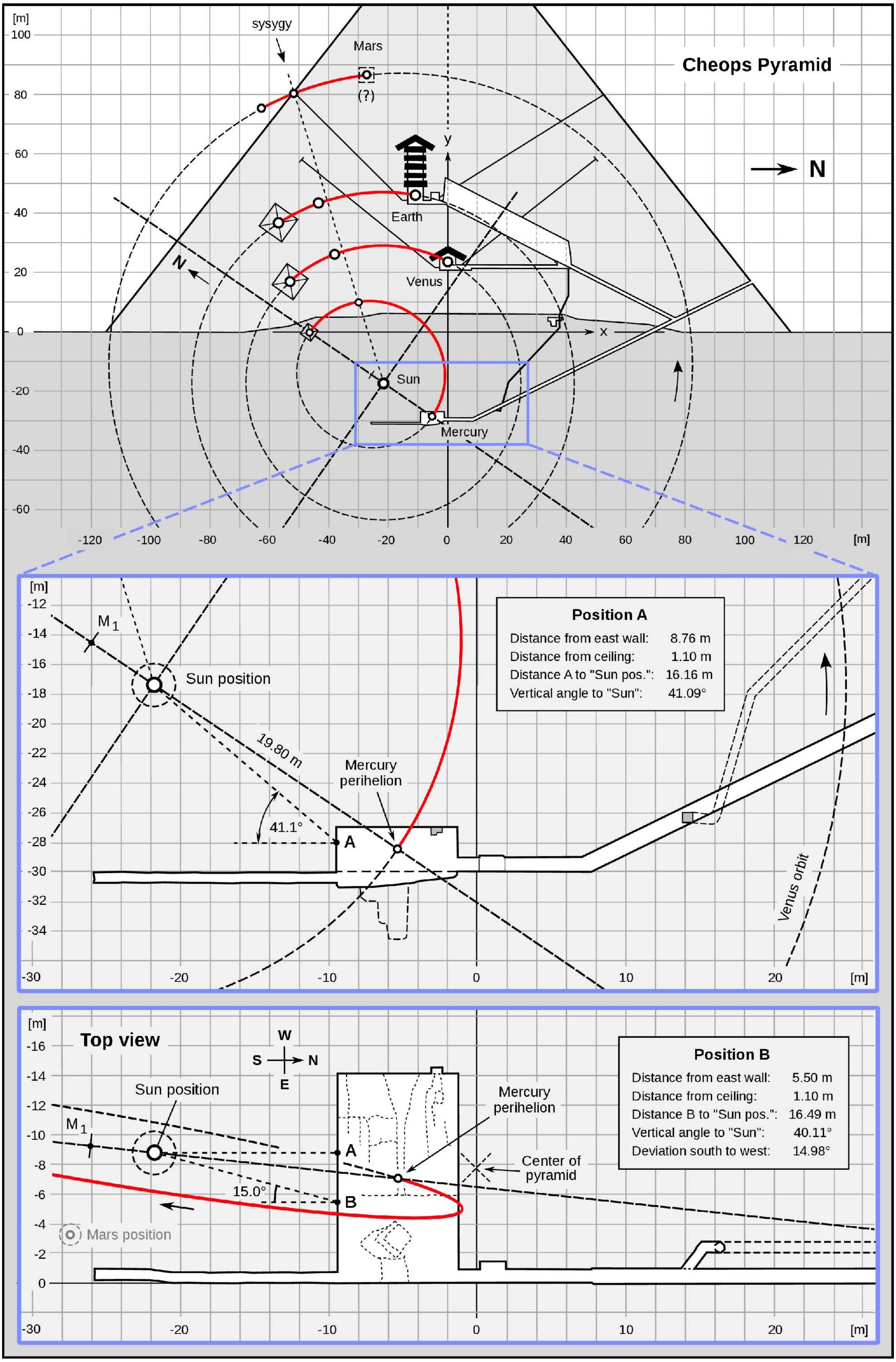
If a horizontal borehole should be drilled from the south side of the pyramid to examine the location of the “Mars position,” the east−west positioning should be determined with both options “middle axis of the south side” and “position of the air shaft” because the exact north−south alignment of the air shaft between King's chamber and pyramid's surface is not sure. The middle axis of the pyramid's south side can be fixed by triangulation from the two south corners of the pyramid. As said before, the “mars position” is located approximately 4.5 m eastward from the central axis of the pyramid. We have an interesting analogy: At the end of the transit, Mercury leaves the Sun's disk, and shortly before, “Mars” leaves the Great Pyramid.

“Sun position”

The “Sun position” is located southward and above the subterranean chamber. The coordinates are given with an accuracy of about 0.20 m. In order to examine how this position can be reached, a detailed view on the area around the subterranean chamber is given in Fig. 20.

---

3 In the given reference the central axis of the pyramid is drawn eastward of the corridors, which is not correct. Actually, the central axis is located westward of the corridors, which can be seen for example in the top view of the Great Pyramid [5, Fig. 163], published originally by Piazzi Smyth [26].



Cheops Pyramid
N
sysygy
Mars
(?)
Earth
Venus
Sun
Mercury
x
y
[m]
Position A
Distance from east wall: 8.76 m
Distance from ceiling: 1.10 m
Distance A to "Sun pos.": 16.16 m
Vertical angle to "Sun": 41.09°
Sun position
19.80 m
Mercury perihelion
41.1°
A
Venus orbit
Top view
W
S
N
E
Sun position
Mercury perihelion
Center of pyramid
15.0°
B
Mars position
Position B
Distance from east wall: 5.50 m
Distance from ceiling: 1.10 m
Distance B to "Sun pos.": 16.49 m
Vertical angle to "Sun": 40.11°
Deviation south to west: 14.98°

Figure 20 (left page): *Cross section of the Great Pyramid with details of the "Sun position" south of the subterranean chamber in 3088 AD. The dimensions of chamber and corridors were taken from drawings of Maragioglio and Rinaldi [9, part IV, maps 3 and 4]. The point A was chosen arbitrarily as an access point if a boreholing is planned in order to examine the transformed Sun position. The position uncertainties of "Sun-" and "Mercury position" are around 20 cm. The drilling should be oriented exactly to the south with an angle of 41.1° above the horizontal plane. (This is only an example how to proceed if an inspection shall be done.) If the residual rock structures in the subterranean chamber are an obstacle, the drilling can also be started closer to the east wall, for example at point B. In this case, the drilling direction is not exactly to the south but with an angular deviation of 15.0° to the west and a vertical angle of 40.1°. Notice that the data of the points A and B in the rectangular frames are not calculated with the "Mercury position" (section 3.4.8) but with the coordinates of the chamber (Table 4). This makes a slight difference. The orbits are orthogonal projections.*

5. Summary and epilogue

Equations (1) – (3) suggest that the three pyramids of Giza represent the three inner planets of our solar system: Mercury, Venus, and Earth (Fig. 21). In all three equations, the Earth is related to the Cheops Pyramid. Venus belongs to the Chefren Pyramid, Mercury to the Mykerinos Pyramid, and the Sun to the light (speed of light). The equations are repeated here.

(numerators)		**(denominators)**
Cheops Pyramid and Earth	$\frac{S_{Cheops}}{c \cdot 1\,\mathrm{s}} = \frac{V_{Earth}}{V_{Sun}}$	Light-second and the Sun
Cheops Pyramid and Earth	$\frac{V_{Cheops}}{V_{Chefren}} = \frac{V_{Earth}}{V_{Venus}}$	Chefren Pyramid and Venus
Cheops Pyramid and Earth	$\frac{S_{Cheops}}{S_{Mykerinos}} = \frac{Q_{Earth}}{Q_{Mercury}}$	Mykerinos Pyramid and Mercury

S and *V* are the base length and volume of the pyramid, *Q* is the aphelion distance from the Sun, and *c* is the speed of light. The first equation, containing a "second," is analyzed in detail in [5], and again in [13] by using the most recent data. Furthermore, the positions of the pyramids and the arrangement of the chambers in the Great Pyramid seem to correlate with the positions of the given three planets. The two points of time, when the positions match exactly, follow each other within a period of 44 days, which is half of the orbital period of Mercury. Between these two events, a conjunction of the four planets Mercury, Venus, Earth, and Mars imply a "linear constellation" of five celestial bodies, namely the four planets and the Sun, which happens with a simultaneous transit of Mercury. This coincidence of the four planets being in conjunction ($dL_{min} < 5°$) and a simultaneous transit only happens more or less every 5,000 years. The basic chronology of the event in terrestrial time is as follows:

Apr. 17, 3088, 06:41:13	:	three inner planets in alignment of chambers, Mercury at perihelion
May 18, 3088, 19:20:59	:	Transit of Mercury in front of solar disk (nearest approach) with *simultaneous* conjunction of Mercury, Venus, Earth, and Mars
May 31, 3088, 06:19:09	:	three inner planets in alignment of pyramids, Mercury at aphelion

Note: While we find the aphelion distance of Mercury in the third equation, Mercury is placed exactly in the aphelion at the "pyramids date" in 3088. Furthermore, the circumstances concerning the obliquity of the ecliptic and the Sun position in midwinter support the planetary correlation. (The constellations can be visualized, e.g., on the web page "Fourmilab," created by John Walker.)

The question is not whether these three equations and the astronomical aspects are correct. They are correct within the given small uncertainties! The main question is whether these equations and correlations are all accidentally valid or not. More precisely, the questions is: How large is the probability that all of these aspects are accidental? In Ref. [5, pp. 87 ff.], a first mathematical estimate of the probability for the *simultaneous* accident was performed and it was found to be less than 1 : 1 million. It follows that these findings most probably are not accidental. By including the additional results of this manual, the probability for the accident becomes even much less!

If the hypothesis of the planetary correlation turns out to be true, some changes in the naming are possible. The Mykerinos Pyramid would be the “First Pyramid,” the Chefren Pyramid would be the “Second Pyramid” and the Cheops Pyramid would be the “Third Pyramid,” according to the sequence of the planets. In the same order, we can alternatively also call them “Mercury Pyramid,” “Venus Pyramid,” and “Earth Pyramid.” The King's chamber, the Queen's chamber, and the subterranean chamber in the Cheops Pyramid could be renamed “Earth chamber,” “Venus chamber,” and “Mercury chamber,” respectively. Furthermore, by continuing the sequence of the planets, the five “relieving chambers” above the King's chamber (Fig. 3) would be named after the five outer planets. Another interesting aspect is that the planetary correlation, calculated with VSOP87, yields a “Sun position” and a “Mars position” within the Cheops Pyramid. Because for many decades scientists have been searching for undetected chambers and corridors in the Cheops Pyramid, these two positions are probably good candidates for a new (secret) chamber.

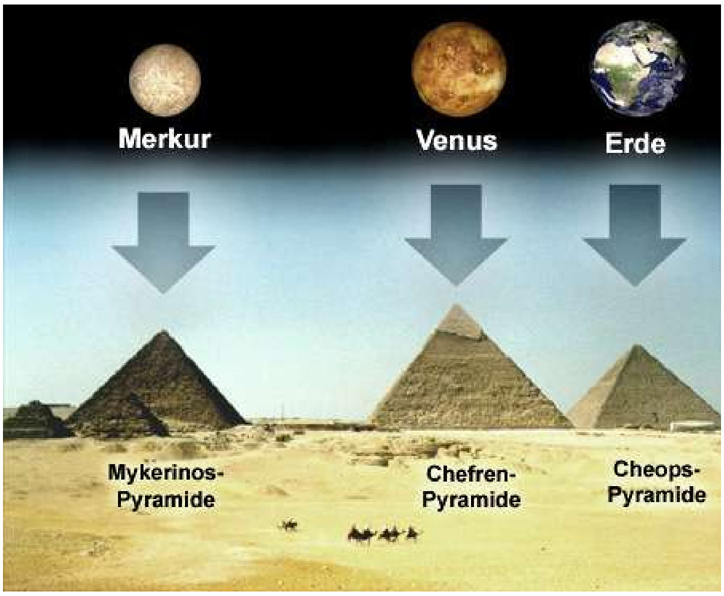


Figure 21: *Planetary correlation of the Giza pyramids. The pyramids are seen from the South (names in German).*

For those who believe that all of these mathematical and astronomical results are accidental coincidences, a technical phenomenon at some of the stone blocks on the Giza plateau was observed, which cannot be explained by neither ancient nor modern technologies. This phenomenon[4] [5, 13] was found at blocks of lime stone and granite and can be proven easily with present experimental methods. Some photos with larger magnification and an explanation of possible experimental tests are provided in technical phenomenon.

The external and own references are listed and readily available. The P4 program, including the executable file, the source code, and associated data files, can be downloaded from the author's website. All used archaeological and astronomical data, as well as the calculations, can be checked by the reader. Most calculations were tested and verified in different ways. Nonetheless, if an error in the calculation or in the approach is found or if the reader has a suggestion for improvement, a note can be sent to: Hans Jelitto, Ewaldsweg 12, D-20537 Hamburg, Germany; contact. If

4 The technical effect is: The original casing stones of the pyramids and, for instance, the granite stones at the valley temple of the Chefren Pyramid have very tiny joints between them with a width of approximately 0.1 mm. This is already known. The new phenomenon at some adjoining blocks is that natural structures, visible on the surface of the blocks, exactly continue across the joint from one to the other block without any misalignment. A “surface effect” because of weathering might – in principle – be possible for lime stone but not for granite. If today a granite block of several tons is cut with a special machine, then a gap of at least a few millimeters exists, and if these blocks are again moved together, then slant lines of natural structures have a displacement or shift of a few millimeters at the joint. This is not the case for a lot of granite blocks in Giza. If this technical effect proves to be true, then it seems that the natural granite was originally “cut” without or nearly without any loss of material. For granite blocks, weighing tons, this is impossible at present, even with high-tech cutting techniques. In the first book [5], this phenomenon is called “fugenübergreifende Strukturen” (German). Translated to English, this could be “joint-exceeding structures” or “joint-transcending structures.”

the reader plans to translate any part of this text or make any modifications to the P4 program the author would appreciate receiving a copy of the results or an Internet link.

Apart from the planetary correlation, additional aspects are propounded in Refs. [5, 13]. In the past, various speculations about mathematical peculiarities concerning the shapes and especially the slope angles of the pyramids were published. The algebraic approaches were classified in [5] and a new interpretation was found by combining the different hypotheses. This is supported by structural conditions of the pyramids: the base area of the Cheops Pyramid not being exactly square, the different rectangular corner sockets for the original casing blocks at the four corners of the Cheops Pyramid, and the original granite casing of the Mykerinos Pyramid.

If only parts of the given results are valid, the consequences for the current research in Egyptology are quite serious. It appears that some high-tech was involved when the pyramids of Giza were built. Since, to our knowledge, the ancient Egyptians did not have any high technology in terms of our present technical level, the next question arises: Did our planet Earth have extraterrestrial visitors in ancient times? Therefore, the principle possibilities of interstellar space travel are discussed in [5, pp. 218 ff.]. Furthermore, the current state of knowledge concerning the so-called exoplanets (extrasolar planets) – planets beyond our solar system – is briefly reviewed, considering this new viewpoint [13].

A detailed discussion of the archaeological measurements and more facets are included in Refs. [5, 13]. Some of the main points in [5] were published as articles in journals (German) [3, 4, 42, 43]. They can be downloaded here or with the links provided in the reference list. Although the main astronomical points of Ref. [13] concerning Giza are presented in this manual, the aspects will be described in more detail in that book and, of course, some new aspects will be presented. At least this is planned.

If the planetary correlation in general is correct, it seems possible that the pyramid builders left some information or something else at the “Mars position” or “Sun position” inside or beneath the Great Pyramid. In this case, it seems important and evident that the information about new chambers, writings, artifacts, or whatsoever – if something will be found – would not only be for some archaeologists or institutions. It would be for the public, which means for everyone who is interested.

Writing this program description required less effort than writing the program code itself. When this manual was written, more or less all astronomical results and details concerning the Giza pyramids were known. When starting programming, we had to start from scratch. From the viewpoint of natural sciences, the scientific context – meaning astronomical and other calculations – is more or less (modern) basic knowledge. Nevertheless, when beginning any such project, new ideas are necessary and there are still a lot of unsolved archaeological problems. I hope that in the future, more private and professional researchers will be interested in such questions within this new young research area.

Assuming that the calculations make sense,
I hope the user has the same enjoyment I had,
when I wrote the program. (Hans Jelitto)

Appendix – P4 Source Code

Fortran 95, free source form

The source code of the P4 program contains notes and comments providing additional technical information; it is intended mainly for programmers. Unfortunately, most (but not all) of the comments are written in German. The version of the program is given by the calendar date at the beginning of the program head. If the source code should be compiled again, it is not necessary to take it from this text because it is available in the file `p4.f95`. Actually, the latter file is the reference! Notice that the compiled P4 source code does not run alone. It needs the supplementary files that are specified in Table 1. The titles and rubrics of this appendix, provided in the "Contents" at the beginning of this manual, are not repeated here. Instead, the entire source code of the executable program is listed continuously. The reader should pay attention to the copyright notes on page 137 concerning the P4 program in general and particular subroutines.

The subroutine VSOP87 [1, 2] has been upgraded (→ VSOP87X), as proposed by Bretagnon and Francou, so that the comprehensive VSOP87 data are read only once from hard disk at program start. The subroutines of FITEX [15, 16] were converted to double precision and all program parts were updated to Fortran 95 standard (gfortran). In principle, the code is converted from the fixed to the free source form, although the maximum length of the code lines is still 72 characters. When a test was performed, not with gfortran but with the Intel Fortran compiler IFORT, available at the Computing Center of the TUHH (Technische Universität Hamburg-Harburg), the source file `p4.f95` had to be renamed to `p4.f90`. For the language standard, the script "Fortran 95 – Nachschlagewerk zur Fortran-Norm ISO/IEC 1539-1:1997" (RRZN, Leibniz Universität Hannover) was used. Unfortunately, this script is sold only to members and students of some universities in Germany, Austria, and Switzerland, and may be used only by them. Anyway, as Fortran is a standardized programming language several other Fortran reference books are useful as well, such as the manual "Using GNU Fortran" [44].

At the beginning of programming, the comments were written only for myself, in order to later understand the logical configuration and meaning of the program. Now, I hope they are also helpful for the reader if needed. For a better readability, the code is highlighted by using the editor "gedit" (print to pdf). I have to admit that the programming style is a bit old-fashioned, like, e.g., the use of the implicit statement. Nevertheless, the program can be started easily, it runs quite fast,[5] and the results seem to be correct.

The program was developed from 1993 until today, but – of course – not continuously. From time to time new ideas arose and were implemented into the program code during countless evenings and weekends. One of the last written subroutines was "pos_angle" (section 4.7.3) to calculate the position angles during a transit, because I was interested in how the transit of the year 3088 looks like. Finally, I hope the program and this manual are also interesting for others.

5 In order to obtain higher processing speed, some "hot spots" in the P4 program were parallelized with the application programming interface (API) "OpenMP." The modified subroutine "VSOP87X" was renamed as "VSOP87Y." For the compilation of the source code, we use the command: `gfortran -fopenmp -O2 p4-4.f95`. The subroutine has been adapted according to four threads, because the used processor has two cores with Hyper Threading. Therefore, the corresponding P4 file names have an additional "4." Now, the calculated *combined* CPU-time is longer than the runtime of P4. So, the execution time, especially of the TYMT-test (64-bit version, see page 16), is determined not only with the subroutine "CPU_time" but also with "date_and_time." The runtime decreases from 46.0 s to about 22.4 s, and with a small terminal window of three lines to 20.0 s. The modification has rather a technical meaning than a practical reason, because the single-thread program is already fast enough. Moreover, with a single-core processor the program would even decelerate. Thus, the original P4 program code is listed in this appendix and the parallelized code files `p4-4.f95`, `p4-4.pdf`, `p4-4-64`, and `p4-4-64.sh` are included in the p4 program package (download).

```
!-----------------------------------------------------------------------!
!                                                                       !
!                           P4 (Fortran 95)                             !
!                                                                       !
!              PLANETENKORRELATION DER PYRAMIDEN VON GIZA               !
!                                                                       !
!                                                                       !
!                                 =                                     !
!                               =   =                                   !
!                             =       =                                 !
!                           =    P 4    =                   =           !
!                         =    Programm   =               =   =         !
!                       =   zur Berechnung  =           =       =       !
!                     =   der Planetenposi-   =       =           =     !
!                   =   tionen und zur Bestim-  =   =               =   !
! =               =   mung des Zeitpunktes, der   =                     !
!   =           =   durch die Pyramidenanordnung    =                   !
!     =       =   bzw. Kammeranordnung vorgegeben     =                 !
!       =   =   ist. Grundlage sind Messungen namhaf-   =               !
!         =   ter Aegyptologen sowie die planetarische    =             !
!           =   Theorie VSOP87 von Bretagnon und Francou    =           !
!             =  (IMCCE, Paris). Das Programm ist eine viel-  =         !
!               =  seitige Weiterentwicklung des Programms P3.  = = =   !
!                 = = = = = = = = = = = = = = = = = = = = = = = = =     !
! = = = = = = = = = =                                                   !
!                                                                       !
!                                                                       !
!                 Hans Jelitto, Hamburg, 6. Juni 2015                   !
!                                                                       !
!                                                                       !
!                           Kurzbeschreibung                            !
!                                                                       !
!         Das Programm P4 berechnet fuer lange Zeitraeume die           !
!         Positionen der Planeten unseres Sonnensystems und er-         !
!         moeglicht einen praezisen Vergleich mit der Anordnung         !
!         der Giza-Pyramiden bzw. der Kammeranordnung innerhalb         !
!         der Cheops-Pyramide. Weiterhin berechnet es die Pha-          !
!         sen der Merkur- und Venustransite vor der Sonne und           !
!         bestimmt Zeitpunkte von "linearen" Planetenkonstella-         !
!         tionen (Syzygium) im Zusammenhang mit den Pyramiden.          !
!         Verschiedene Theorievarianten und eine Vielzahl von           !
!         Optionen ermoeglichen Quervergleiche. Es reproduziert         !
!         die astronomischen Berechnungen in den zwei Buechern:         !
!                                                                       !
!         1. "PYRAMIDEN UND PLANETEN - Ein vermeintlicher Mess-         !
!         fehler und ein neues Gesamtbild der Pyramiden von             !
!         Giza", Wissenschaft und Technik Verlag, Berlin (1999),        !
!         ISBN 3-89685-507-7                                            !
!                                                                       !
!         2. Buch 2 (in Vorbereitung)                                   !
!                                                                       !
!                  ----------------------------------                   !
!                  *         COPYRIGHTS UND         *                   !
!                  *    VERWENDUNG DES PROGRAMMS    *                   !
!                  ----------------------------------                   !
!                                                                       !
!         Bezogen auf das Copyright von H. Jelitto stehen das           !
!         Programm P4 und die uebrigen Programmteile, mit Aus-          !
!         nahme der Datei "p4-manual-06-2015.pdf" und ihren vor-        !
```



```
!         hergehenden Versionen, fuer wissenschaftliche, private,       !
!         Ausbildungs- und paedagogische Zwecke zur freien Ver-         !
!         fuegung, solange der Name des Urhebers ordnungsgemaess        !
!         genannt wird, und duerfen nicht fuer kommerzielle             !
!         Zwecke irgendeiner Art verwendet werden. Kommerzielle         !
!         Nutzung bedarf der schriftlichen Genehmigung. Fuer die        !
!         anderen Programmteile (A. bis C.), die im Folgenden           !
!         aufgezaehlt sind, ist zu pruefen, ob eine Genehmigung         !
!         der Urheber bzw. Copyright-Inhaber erforderlich ist.          !
!                                                                       !
!         (Informationen zur Nutzung und zum Copyright der Datei        !
!         "p4-manual-06-2015.pdf" stehen zu Anfang jener Datei.)        !
!                                                                       !
!         Das Programm P4 wird in der Hoffnung zur Verfuegung           !
!         gestellt, dass es fuer andere nuetzlich ist, jedoch           !
!         ohne irgendeine Art von Garantie oder Gewaehrleistung.        !
!                                                                       !
!         Die folgenden Angaben (A. bis D.) beziehen sich ent-          !
!         sprechend auf das Programm P4, die vorherige Version          !
!         P3 und alle zugehoerigen, unten aufgefuehrten Dateien.        !
!                                                                       !
!      A. Unterprogramm VSOP87X (basierend auf der Theorie              !
!         "Variations Seculaires des Orbites Planetaires") und          !
!         zugehoerige Datenfiles: P. Bretagnon und G. Francou,          !
!         Institut de mecanique celeste et de calcul des                !
!         ephemerides (IMCCE), 77 Avenue Denfert-Rochereau,             !
!         F-75014 Paris, France.                                        !
!                                                                       !
!      B. Programmpaket FITEX (bestehend aus 4 Unterprogrammen          !
!         im hinteren Programmteil): KIT, Karlsruhe Institute of        !
!         Technology (zuvor: FZK, Forschungszentrum Karlsruhe in        !
!         der Helmholtz-Gemeinschaft), Institut fuer Kernphysik,        !
!         Postfach 3640, D-76021 Karlsruhe. FITEX wurde von             !
!         G.W. Schweimer um 1972 entwickelt und erstmals ver-           !
!         oeffentlicht in: H.J. Gils: "The Karlsruhe Code MODINA        !
!         for Model Independent Analysis of Elastic Scattering          !
!         of Spinless Particles." KfK 3063, Nov. 1980, Kernfor-         !
!         schungszentrum Karlsruhe (KfK), Zyklotron Laboratorium,       !
!         and KfK 3063, 1. Supplement, Dec. 1983.                       !
!                                                                       !
!      C. Umrechnung von "terrestrial time" (TT) in "universal          !
!         time" (UT) mittels  delta-T = TT - UT: Fred Espenak,          !
!         und Jean Meeus, NASA Eclipse Web Site, Polynomial             !
!         expressions for DELTA-T.                                      !
!                                                                       !
!      D. Das Hauptprogramm P4 und die uebrigen Programmteile,          !
!         einschliesslich der Modifikation des Unterprogramms           !
!         VSOP87 (--> "VSOP87X"): (c) 2014, 2015 Hans Jelitto,          !
!         Ewaldsweg 12, D-20537 Hamburg, Germany.                       !
!                                                                       !
!                    --------- Danksagung --------                      !
!                                                                       !
!         Das Unterprogramm jdedate zur Umrechnung von JDE in           !
!         ein Kalenderdatum basiert auf einem Algorithmus aus dem       !
!         Buch von Jean Meeus: "Astronomical Algorithms", 1991,         !
!         Willmann-Bell, Inc., P.O.Box 35025, Richmond, Virginia        !
!         23235, USA, S.63. Dafuer und fuer die Auflistung der          !
!         gekuerzten Reihen der VSOP87D-Parameter gilt mein herz-       !
!         licher Dank! Ebenfalls war das Buch "Transits" von            !
```

```
!        J. Meeus (derselbe Verlag) als Basis und zum Testen           !
!        der Transitberechnungen aeusserst hilfreich.                   !
!                                                                       !
!                   -----------------------------                       !
!                                                                       !
!        Zum Programm P4 gehoeren nachfolgende 32 Dateien:              !
!                                                                       !
!        Datei                      Kurzbeschreibung                    !
!    -----------------------------------------------------------        !
!        p4.f95  . . . . FORTRAN-95-Quellcode (dieser Text)             !
!        p4-32 . . . . . Ausfuehrbare Datei fuer 32-bit-System          !
!        p4-64 . . . . .      "          "    "   64-bit-System         !
!        p4-32.sh  . . . loescht Bildschirm und startet p4-32           !
!        p4-64.sh  . . .    "        "        "      "    p4-64         !
!        p4-manual-06-2015.pdf:  Bedienungsanleitung zu P4 und          !
!                        Uebersicht der Planetenkorrelation             !
!        README  . . . . Kurzinformation zur Theorie VSOP87             !
!        vsop87.doc  . . Ausfuehrlichere Information zur Theo-          !
!                        rie "Planetary Solutions VSOP87"               !
!        out.txt . . . . Ergebnis-Datei. Wenn diese nicht be-           !
!                        reits existiert, wird sie bei entspre-         !
!                        chender Option vom Programm erstellt.          !
!                                                                       !
!        inedit.t  . . . Datei zum Editieren der Eingabeparame-         !
!                        ter --> Parametersatz fuer "inparm.t"          !
!        inparm.t  . . . Input gemaess Schnellstart-Optionen            !
!        inpdata.t . . . Parameter f. FITEX, Kammer-Koordinaten         !
!                        in der Cheops-P. und Pyramiden-Koord.          !
!        inserie.t . . . Transitserien fuer Merkur und Venus            !
!        invsop1.t . . . VSOP87D, gekuerzt, Meeus: Astr. Alg.           !
!        invsop3.t . . . Polynomdarstellung der Bahnelemente,           !
!                        berechn. aus VSOP82, Meeus: Astr. Alg.         !
!                                                                       !
!                        VSOP87A, kart. Koord. (Ekl. J2000.0)           !
!        VSOP87A.mer . . Merkur          (Diese und die folgen-         !
!        VSOP87A.ven . . Venus           den Dateien enthalten          !
!        VSOP87A.ear . . Erde            die Parameter zur              !
!        VSOP87A.mar . . Mars            VSOP87-Theorie voll-           !
!        VSOP87A.jup . . Jupiter         staendig.)                     !
!        VSOP87A.sat . . Saturn                                         !
!        VSOP87A.ura . . Uranus                                         !
!        VSOP87A.nep . . Neptun                                         !
!        VSOP87A.emb . . Erde-Mond-Schwerpunktsystem                    !
!                                                                       !
!                        VSOP87C, kart. Koord. (Ekl. d. Epoche)         !
!        VSOP87C.mer . . Merkur                                         !
!        VSOP87C.ven . . Venus                                          !
!        VSOP87C.ear . . Erde                                           !
!        VSOP87C.mar . . Mars                                           !
!        VSOP87C.jup . . Jupiter                                        !
!        VSOP87C.sat . . Saturn                                         !
!        VSOP87C.ura . . Uranus                                         !
!        VSOP87C.nep . . Neptun                                         !
!    -----------------------------------------------------------        !
!                                                                       !
!        Die VSOP87-Dateien wurden 2007 erneut aus dem Internet         !
!        heruntergeladen. Sie sind vom April 2005. Gross- und           !
!        Kleinschreibung sind zu beachten.                              !
!                                                                       !
```



```
!     -----------------------------------------------------------        !
!                    DIE VERSCHIEDENEN OPTIONEN                          !
!     -----------------------------------------------------------        !
!                                                                        !
!              -->  Neue Optionen und Ergaenzungen:                      !
!                                                                        !
!            Gegenueber der Programmversion P3, die fuer das             !
!            Buch "Pyramiden und Planeten" verwendet wurde,              !
!            kommen hier die folgenden Ergaenzungen hinzu:               !
!                                                                        !
!       >    a)  Zu typischen Parameterkombinationen gibt es             !
!       >        eine Reihe von Schnellstart-Optionen (1..15)            !
!       >        und zusaetzlich eine Info-Option (111).                 !
!       >    b)  Verborgene Optionen: Ebenfalls Schnellstart-            !
!       >        optionen - aber nicht im Eingabe-Menue ange-            !
!       >        zeigt - existieren fuer die Resultate in den            !
!       >        Tabellen 39 bis 51 des Buches "Pyramiden und            !
!       >        Planeten" und fuer die Tabellen 17 bis 36 des           !
!       >        Buches 2, das sich in Vorbereitung befindet.            !
!       >        Die Tabelle 39 zum Beispiel besitzt drei Ab-            !
!       >        schnitte, die sich mit den Zahlen 390, 391              !
!       >        und 392 aufrufen lassen, zusammengesetzt aus            !
!       >        39 und 0 bis 2. Das heisst, alle verborgenen            !
!       >        Optionen bestehen aus drei Ziffern!                     !
!       >    c)  Spezialoption -803: Diese erzeugt die Liste             !
!       >        der JDE-Nummern und Transit-Serien in einer             !
!       >        neuen Datei "inser-2.t". Wenn gewuenscht kann           !
!       >        diese Datei dann "inserie.t" ersetzen (im               !
!       >        Allgemeinen nicht erforderlich).                        !
!       >    d)  Optional: Programmstart mit einer Input-Datei           !
!       >        "inedit.t", in der die Parameter manuell edi-           !
!       >        tiert werden koennen (Option 999).                      !
!       >    e)  Koordinaten der drei Kammern der Cheops-Pyra-           !
!       >        mide zum Positionsvergleich mit den Planeten.           !
!       >    f)  Positionsvorgabe durch die Kammermittelpunkte,          !
!       >        bzw. die Mittelpunkte der Ost- und Westwaende           !
!       >        der Kammern.                                            !
!       >    g)  Sechs verschiedene moegliche Zuordnungen der            !
!       >        Planeten Erde, Venus und Merkur zu den drei             !
!       >        Kammern in der Cheops-Pyramide.                         !
!       >    h)  Perihelzeiten beim Merkur, Zeitpunkte nahe              !
!       >        der Periheldurchgaenge und freier Zeitpunkt.            !
!       >    i)  Automatische Erkennung und Markierung der               !
!       >        Planetenkonstellationen 1 bis 14 bei Verwen-            !
!       >        dung beliebiger Optionen.                               !
!       >    j)  Uebertragung der Positionen von Merkur bis              !
!       >        Neptun ins Pyramidengelaende auf Basis der              !
!       >        Pyramiden- bzw. Kammeranordnung (bei 3D-Be-             !
!       >        rechnung mit FITEX, Einzelberechnung, Konst.            !
!       >        1 bis 14). Geographische Koord. (GPS) nur bei           !
!       >        Konst. 12, alle Etappen, fuer Merkur bis Mars.          !
!       >    k)  Kombination VSOP87-Kurzversion und -Voll-               !
!       >        version: Konstellationen, die mit der Kurz-             !
!       >        version gefunden wurden, werden automatisch             !
!       >        mit der Vollversion nachberechnet. Darueber             !
!       >        hinaus: "Zeitintervall um Aphel bzw. um Peri-           !
!       >        hel" auch fuer die Vollversion VSOP87 (sinn-            !
!       >        voll wegen schnellerer Mikroprozessoren und             !
!       >        der Programmoptimierung).                               !
```

```
!        >    l)  Ausser den beiden Optionen "Blick aus Richtung      !
!        >        ekl. Nordpol" und "ekl. Suedpol" sind jetzt         !
!        >        beide Optionen kombiniert moeglich.                 !
!        >    m)  Zeitraeume werden nicht mehr mit der k-Nummer       !
!        >        des Aphel- bzw. Periheldurchgangs des Merkurs       !
!        >        angegeben, sondern mit der eher gebraeuchli-        !
!        >        chen Jahreszahl.                                    !
!        >    n)  Die Berechnungen mit VSOP87 wurde auf den Zeit-     !
!        >        raum 13000 v.Chr. bis 17000 n.Chr. begrenzt.        !
!        >        Ausnahme: "Orbital Elements" und Loesung der        !
!        >        Keplerschen Gl., 30000 v.Chr. bis 30000 n.Chr.      !
!        >    o)  Syzygium: Merkur bis Erde bzw. Merkur bis Mars      !
!        >        in Konjunktion, d.h. 4 bzw. 5 Himmelskoerper        !
!        >        des Sonnensystems in einer Reihe: Sonne, Mer-       !
!        >        kur, Venus, Erde und optional auch Mars.            !
!        >    p)  Zusaetzlich werden Merkur- und Venustransite        !
!        >        vor der Sonnenscheibe registriert (VSOP87C).        !
!        >    q)  Zum Testen der Transit-Berechnung kann man          !
!        >        sich lueckenlos alle Transite von Merkur und        !
!        >        Venus anzeigen lassen, was einen Vergleich          !
!        >        mit Tabellen aus der Literatur bzw. aus dem         !
!        >        Internet ermoeglicht. In diesem Fall werden         !
!        >        Datum und Uhrzeit der Konjunktion, aufsteigen-      !
!        >        der bzw. absteigender Knoten und die Nummer         !
!        >        der jeweiligen Transitserie angegeben.              !
!        >    r)  Als Zeitpunkt fuer den Planetentransit gibt         !
!        >        es erstens das Kriterium "gleiche ekliptikale       !
!        >        Laengen", zweitens "minimale Separation zwi-        !
!        >        schen Sonne und Planet" (ohne Beruecksichti-        !
!        >        gung der Lichtlaufzeit) und drittens "Beginn,       !
!        >        Mitte und Ende des Transits", d.h. die genau-       !
!        >        en Kontaktzeitpunkte bzw. Phasen.                   !
!        >    s)  Bei der Phasenbestimmung gibt es die Option,        !
!        >        zusaetzlich die Positionswinkel des Planeten        !
!        >        waehrend der Phasen in Bezug auf die scheinba-      !
!        >        re Bewegungsrichtung der Sonne zu berechnen.        !
!        >        Hierbei ist eine Zeilenlaenge auf dem Monitor       !
!        >        von mindestens 148 Zeichen erforderlich.            !
!        >    t)  Fuer die Transitphasen gibt es die zwei Zeit-       !
!        >        systeme "terrestrial (dynamical) time" (TT)         !
!        >        und "universal time" (UT). Die Umrechnung mit       !
!        >        delta-T = TT - UT wird ueber analytische Glei-      !
!        >        chungen erreicht (F. Espenak und J. Meeus,          !
!        >        siehe NASA Eclipse Web Site).                       !
!        >    u)  Fuer die Angabe der Transitphasen von Merkur        !
!        >        und Venus wurde eine Datumsberechnung von           !
!        >        J. Meeus integriert. Hierbei gibt es die auto-      !
!        >        matische Kalenderwahl (julianischer bzw. gre-       !
!        >        gorianischer Kalender) oder es wird der grego-      !
!        >        rianische Kalender fuer alle Zeiten verwendet.      !
!        >        Die Datumsberechnung wurde derart modifiziert,      !
!        >        dass sie jetzt auch fuer negative JDE gilt.         !
!        >    v)  Die Berechnung der dezimalen Jahreszahl wurde       !
!        >        insofern verbessert, dass sie jetzt durch 2         !
!        >        lineare Funktionen dargestellt wird, die je-        !
!        >        weils fuer den Zeitraum des julianischen und        !
!        >        des gregorianischen Kalenders stehen (abhaen-       !
!        >        gig von der Kalenderwahl).                          !
!        >    w)  Die Option fuer die Programm-Ausgabe "Drucken"      !
```



```
!        >        im Programm "P3" wurde durch "in Datei" er-         !
!        >        setzt. Hierbei werden die Ergebnisse gleich-        !
!        >        zeitig auf den Bildschirm und in die Datei          !
!        >        "out.txt" geschrieben. Um die Resultate dauer-      !
!        >        haft zu speichern, muss die Datei "out.txt"         !
!        >        nach dem Programmlauf umbenannt werden. Sonst       !
!        >        kann sie beim naechsten Programmlauf ungewollt      !
!        >        ueberschrieben werden.                              !
!        >    x)  Ebenfalls wurde zur Anzeige der Ergebnisse          !
!        >        ein neues Format ergaenzt (special), das fuer       !
!        >        eine Konstellation (z.B. 12) einige spezielle       !
!        >        Parameter ausgibt. Damit lassen sich die we-        !
!        >        sentlichen Tabellen aus dem Buch 2, z.B. mit        !
!        >        den verborgenen Optionen (siehe oben Punkt b),      !
!        >        relativ einfach reproduzieren.                      !
!        >    y)  Optimierung der Rechengeschwindigkeit, unter        !
!        >        anderem durch Modifikation des Daten-Aufrufs        !
!        >        im VSOP87-Unterprogramm (neuer Name: VSOP87X).      !
!        >    z)  Verbesserung der Programm-Ausgabe, z.B. durch       !
!        >        ausfuehrlichere Kopfzeilen, jetzt in Englisch.      !
!        >        Am Ende des Programmlaufs wird die benoetigte       !
!        >        Rechenzeit angegeben (CPU-time).                    !
!     -----------------------------------------------------------     !
!                                                                     !
!                          Optionen insgesamt:                        !
!                                                                     !
!     ---------- Schnellstart-Optionen: ------------------------      !
!            1-15     -->  Die wesentlichen astr. Berechnungen        !
!            111      -->  Information zu Autoren u. Copyrights       !
!            390-512 -->  Tabellen 39-51 aus "Pyram. u. Plan."        !
!            170-381 -->  Tabellen 17-33 und 35-38 aus Buch 2         !
!                          (Das Buch ist in Vorbereitung.)            !
!            999      -->  Input aus "inedit.t" (editierbar)          !
!           -803      -->  Erzeugung der Datei "inser-2.t"            !
!            (0)      -->  Parameter einzeln eingeben                 !
!                                                                     !
!     ---------- Planetenpositionen: ---------------------------      !
!            1.   Anordnung der 3 Pyramiden                           !
!            2.   Anordnung der 3 Kammern der Cheops-Pyramide         !
!            3.   Konjunktionen (Transit, Syzygium)                   !
!                                                                     !
!     ---------- VSOP87-Version: -------------------------------      !
!            1.   Kombination von Kurz- u. Vollversion VSOP87         !
!            2.   VSOP87 Kurzversion (Buch von J. Meeus)              !
!            3.   Keplersche Gleichung mit VSOP82 (Meeus)             !
!            4.   VSOP87 Vollversion (IMCCE, Internet)                !
!                                                                     !
!     ---------- Koordinatensystem in VSOP87: ------------------      !
!            1.   Ekliptik der Epoche (VSOP87C)                       !
!            2.   J2000.0 (VSOP87A, Vollv. und Kepl. Gl.)             !
!                                                                     !
!     ---------- Umfang der Progranmm-Ausgabe: -----------------      !
!            1.   normal (eine Zeile pro Konstellation)               !
!            2.   detailliert (mehrere Zeilen pro Konstell.)          !
!                                                                     !
!     ---------- Zuordnung: Planeten <-> Kammern: --------------      !
!            1.-6.  Sechs moegl. Zuordnungen von Erde, Venus          !
!                 und Merkur zu Koenigs-, Koeniginnen- und            !
!                 Felsenkammer: 1. E-V-M (Standard), 2. E-M-V,        !
```

```
!              3. V-E-M,  4. V-M-E,  5. M-E-V,  6. M-V-E.        !
!                                                                !
!     ---------- Zeitpunkte: -------------------------------------
!           1.  Apheldurchgang des Merkurs                       !
!           2.  Perihelddurchgang des Merkurs                    !
!           3.  Aequidistante Abfolge von Zeitpunkten in         !
!               Zeitintervallen, die jeweils den Aphel-          !
!               durchgang des Merkurs enthalten                  !
!           4.  Aequidistante Abfolge von Zeitpunkten ana-       !
!               log um den Perihelddurchgang des Merkurs         !
!           5.  Zeitpunkt voellig frei und Minimierung der       !
!               Abweichung zwischen Pyramiden und Planeten-      !
!               anordnung durch Variation des Zeitpunkts         !
!                                                                !
!     ---------- "Sonnenposition": ------------------------------
!           1.  genau suedlich Mykerinos-Pyramide (1D)           !
!           2.  genau suedlich Chefren-Pyramide (1D)             !
!           3.  unbestimmt (2D und 3D)                           !
!                                                                !
!     ---------- Berechnung ("Sonnenposition" unbestimmt): ------
!           1.  2-dimensional, Projektion auf Hauptebene         !
!           2.  3-dimensional, durch lineares Gleichungs-        !
!               system und Uebertragung der Loesung              !
!           3.  3-dimensional, Koordinatentransformation         !
!               mit Fit-Programm FITEX                           !
!                                                                !
!     ---------- Referenzsystem bei 2D-Berechnung: --------------
!           1.  Ekliptikales System                              !
!           2.  Merkurbahn-System, Transformtion A, B oder       !
!               C (Gerade "Sonne - Merkur-Aphel" = x-Achse,      !
!               Merkurbahn def. xy-Ebene, Ekl. d. Epoche)        !
!           3.  Venusbahn-System, Transformation A, (Pro-        !
!               jektion "Aphel - Merkur" genau auf x-Achse,      !
!               Venusbahn def. xy-Ebene, Ekl. der Epoche)        !
!                                                                !
!     ---------- "Polaritaet" bei Projektion (2D): --------------
!           1.  Blick vom ekliptikalen Nordpol                   !
!           2.  Blick vom ekliptikalen Suedpol                   !
!           3.  Beide Optionen 1. oder 2.                        !
!                                                                !
!     ---------- Vorgegebene Hoehenlagen (3D): ------------------
!           1.  Grundflaechen der Pyramiden                      !
!           2.  Schwerpunkte   "      "                          !
!           3.  Spitzen        "      "                          !
!                                                                !
!     ---------- Kammerpos. in Cheops-P. (3D, z-Koord.): --------
!           1.  Ostwaende der Kammern                            !
!           2.  Mitte      "     "                               !
!           3.  Westwaende "     "                               !
!                                                                !
!     ---------- Zeitpunkt-Eingabe: -----------------------------
!           1.  Angabe der Konstellation (Nr. 1 bis 14)          !
!           2.  Jahr bzw. Jahresintervall (von ... bis ...)      !
!           3.  Aphel- bzw. Perihelddurchgang (k-Nummer)         !
!           4.  Julian Ephemeris Day (JDE)                       !
!                                                                !
!     ---------- Planeten in Konjunktion: -----------------------
!           1.  Alle Merkur-Transite in einem Zeitintervall      !
!           2.  Alle Venus-Transite   "   "          "           !
```



```
!           3.  Merkur bis Erde in einer Reihe (Syzygium)        !
!           4.  Merkur bis Mars  "   "     "   (    "   )        !
!           5.  Syzygium (Pkt. 3./4.) mit simultanem Transit     !
!                                                                !
!     ---------- Transit-Bestimmung: ----------------------------
!           1.  Transite: gleiche eklipt. Laenge Planet/Erde     !
!           2.  Transite: minimale Separation Planet/Sonne,      !
!               1./2.: ohne Beruecksicht. der Lichtlaufzeit      !
!           3.  Phasen und minimale Separation von der Erde      !
!               aus gesehen, Lichtlaufzeit beruecksichtigt       !
!           4.  Phasen wie in 3. und Positionswinkel             !
!                                                                !
!     ---------- Kalendersystem: --------------------------------
!           1.  Automatische Wahl des Kalenders                  !
!               (Greg. < 4712 BC < Julian. < 1582 AD < Greg.)    !
!           2.  Gregorianischer Kalender fuer alle Zeiten        !
!                                                                !
!     ---------- Zeitsysteme: -----------------------------------
!           1.  "terrestrial dynamical time" (TT) bzw. JDE       !
!           2.  "universal time" (UT), basierend auf delta-T     !
!               (NASA Eclipse Web Site).                         !
!                                                                !
!     ---------- Ausgabegeraet: ---------------------------------
!           1.  Monitor                                          !
!           2.  Monitor + Datei auf Festplatte ("out.txt")       !
!           3.  Spezial-Programmausgabe (auf Mon. + Datei)       !
!           4.  Programm-Abbruch                                 !
!                                                                !
!     -----------------------------------------------------------
!                                                                !
!                          Anmerkungen:                          !
!                                                                !
!     Die letztere Aufzaehlung (Optionen insgesamt) wurde der    !
!     Uebersichtlichkeit halber etwas vereinfacht. Sie entspricht !
!     nicht immer dem Eingabe-Menue, das beim Programmstart mit  !
!     "detailed options (0)" abgefragt wird. Ausserdem sind nicht !
!     alle Kombinationen der Optionen durchfuehrbar. Solche, die !
!     nicht erlaubt sind, werden beim Programmstart gar nicht zur !
!     Auswahl gestellt. Das Programm ist gegen inkorrekte Eingabe !
!     weitestgehend abgesichert. Eine Kontrolle entfaellt nur, wenn !
!     die Input-Parameter in der Datei "inedit.t" manuell editiert !
!     werden und der Programmstart mit der Option 999 erfolgt.   !
!                                                                !
!     Anstelle des FORTRANN-77-Compilers (IBM Professional For-  !
!     tran Compiler, Version 1.0, Ryan McFarland) wird jetzt un- !
!     ter Ubuntu Linux der GNU-Compiler "gfortran" verwendet,    !
!     der den vollen Sprachumfang von Fortran 95 sowie Teile von !
!     Fortran 2003 und Fortran 2008 enthaelt. Das feste Zeilen-  !
!     format wurde (im Prinzip) durch das freie Format ersetzt.  !
!                                                                !
!     Zum Programmpaket FITEX:                                   !
!     Alle Real-Konstanten wurden mit Exponent "D" versehen, eben- !
!     falls Funktionen wie DSQRT usw. eingefuehrt, sowie REAL(8) !
!     und INTEGER(4). EPS wurde von 1.D-5 auf 1.D-8 gesetzt. (An- !
!     passung an Fortran-95-Standard.)                           !
!                                                                !
!     Zum Unterprogramm VSOP87 bzw. VSOP87X:                     !
!     Die VSOP87-Routine wurde dahingehend modifiziert, dass die !
!     umfangreichen Dateien der VSOP87-Theorie nur einmal gelesen !
```

```fortran
!     und im Rechenspeicher in ein Array geschrieben werden. Bei      !
!     den folgenden Berechnungen besteht dann direkter Zugriff auf    !
!     diese Daten, was einen erheblichen Geschwindigkeitsvorteil      !
!     bringt. (Anpassung an Fortran-95-Standard)                      !
!                                                                     !
!     Bei den Konstellationen 13, 14, sowie den "quick start          !
!     options" 371 und 372 wird automatisch auch die jeweilige        !
!     Merkur-Aphelposition berechnet, da sich hierbei der Merkur      !
!     nicht im Aphel seiner Bahn befindet. Dies geschieht jedoch      !
!     nur bei Verwendung bestimmter Optionen, wie z.B. 3D/FITEX.      !
!                                                                     !
!     Dieses Quellprogramm enthaelt auch Code-Abschnitte, die de-     !
!     aktiviert wurden (durch "!c", "!f", "!h" bzw. "!t") und fuer    !
!     spezielle Zwecke gedacht sind. Das Aktivieren einiger Zeilen    !
!     durch Entfernen von z.B. "!h" am jeweiligen Zeilenanfang be-    !
!     wirkt das Einsortieren der Genauigkeiten Fpos in ein Array      !
!     (--> Histogramm: Fpos(0...5%) in Schritten von 0.05%).          !
!                                                                     !
!     Groessere Stellenanzahl in der Ergebnisausgabe (siehe "!f"):    !
!     Fuer einige Programmlaeufe koennen mehr Dezimalstellen ange-    !
!     zeigt werden. Dafuer sind entspechende Format-Statements zu     !
!     ersetzen. Schnellstart-Optionen 4, 9: s. Ende des Hauptpro-     !
!     gramms; 3, 8: s. Ende des Unterprogramms "plako" (durch Akti-   !
!     vieren bzw. Deaktivieren entsprechender Formatzeilen).          !
!                                                                     !
!     Um bei Verwendung der Compiler-Option "-Wuninitialized" bzw.    !
!     "-Wall" Warnmeldungen zu vermeiden, wurden einige Variablen     !
!     zusaetzlich vorab initialisiert und mit "pre-init." markiert.   !
!                                                                     !
!---------------------------------------------------------------------!

!-----Module-----------------------------------------------------------
      module base !    GRUNDLEGENDE VARIABLEN UND KONSTANTEN
      save        ! (Laengen in Metern, Zeiten in julian. Tagen)

      integer(4) :: lmax(15),jp(12,6),il(3)
      real(8)    :: xyr(37),re(78),pyr(40)
      real(8)    :: ax,ay,az,bx,by,bz,cx,cy,cz,ao,ai,at

      real(8), parameter :: pi   = 3.1415926535897932d0, &
        pidg = pi/180.d0,   zjd0 = 2451545.d0, &
        gdpi = 180.d0/pi,   c    = 299792458.d0, &
        tcen = 36525.d0,    AE   = 149597870610.d0, &
        tmil = 365250.d0,   z0   = 0.d0, &

!       ("Allen's Astrophys. Q.", R-Sonne: 695508 km bzw. 958,966",
!       Sonnenradius in "Transits", Meeus: 695990 km bzw. 959,63")
        R0   = 695508000.d0, & ! R-Sonne (Brown/Christensen-Alsgaard)
        R3a  = 6378136.6d0,  R3p = 6356751.9d0, & ! R-Erde, IERS 2003
        pmer = 2451590.257d0, & ! Erste Merkur-Perihelzeit nach J2000
        ymer = 87.96934963d0  ! Merkur-Umlaufzeit: Perihel -> Perihel

      real(8), dimension(2), parameter :: &
!       Radien: Merkur 3,3629", Venus 8,41", Venusradius mit knapp
!       50 km Atmosphaere (ohne Atm. 6051000 m)
        Ra   = (/ 2439000.d0, 6099500.d0 /), & ! Radien (Mer., Ven.)
        tsid = (/  87.9693d0, 224.7008d0 /), & ! T-siderisch ( ", ")
        tsyn = (/ 115.8775d0, 583.9214d0 /), & ! T-synodisch ( ", ")
```



```fortran
!       Theoretischer Massstabsfaktor (Planetenpositionen : Pyramiden-
        zthe = (/ 9.7073d7, 2.3614d9 /) !     bzw. Kammerpositionen)

      real(8), dimension(14), parameter :: &
!       Nummern des Merkur-Apheldurchgangs der Konstellationen 1-14
        akon = (/-38912.d0,  -23134.d0,  -7356.d0,   8422.d0, &
                  24200.d0,  -24130.d0,  -8352.d0,   7426.d0, &
                  23204.d0,   38982.d0,  -4781.d0,   4519.d0, &
                  39313.9134336d0,  -20240.1362451d0 /)
!c                39313.91342804d0, -20240.136249887d0 /)
!                 (alte Werte, Konst. 13, 14, manuell und
!                 iterativ mit P3 bestimmt)
      end module

      module astro
      save
!     Parameter der VSOP87-Kurzversion nach Meeus
      real(8)    :: par1(3,69,6,12)
!     Parameter der VSOP87-Vollversion
      real(8)    :: par2(3,2048,0:5,3,9)
      integer(4) :: it2(0:5,3,9),in2(0:5,3,9),iv2(9)
!     zur Berechnung mittels Keplerscher Gleichung
      real(8)    :: par3(4,6,8,2)
!     zur Bestimmung der Transit-Serie
      real(8), parameter :: t13BC = -3027093.d0, t17AD = 7930183.d0
      real(8), dimension(2), parameter :: cc=(/16802.20d0,88756.13d0/)
      integer(4), dimension(4), parameter :: jj = (/-150,154,-6,19/)
      integer(4), dimension(2), parameter :: ji = (/15,7/)
      real(8)    :: ser(-180:170,2),ase(-180:170),zstart
      integer(4) :: ise(-180:170),isflag,ismax
      end module

      program P4
!-----Hauptprogramm----------------------------------------------------

!-----Deklarationen und Initialisierungen
      use base; use astro
      implicit double precision (a-h,o-z)
      dimension :: res(12),rp(3,4),md(0:9),pan(5),sd(2),zjda(4)
      dimension :: df(6),diff(9),r(6),rku(3),rk(12)
      dimension :: x(7),e(7),iw(100),f(9),y(9),z(9),w(1000)
      dimension :: x0(7),iw0(4),w0(3),zmem(78),inum(0:4)
      dimension :: ida(7),da(7),id5(5,7),da5(5,7)
      dimension :: xx(5),yy(5),test(10),ort(0:9,4),rcm(3),acm(3)
!h    dimension :: ihis(100)  !h
      character(1) :: t1(3),tra(2),tr,dp,ts,sl
      character(2) :: dd,dn,ds,dss,kon
      character(3) :: dk,pla(0:9)
      character(5) :: dmo,dmo5(5)
      character(7) :: emp
      character(8) :: str,str2,str3
      character(10) :: plan(0:9)
      character(20) :: dummy
      character(23) :: text(0:9),tt(2)
      character(49) :: titab
      data diff/0.d0,12.19d0,21.41d0,0.d0,-34.784d0,145.d0,60.4d0, &
               168.d0,21.41d0/,pla/'Sun','Mer','Ven','Ear','Mar', &
              'Jup','Sat','Ura','Nep','E-M'/
      data titab/'body            x[m]      y[m]      z[m]    dr[m]'/
```

```
      data tt/  '  (pyramid positions)  ','  (chamber positions)  '/
      data text/'                       ','  of the "planets"     ', &
              7*'                       ','        barycenter   -->'/
      data plan/'Sun       ','Mercury   ','Venus     ','Earth     ', &
                'Mars      ','Jupiter   ','Saturn    ','Uranus    ', &
                'Neptune   ','Earth-Moon'/
      data str/'    --- '/,str2/'   --    '/,str3/'     --  '/
      data emp/'   --- '/,dn/'  '/,ds/' *'/,dss/' <'/,dp/':'/
      data zjde0/0.d0/,ifitrun/0/,zjdelim/0.d0/,izmin/0/ ! pre-init.
      data i0/0/,ic/0/,ierr/0/ ! pre-init.

!-----Input-Daten und Programmstart
      call inputdata(ipla,ilin,imod,imo4,ikomb,io,lv,ivers, &
       itran,isep,iuniv,ical,ika,iaph,iamax,step,ison,ihi,irb,ijd, &
       zmin,zmax,ak,zjde1,dwi,dwikomb,dwi2,dwi3,nurtr,iek,iop0,iout)
      if (iout==4) then; write(6,*); go to 500; endif
      call cpu_time(zia)
      write(6,'(/''  <P4>   Computation started ...'')')

! . . Die Input-Parameter werden in die Datei "inedit.t" geschrieben.
!     Man kann sie dann gegebenenfalls manuell an geeigneter Stelle in
!     "inparm.t" (Liste der Schnellstart-Optionen) einfuegen, wobei
!     allerdings im Unterprogramm "inputdata" die Schnellstart-
!     Optionen angepasst werden muessen.
      if (iop0/=999 .and.iout/=1) then
        call inputfile(ipla,ilin,imod,imo4,ikomb,io,lv,ivers, &
         itran,isep,iuniv,ical,ika,iaph,iamax,step,ison,ihi,irb,ijd, &
         zmin,zmax,ak,zjde1,dwi,dwikomb,dwi2,dwi3,nurtr,iek,iop,2,iout)
      endif

! . . Parameter fuer Spezial-Output (Konst. 12) --> is12 = 1
      is12 = 0
      if (((ipla==1 .and.iaph==1).or.(ipla==2 .and. &
          iaph==2 .and.ika==1)).and.imod<=2 .and. &
          ikomb==0 .and.iuniv==1 .and.io==2 .and. &
          ison==5 .and.ijd==12 .and.iout==3) is12 = 1

! . . Erstellung weiterer Parameter
      if (iout==1) then
        ix = 6
      else
        ix = 1
        open(unit=ix,file='out.txt')
        write(6,'(9x,''Output file: "out.txt"'')')
      endif
   10 write(6,*); kmin = 0; kmax = 0
      if (ipla/=3) then
        if (ijd>=1 .and.ijd<=14) then
          ak = akon(ijd)
          if (ipla==2 .and.iek==1) ak = ak - 1.d0
          call ephim(0,iaph,ipla,ical,ak,iak,zjde1,zjahr,delt)
        endif
        if (ijd==15 .and.imod==2 .and.iaph<=2) &
          call ephim(0,iaph,ipla,ical,ak,iak,zjde1,zjahr,delt)
      endif
      if (ipla==3 .or.(ipla/=3 .and.ijd==15 .and. &
        (imod/=2 .or.(imod==2 .and.(iaph==3 .or.iaph==4))))) then
        call ephim(2,iaph,ipla,ical,ak,kmin,zjdemin,zmin,delt)
        call ephim(2,iaph,ipla,ical,ak,kmax,zjdemax,zmax,delt)
```



```
        if (ipla==3) izmin = idint(zmin)
      endif

! . . Parameter fuer Transit-Pruefung
      if (ilin==1) then
        itransit=1; il(1)=1; il(2)=3; il(3)=2
      elseif (ilin==2) then
        itransit=2; il(1)=2; il(2)=3; il(3)=1
      else
        itransit=0; il(1)=1; il(2)=4; il(3)=1
      endif

!-----Einlesen der Startwerte und Parameter fuer FITEX
!     sowie der Koordinaten der Pyramiden bzw. Kammern
      j0 = 0
      if (ipla==1) j0 = 18
      if (ipla==3) e(1) = 1.d-6
      if (ipla==1 .or.ipla==2) then
        open(unit=10,file='inpdata.t')
        do i=1,8+j0; read(10,*); enddo
        read(10,*) dummy,(x0(i),i=1,7)
        read(10,*) dummy,(e(i),i=1,7)
        read(10,*)
        read(10,*) dummy,(iw0(i),i=1,4)
        read(10,*) dummy,(w0(i),i=1,3)
        read(10,*)
        read(10,*) dummy,iter
        read(10,*); read(10,*)
!       Indizes von rp: 1. Pyr./Kammern, 2. Koordinaten und "Hoehe"
        read(10,*) dummy,(rp(1,i),i=1,4)
        read(10,*) dummy,(rp(2,i),i=1,4)
        read(10,*) dummy,(rp(3,i),i=1,4)
        read(10,*)
        if ((ison==2 .or.ipla==2).and.is12==0) then
          read(10,*) dummy,diff(2),diff(3)
        else
          read(10,*)
        endif
        do i=1,22-j0; read(10,*); enddo
        do i=1,4; read(10,*) dummy,zjda(i); enddo
        close(10)
        if (ipla==2 .and.imod/=3) call chambers(ika,rp)
      endif

!-----Einlesen der Transitserien zum Festlegen der Startnummer(n)
      if (ilin<=2) then
        do i=-180,170
          ase(i) = z0; ise(i) = i0
          if (.not.(iop0==-803 .and.ilin==2)) ser(i,1) = z0
          ser(i,2) = z0
        enddo
        if (iop0/=-803) then
          open(unit=10,file='inserie.t')
          do i=1,5; read(10,*); enddo
          do i=-150,150,5; read(10,*)idummy,(ser(i+j,1),j=0,4); enddo
          do i=1,4; read(10,*); enddo
          do i=-10,15,5; read(10,*)idummy,(ser(i+j,2),j=0,4); enddo
          close(10)
        endif
```

```
        ismax = -10000; zstart = 99.99d0
      endif

!-----Weitere Initialisierungen
      do i=0,4; inum(i) = i0; enddo
      isflag = i0; iflag1 = i0; iflag2 = i0
      ifl = i0; ipos = i0; nfit = 7; mfit = 9
      ipar = i0; if (isep==4) ipar = 2
      indx = 1; iekk = iek; prec = z0
      lu = 10; delt = z0; step = step/24.d0
      diff1 = diff(2); diff2 = diff(3)
      zamax = dfloat(iamax); zjdevor = -1.d10
      do i=0,9; md(i) = 1; enddo
!h    do i=1,100; ihis(i) = i0; enddo  !h

!     Initialisierung zur Berechnung fuer die Datei "inserie.t",
!     (--> "inser-2.t", danach manuelles Kopieren nach "inserie.t")
      if (iop0==-803) then
        if (ilin==1) is = -177  ! fuer Merkur, Jahre -18000 bis 18000
        if (ilin==2) is = -6    ! fuer  Venus, Jahre -30000 bis 30000
      endif

! . . Berechnung des Zeitsprungs fuer die Option "Linearkonstell.";
!     "tsprung" ist ein Zeitintervall in Tagen, das nach dem Ablauf
!     einer Konjunktion von Venus und Erde uebersprungen wird. Dieses
!     darf nicht zu gross sein, um alle Ereignisse zu erfassen.
!     Das erste Ereignis im Intervall der Jahre -13000 bis 17000 geht
!     verloren fuer tsy = 577 Tage (tsprung = 557 Tage, dwin = 5 Grad),
!     d.h. "tsprung" waere zu gross. Darueber hinaus ergab sich je-
!     weils als groesster zulaessiger Wert fuer tsy (Version Kepl.):
!
!        dwin    tsy   tsprung       dwin    tsy   tsprung
!       [Grad]  [Tage]  [Tage]      [Grad]  [Tage]  [Tage]
!      ------------------------     ------------------------
!         5     576     557           20     577     510
!        10     578     543           45     578     430 (not used)
!        15     578     527           90     575     286 (not used)
!      ------------------------     ------------------------
!
!     Die Gleichung fuer tsprung (siehe unten) ist sinnvoll, da alle
!     tsy-Werte etwa gleich gross sind, was auch fuer die Optionen
!     "Kurzv." und "Kombi." gilt. Zur Sicherheit wurde tsy = 570 Tage
!     festgelegt (synodische Umlaufzeit der Venus: 583.9 Tage).
      if (ipla==3 .and.ison==5) step = 1.d0
      dwi0 = dwi;  tsy = 570.d0    ! (fuer Syzygien)
      if (ilin==1) tsy = 115.7d0   ! (Merkur, optim.)
      if (ilin==2) tsy = 582.7d0   ! (Venus, optim.)
      if (ipla==3 .and.ikomb==i0) dwi = dwi + 1.d0; dwin = dwi
      if (ilin<=2) tsprung = tsy
      if (ilin>=3) tsprung = dnint(tsy*(1.d0-dwin/180.d0))
      if (tsprung<1.d0) tsprung = 1.d0

! . . Blickrichtung von der suedlichen ekliptikalen Hemisphaere
      if (iek==2 .and.ipla/=3) then
        diff1 = -diff1; diff2 = -diff2
        do i=1,9; diff(i) = -diff(i); enddo
      endif
      if (ipla==3) go to 20
```



```
!-----Pyramidenabstaende und Winkel
!     Indizes von "pyr":
!         1 bis 5: leer
!         6: leer     7: pdx      8: pdy      9: pdz     10: leer
!        11: pax     12: pbx     13: pcx     14: pay     15: pby
!        16: pcy     17: paz     18: pbz     19: pcz     20: leer
!        21: pa      22: pb      23: pc      24: pb/pa oder pbx/pax
!        25: pc/pa oder pby/pay  26: pc/pb oder pby/pbx  27: alpha
!        28: beta    29: gamma   30: leer    31: alpha1  32: alpha2
!        33: alpha3  34: pax/2   35: pay/2   36: pbx/2   37: pby/2
!        38: (pax+pbx)/2   39: (pay+pby)/2   40: leer
!     Indizes 11-19 und 21-29 bei "pyr" und "xyr" entsprechen sich.
!
! . . Anpassung der Koordinaten fuer Grundflaeche, Schwerpunkt und
!     Spitze der Pyramiden bzw. Ostwand, Mitte und Westwand der
!     Kammern.
      if (ihi==2) then
        cm = 0.25d0
        if (ipla==2) cm = 0.5d0
        do i=1,3; rp(i,4) = rp(i,4) * cm; enddo
      endif
      if (ihi==2 .or.ihi==3) then
        do i=1,3; rp(i,3) = rp(i,3) + rp(i,4); enddo
      endif
! . . Abstaende der Pyramiden bzw. Kammern und weitere Groessen.
      pyr(11) = rp(2,1)-rp(3,1)
      pyr(12) = rp(1,1)-rp(3,1)
      pyr(14) = rp(2,2)-rp(3,2)
      pyr(15) = rp(1,2)-rp(3,2)
      pyr(17) = rp(2,3)-rp(3,3)
      pyr(18) = rp(1,3)-rp(3,3)
      pyr(13) = pyr(12)-pyr(11)
      pyr(16) = pyr(15)-pyr(14)
      pax = pyr(11);  pay = pyr(14);  paz = z0
      pbx = pyr(12);  pby = pyr(15);  pbz = z0
      pcx = pyr(13);  pcy = pyr(16);  pcz = z0
      if (ison==3) then
        pyr(31) = - datan(pyr(14)/pyr(11))
        pyr(32) = - datan(pyr(15)/pyr(12))
        pyr(33) = - datan(pyr(16)/pyr(13))
        pyr(34) = pyr(11)*0.5d0;  pyr(35) = pyr(14)*0.5d0
        pyr(36) = pyr(12)*0.5d0;  pyr(37) = pyr(15)*0.5d0
        pyr(38) = (pyr(11)+pyr(12))*0.5d0
        pyr(39) = (pyr(14)+pyr(15))*0.5d0
      endif
!     Koordinaten des gemeinsamen Zentrums "rcm" der drei Pyramiden
!     bzw. Kammern und mittlerer Abstand zu den Pyramiden bzw. Kammern
!     "dmi" (zur Fehlerberechnung von "Sonnen-","Planeten- und Aphel-
!     positionen" in Giza in den Subroutinen "sonpos", "aphelko" und
!     "plako")
      do i=1,3; rcm(i) = (rp(1,i) + rp(2,i) + rp(3,i))/3.d0; enddo
      do i=1,3
        acm(i) = dsqrt((rp(i,1)-rcm(1))**2 + (rp(i,2)-rcm(2))**2 &
                     + (rp(i,3)-rcm(3))**2)
      enddo
      dmi = (acm(1) + acm(2) + acm(3))/3.d0
!c    do i=1,8
!c      write(6,'(5f12.6)') (pyr(5*(i-1)+j),j=1,5)
!c    enddo
```

```fortran
! . . Zusaetze zur 3-dim. Berechnung
      if (ison>=4) then
        pyr(19) = pyr(18) - pyr(17)
        paz = pyr(17); pbz = pyr(18); pcz = pyr(19)
!c      write(6,'('' x: '',3f12.3)') (pyr(i),i=11,13)
!c      write(6,'('' y: '',3f12.3)') (pyr(i),i=14,16)
!c      write(6,'('' z: '',3f12.3)') (pyr(i),i=17,19)
! . . . Erzeugung eines Vektors pd, der auf pa und pb senkrecht steht.
        pdx = pby * paz - pay * pbz
        pdy = pax * pbz - pbx * paz
        pdz = pbx * pay - pax * pby
        aba = dsqrt(pax*pax + pay*pay + paz*paz)
        abb = dsqrt(pbx*pbx + pby*pby + pbz*pbz)
        abd = dsqrt(pdx*pdx + pdy*pdy + pdz*pdz)
        dfakt = (abb + aba) * 0.5d0/abd
        pyr(7) = pdx * dfakt
        pyr(8) = pdy * dfakt
        pyr(9) = pdz * dfakt
! . . . Modellwerte fuer FITEX
        if (ison==5) then
          z(1) = z0;   z(2) = z0;   z(3) = z0
          z(4) = pax;  z(5) = pay;  z(6) = paz
          z(7) = pbx;  z(8) = pby;  z(9) = pbz
        endif
      endif
! . . Laengen, Laengenverhaeltnisse, Winkel
      if (ison<=2) then
        pyr(24) = pbx/pax
        pyr(25) = pby/pay
        pyr(26) = pby/pbx; if (iek==2) pyr(26) = -pyr(26)
      else
        pyr(21) = dsqrt(pax*pax + pay*pay + paz*paz)
        pyr(22) = dsqrt(pbx*pbx + pby*pby + pbz*pbz)
        pyr(23) = dsqrt(pcx*pcx + pcy*pcy + pcz*pcz)
        pyr(24) = pyr(22)/pyr(21)
        pyr(25) = pyr(23)/pyr(21)
        pyr(26) = pyr(23)/pyr(22)
        pyr(27) = dacos((pax*pbx+pay*pby+paz*pbz)/(pyr(21)*pyr(22)))
        pyr(28) = dacos((pax*pcx+pay*pcy+paz*pcz)/(pyr(21)*pyr(23)))
        pyr(29) = dacos((pbx*pcx+pby*pcy+pbz*pcz)/(pyr(22)*pyr(23)))
      endif

!-----Einlesen aller Parameter der VSOP87D-Kurzversion (Meeus)
   20 if (imod==1) then
        open(unit=10,file='invsop1.t')
        read(10,*)
        do n=1,12
          read(10,*); read(10,*) lmax(n)
          read(10,*) (jp(n,j),j=1,lmax(n))
          do m=1,lmax(n)
            read(10,*)
            do j=1,jp(n,m)
              read(10,*) idummy,(par1(i,j,m,n),i=1,3)
            enddo
          enddo
        enddo
        close(10)
      endif
```



```fortran
!-----Bahnparameter als Polynome 3. Grades aus VSOP82 (Meeus)
      if (io==2 .or.irb/=1 .or.imod==3 .or.ipla==3) then
        open(unit=10,file='invsop3.t')
        do ll=1,2
          do n=1,3; read(10,*); enddo
          do k=1,8
            do n=1,2; read(10,*); enddo
            do j=1,6; read(10,*) (par3(i,j,k,ll),i=1,4); enddo
          enddo
        enddo
        close(10)
      endif

!-----Titelzeilen
      do iu=ix,6,5
        call titel1(iaph,ijd,iu,ison,ipla,ilin,isep,nurtr, &
          iuniv,is12,iop0)
        call titel2(iu,imod,ivers,irb,ipla, &
          ison,ihi,iek,ijd,ika,iaph,ilin,ical,ak,zjde1,zjahr,delt, &
          dwi,dwikomb,dwi0,dwi2,dwi3,iamax,step,ikomb,zmin,zmax)
! . . . Tabellenkopf
        call tabe(iaph,imod,iek,iu,io,ison,ipla,ilin,itran,is12, &
          iop0,iout)
      enddo
      if (iaph==5) go to 200
      if (ipla==3) go to 300

!     Anmerkung: In jedem Programmlauf wird nur eine
!     der drei folgenden Hauptschleifen verwendet.

!======================================================================
!------------------------- 1. Hauptschleife --------------------------
!======================================================================

!-----1. Hauptschleife (Pyramiden- und Kammerpositionen---------------
!     sowie Aphel- und Perihelzeitpunkte des Merkur)
      k = kmin
  100 zk = dfloat(k)
      if (imod==2 .and.ijd==15 .and.iaph<=2) zk = ak
      isw = 1; if (iaph<=2 .and.iout==3) isw = 2
      jmax = i0; ncount = i0

!.....JDE-Zeitpunkt (Merkur im und ausserhalb des Aphels)
  120 zjde = zjde1
      if (ijd==15 .or.iaph==3 .or.iaph==4) then
        ik = k
        if (isw==1 .or.(isw==2 .and.iaph<=2)) then
          if (ijd==15 .and.(imod/=2 .or. &
            (imod==2 .and.(iaph==3 .or.iaph==4)))) ak = zk
          if (ijd==15) then
            call ephim(i0,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
          else
            call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
          endif
        else
          acount = dfloat(ncount)
          if (ijd==15) then
            ak = zk + step * (acount - zamax * 0.5d0)/ymer
            call ephim(i0,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
```

```
          else
            zjde = zjdel + step * (acount - zamax * 0.5d0)
            call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
          endif
        endif
      endif
      if (ijd==i0) call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
      ik = idnint(ak)
      time = (zjde - zjd0)/tcen
      tau  = (zjde - zjd0)/tmil
      if (ison==5) then
        do i=1,4; iw(i) = iw0(i); enddo
        do i=1,3; w(i) = w0(i); enddo
        do i=1,7; x(i) = x0(i); enddo
        do i=4,6; x(i) = x(i) * pidg; enddo
      endif
      inum(1) = inum(1) + 1

!.....Variante 1 (VSOP87D, Kurzversion aus "Meeus")
      if (imod==1) then
        do i=1,9; call vsop1(i,tau,resu); re(i) = resu; enddo
      endif

!.....Variante 2 (VSOP87A/C, Vollversion)
  140 if (imod==2) then
        do i=1,3; ii = 3*(i-1)
          call vsop2(zjde,ivers,i,md,ix,prec,lu,r,ierr,rku)
          do j=1,3; re(ii+j) = rku(j); enddo
        enddo
      endif

!.....Variante 3 (Keplersche Gleichung, Polynome 3. Grades nach VSOP82)
      if (io==2 .or.irb/=1 .or.imod==3) then
        immax = 3; if (io==2) immax = 4
        do i=1,immax; ii = 6*i
          call vsop3(lv,i,ix,ir,time,res)
          if (ir/=i0) go to 500
          re(25+ii) = res(1);  re(28+ii) = res(5)
          re(26+ii) = res(2);  re(29+ii) = res(4)
          re(27+ii) = res(3);  re(30+ii) = res(6)
          if (imod==3 .and.i<=4) re(3*i-2) = res(11)
        enddo
      endif

!.....Koordinaten-Transformation und Bestimmung von F-pos
      if (irb>=2 .or.imod/=3) call kartko(ison)
      if (irb>=2) call transfo(irb,rku)
      if (irb>=2 .or.imod/=3) &
         call relpos(ipla,ison,ijd,iek,iekk,ika)

!.....Korrelation der Positionen pruefen, Output
      ic   = i0
      err3 = z0
      err4 = z0
      dif1 = re(1) - re(4); call reduz(dif1,i0,i0)
      dif2 = re(1) - re(7); call reduz(dif2,i0,i0)
      if (ison<=2) then
        err1 = dif1 - diff1; call reduz(err1,i0,i0)
        err2 = dif2 - diff2; call reduz(err2,i0,i0)
```



```
        if (iek==3) then
          err3 = dif1 + diff1; call reduz(err3,i0,i0)
          err4 = dif2 + diff2; call reduz(err4,i0,i0)
        endif
!.......Hauptbedingung pruefen (ison = 1, 2). . . . . . . . . . . . . .
        if ((dabs(err1)<=dwi.and.dabs(err2)<=dwi).or.ijd/=15 &
          .or.(iek==3 .and.dabs(err3)<=dwi.and.dabs(err4)<=dwi) &
          .or.(ijd==15 .and.imod==2 .and.ikomb==i0)) then
          if (ikomb==1 .and.imod==1) then
            imod = 2; dwi = dwikomb; go to 140
          endif
          if (iek==3) then
            iekk = 1
            if (dabs(err3)<=dwi.and.dabs(err4)<=dwi) iekk = 2
          endif
          inum(2) = inum(2) + 1
          ic = 1
!         Resultat Output
          call konst(ik,kon)
          dd = dn
          if (iek==2 .or.iekk==2) dd = ds
          do iu=ix,6,5
            if (imod/=3) then
              if (iek==3 .and.iekk==1) then
                write(iu,56)kon,ik,zjde,zjahr,re(1), &
                  dif1,dif2,err1,err2,dd
              elseif (iek==3 .and.iekk==2) then
                write(iu,56)kon,ik,zjde,zjahr,re(1), &
                  dif1,dif2,err3,err4,dd
              else
                write(iu,55)kon,ik,zjde,zjahr,re(1), &
                  dif1,dif2,err1,err2,xyr(36)
              endif
            else
              if (iek==3 .and.iekk==2) then
                write(iu,56)kon,ik,zjde,zjahr,re(1), &
                  dif1,dif2,err3,err4,dd
              else
                write(iu,56)kon,ik,zjde,zjahr,re(1), &
                  dif1,dif2,err1,err2,dd
              endif
            endif
          enddo
        endif
      else
        if ((iaph==3 .or.iaph==4).and.isw==1 .and.ijd==15) then
          ifl = i0; if (xyr(36)<=dwi2) ifl = 1
        endif
!.......Hauptbedingung pruefen (ison = 3, 4, 5) . . . . . . . . . . . .
        if (((isw==1 .or.(isw==2 .and.iaph<=2)).and. &
           (xyr(36)<=dwi.or.ijd/=15 .or. &
           (imod==2 .and.ikomb==i0.and.iaph<=2))).or. &
           (isw==2 .and.((ifl==1 .and.xyr(36)<=dwi3.and. &
           ijd==15).or.ijd/=15))) then
          if (ikomb==1 .and.imod==1) then
            imod = 2; dwi = dwikomb
            go to 140
          endif
          inum(2) = inum(2) + 1
```

```
      !         Sonnenposition
                call sonpos(ison,iek,ix,rp(3,1),rp(3,2),rp(3,3),rcm,dmi, &
                            iter,iw,ke,mfit,nfit,f,x,e,w,y,z)
                ic = 1; dd = dn
                if (iek==2) dd = ds
                do isun=1,4; ort(i0,isun) = xyr(30+isun); enddo
      !         Resultat Output
                if (isw==1) then
                  call konst(ik,kon)
                  do iu=ix,6,5
                    if (ison==5) then
                      if (ipla==2) then
                        write(iu,184)kon,ik,zjahr,dif1,dif2,ke,iw(3), &
                          (xyr(30+i),i=1,4),dd,xyr(36)
                      else
                        write(iu,165)kon,ik,zjahr,dif1,dif2,ke,iw(3), &
                          (xyr(30+i),i=1,4),dd,xyr(36)
                      endif
                    elseif (ison==3) then
                      write(iu,67)kon,ik,zjahr,re(1),dif1,dif2, &
                        xyr(31),xyr(32),emp,xyr(34),dd,xyr(36)
                    else
                      if (ipla==2) then
                        write(iu,85)kon,ik,zjahr,re(1),dif1,dif2, &
                          (xyr(30+i),i=1,4),dd,xyr(36)
                      else
                        write(iu,65)kon,ik,zjahr,re(1),dif1,dif2, &
                          (xyr(30+i),i=1,4),dd,xyr(36)
                      endif
                    endif
                  enddo
                else
                  if (((xyr(36)<=dwi2.or.iaph<=2).and.ijd==15).or. &
                      ijd/=15 .or.imod==2) then
                    if (iout==3) then
                      call konst(ik,kon)
                      delh = delt * 24.d0
                      call reduz(x(5),1,i0)
                      if (ipla==1) then
                        xma = xyr(35) * 1.d-7
                        sonne = -datan((xyr(33)-rp(3,3))/xyr(31))*gdpi
                      else
                        xma = xyr(35) * 1.d-9
                        dxr = xyr(31)-rp(3,1); dyr = xyr(32)-rp(3,2)
                        sonne = -datan(dyr/dxr)*gdpi
                        if (dxr*dcos(sonne*pidg)>0.d0) sonne = sonne + 180.d0
                        call reduz(sonne,i0,i0)
                      endif
                      do iu=ix,6,5
                        if (iaph==3 .or.iaph==4) then
                          if (ipla==2) then
                            write(iu,275)zjde,delh,x(5)*gdpi,xma, &
                              sonne,(xyr(30+i),i=1,4),dd,xyr(36)
                          else
                            write(iu,255)zjde,delh,x(5)*gdpi,xma, &
                              sonne,(xyr(30+i),i=1,4),dd,xyr(36)
                          endif
                        elseif (iaph<=2) then
                          if (ipla==2) then
```



```
                            write(iu,276)kon,ik,zjahr,x(5)*gdpi,xma, &
                              sonne,(xyr(30+i),i=1,4),dd,xyr(36)
                          else
                            write(iu,256)kon,ik,zjahr,x(5)*gdpi,xma, &
                              sonne,(xyr(30+i),i=1,4),dd,xyr(36)
                          endif
                        endif
                      enddo
                    else
      !                 Pruefung zur Signifikanz --> dk
                      dk = '   '
                      zf = dabs((xyr(35)-zthe(ipla))/zthe(ipla))
                      if (zf<=2.d-2 .and.xyr(36)>0.5d0) dk = 'M  '
                      if (zf>2.d-2 .and.xyr(36)<=0.5d0) dk = 'F  '
                      if (zf<=2.d-2 .and.xyr(36)<=0.5d0) dk = 'FM '
                      if (zf<=1.d-3 .and.xyr(36)<=0.1d0) dk = '>>>'
                      do iu=ix,6,5
                        if (ison==5) then
                          if (ipla==2) then
                            write(iu,386)dk,ik,zjde,xyr(35),ke,iw(3), &
                              (xyr(30+i),i=1,4),dd,xyr(36)
                          else
                            write(iu,366)dk,ik,zjde,xyr(35),ke,iw(3), &
                              (xyr(30+i),i=1,4),dd,xyr(36)
                          endif
                        elseif (ison==3) then
                          write(iu,367)dk,ik,zjde,xyr(35),ncount-iamax/2, &
                            xyr(31),xyr(32),emp,xyr(34),dd,xyr(36)
                        else
                          if (ipla==2) then
                            write(iu,384)dk,ik,zjde,xyr(35),ncount-iamax/2, &
                              (xyr(30+i),i=1,4),dd,xyr(36)
                          else
                            write(iu,365)dk,ik,zjde,xyr(35),ncount-iamax/2, &
                              (xyr(30+i),i=1,4),dd,xyr(36)
                          endif
                        endif
                      enddo
                    endif
                  endif
                endif
      !h        call histogramm(xyr(36),ihis)  !h
              endif
            endif

      !.....Weiterer Output
            do iu=ix,6,5
              if (ic==1 .and.imod/=3 .and.io==2 .and.is12==0) then
                call linie(iu,2)
                write(iu,57) (re(i),i=1,9)
                do i=1,3; t1(i) = ' '; if (xyr(3+i)<z0) t1(i) = '-'; enddo
                write(iu,54) (xyr(i),i=1,3),t1(1),dabs(xyr(4)), &
                  t1(2),dabs(xyr(5)),t1(3),dabs(xyr(6)),(xyr(i),i=7,9)
                write(iu,'(1x,6f9.6,f22.8,'' %'')') xyr(11),xyr(12), &
                  xyr(14),xyr(15),xyr(17),xyr(18),xyr(36)
                call linie(iu,2)
              endif
              if (is12/=0) call linie(iu,1)
              if (is12==0 .and.ic==1.and.imod==3.and.io==2) call linie(iu,2)
```

```
          if (ic==1 .and.io==2 .and.is12==0) then
            if (imod/=3) then
              if (ivers==3) then
                write(iu,'(''  ascending node (M/V/E/Ma): '',2f12.6, &
                 & ''      ---   '',f12.6)')re(34),re(40),re(52)
              else
                write(iu,'(''  ascending node (M/V/E/Ma): '',4f12.6)') &
                 (re(28+6*i),i=1,4)
              endif
              write(iu,'(''  inclination i  (M/V/E/Ma): '',4f12.6)') &
                (re(29+6*i),i=1,4)
              write(iu,'(''  perihelion pi  (M/V/E/Ma): '',4f12.6)') &
                (re(30+6*i),i=1,4)
              if (imod/=3 .and.irb/=1) &
              write(iu,'(''  ang. par. (omega, i, tau): '',3f12.6)') &
                ao*gdpi,ai*gdpi,at*gdpi
              if (ison==5) then
                write(iu,'(''  transl. X1, X2, X3; del-t: '',3f12.6, &
                  & f9.3,'' days'')') (x(i),i=1,3),delt
                do i=4,6; call reduz(x(i),1,i0); enddo
                write(iu,'(''  Euler angl. X4, X5, X6; M: '',3f12.6, &
                  & f13.0)') (x(i)*gdpi,i=4,6),xyr(35)
!c              write(6,'(''  X7: '', f12.6)') x(7)
              endif
            else
              do i=5,8; ii = 6*i
                call vsop3(lv,i,ix,ir,time,res); if (ir/=i0) go to 500
                re(25+ii) = res(1);  re(28+ii) = res(5)
                re(26+ii) = res(2);  re(29+ii) = res(4)
                re(27+ii) = res(3);  re(30+ii) = res(6)
              enddo
              call elements(iu,ivers,pla)
            endif
            if ((ison==3 .and.ijd>=1 .and.ijd<=10).or.ison==4) write(iu,&
              & '(''  scale factor M           : '',f13.0)')xyr(35)
            call linie(iu,1)
          endif
        enddo

!.....Output: Koordinaten aller Planeten einschliesslich Neptun und
!     des Schwerpunktsystems Erde-Mond, letzteres nur fuer VSOP87A,
!     sowie transformierte "planetarische" Koordinaten in Giza
        if ((imo4==1 .and.iaph<=2 .and.is12==0 .and.io==2) &
            .or.is12/=0) then
          call plako(diff,ipla,ijd,ik,ison,ipos, &
           rcm,x,y,ort,rp,dd,dn,dss,pla,plan,emp,text,tt,titab, &
           is12,dmi,zjda,zjde,ivers,md,ix,prec,lu,r,ierr,rku)
        endif

! . . Ruecksprung fuer Aphel-Umgebung
        if (ikomb==1 .and.imod==2) then
          imod = 1; dwi = dwi0
        endif
        if (iaph==3 .or.iaph==4) then
          ncount = ncount + 1
          if (ncount>jmax) then
            ncount = i0
            if (isw==1) then
              if (ijd==15 .and.ifl==i0) go to 190
```



```
                isw = 2; jmax = iamax; go to 120
              endif
            else
              go to 120
            endif
          endif

! . . Standardruecksprung
  190 k = k + 1
      if (k<=kmax) go to 100

!.....Aphelposition der Merkurbahn fuer Konstellation 13 bzw. 14,
!     sowie "quick start option" 371 und 372
      if (ipla/=3) call aphelko(imod,ivers,iaph,ipla, &
         ison,ijd,io,iop0,ix,rp(3,4),x,y,rcm,dmi)

!-----Ende der 1. Hauptschleife (Pyramiden- und Kammerpositionen)------
      go to 400


!=====================================================================
!------------------------- 2. Hauptschleife --------------------------
!=====================================================================

!-----2. Hauptschleife (freier Zeitpunkt und Minimierung von Fpos------
!     fuer Pyramiden- und Kammeranordnung, Tabelle 51 in "Pyramiden
!     und Planeten" und Tabelle 20 (?) im zweiten Buch)
  200 zjde = zjdemin
      dfe = 0.3d0; eep = e(1)
      irestart = i0; x36 = z0
!     VORSICHT: "zfact" und "zstep" nicht zu gross waehlen, weil sonst
!     beim Ruecksprung (s.u.) Konstellationen verloren gehen. Standard-
!     werte fuer Pyramiden: 0.5/ 1.0 und fuer die Kammern: 0.1/ 0.2
      if (ipla==1) then
        zfact = 0.5d0; zstep = 1.d0
      else
!       (optimiert fuer alle Kammerzuordnungen)
        zfact = 0.1d0; zstep = 0.2d0
      endif

!.....Startparameter fuer "fitmin"
  220 ifitrun = i0; itin = i0
      imodus = 1; iflag = i0
      ke = 1; indx = 1; nu = i0
      ddx1 = 1.d0; ddx2 = 1.d0
      do i=1,10; test(i) = z0; enddo
      do i=1,5
        xx(i) = z0; yy(i) = z0
      enddo
      xx(1) = zjde; go to 250
  240 call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
  250 tau = (zjde - zjd0)/tmil
      if (ison==5) then
        do i=1,4; iw(i) = iw0(i); enddo
        do i=1,3; w(i) = w0(i); enddo
        do i=1,7; x(i) = x0(i); enddo
        do i=4,6; x(i) = x(i) * pidg; enddo
      endif
      inum(1) = inum(1) + 1
```

```
!.....Variante 1 (VSOP87D, Kurzversion aus "Meeus")
      if (imod==1) then
        do i=1,9; call vsop1(i,tau,resu); re(i) = resu; enddo
      endif

!.....Variante 2 (VSOP87A/C, Vollversion)
      if (imod==2) then
        do i=1,3; ii = 3*(i-1)
          call vsop2(zjde,ivers,i,md,ix,prec,lu,r,ierr,rku)
          do j=1,3; re(ii+j) = rku(j); enddo
        enddo
      endif

!.....Koordinaten-Transformation und Bestimmung von F-pos
      call kartko(ison)
      call relpos(ipla,ison,ijd,iek,iekk,ika)
      if (ison==5) yy(indx) = xyr(36)

! . . zjde so lange erhoehen, bis relativer Fehler nicht mehr steigt.
!c    write(6,'('' zjde,irestart,xyr(36),dwi,imod = '',f18.7,i3, &
!c      & 2f9.3,i3)') zjde,irestart,xyr(36),dwi,imod
      if (xyr(36)>10.d0) imod = 1
      if (irestart==1) then
        if (xyr(36)>x36) then
          go to 290
        else
          zjdelim = zjde
        endif
      endif
      irestart = i0

! . . Bedingung zum Aufruf von fitmin pruefen
      if (xyr(36)>dwi.and.ifitrun==i0) go to 290
      if (ikomb==1) imod = 2

! . . Minimierung des relativen Fehlers F-pos mit "fitmin"
      ifitrun = 1; imodus = 1
      if (ddx1<dfe.or.ddx2<dfe) imodus = 2
      call fitmin(imod,imodus,iaph,ke,xx,yy,eep,step,nu,iflag, &
             ddx1,ddx2,test,itin,indx,ix)
      zjde = xx(indx); if (ke==1) go to 240
      irestart = 1

! . . verhindert, dass fitmin endlos ins vorherige Minimum faellt
      if (dabs(zjde-zjdevor)<=0.1d0) then
         zjde = zjdelim; go to 290
      endif
      zjdevor = zjde

!.....Hauptbedingung pruefen (ison = 5) . . . . . . . . . . . . . . . .
      if (xyr(36)>=dwikomb) go to 290
      inum(2) = inum(2) + 1

! . . Sonnenposition und Output
      call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
      call konst(iak,kon)
      call sonpos(ison,iek,ix,rp(3,1),rp(3,2),rp(3,3), &
                  rcm,dmi,iter,iw,ke,mfit,nfit,f,x,e,w,y,z)
      dd = dn
```



```
      if (iek==2 .or.iekk==2) dd = ds
      xma = xyr(35) * 1.d-9
      if (ipla==1) xma = xyr(35) * 1.d-7
      call reduz(x(5),1,i0)
      do iu=ix,6,5
        if (iout==3) then
          if (ipla==1) then
            write(iu,405)kon,iak,zjahr,delt,x(5)*gdpi,xma, &
              (xyr(30+i),i=1,3),dd,xyr(36)
          else
            write(iu,406)kon,iak,zjahr,delt,x(5)*gdpi,xma, &
              (xyr(30+i),i=1,3),dd,xyr(36)
          endif
        else
          if (ipla==1) then
            write(iu,407)kon,iak,zjde,zjahr,ke,iw(3), &
              (xyr(30+i),i=1,4),dd,xyr(36)
          else
            write(iu,408)kon,iak,zjde,zjahr,ke,iw(3), &
              (xyr(30+i),i=1,4),dd,xyr(36)
          endif
        endif
      enddo
!h    call histogramm(xyr(36),ihis)  !h

! . . Standardruecksprung
  290 zjump = xyr(36)*zfact + zstep
      zjde = zjde + zjump
      x36 = xyr(36)
      if (zjde<=zjdemax) go to 220

!-----Ende der 2. Hauptschleife (freier Zeitpunkt)--------------------
      go to 400

!=====================================================================
!------------------------- 3. Hauptschleife -------------------------
!=====================================================================

!-----3. Hauptschleife (Suche von Linearkonstellationen)--------------
!     Syzygium von Sonne, Merkur, Venus, Erde und Mars,
!     sowie Bestimmung der Transite von Merkur und Venus.

!     "zfact" und "zstep" wie in 2. Hauptschleife (nicht zu gross)
  300 zfact = 0.025d0 * (1.d0 + (21.d0-dwi)/20.d0)
      if (dwi>=21.d0) zfact = 0.025d0
      zstep = 0.01d0
      sz = (1.d0 + 10.d0*zfact)
      zjde = zjdemin
      dfd = 5.d0; dfe = 0.5d0
      izp = 1; icv = 0
  310 zjdestep = zjde
      if (ilin==2 .and.inum(0)>1 .and.iop0/=-803) dfd = 0.02d0
      call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
      ik = idnint(ak)
      inum(0) = inum(0) + 1
      if (ilin>=3) itransit = i0
      do i=1,2; tra(i) = ' '; enddo
      if (ison==5) ifitrun = i0
      if (ilin<=2) ifitrun = 1
```

```
!.....Startparameter fuer "fitmin", "sekante" und "ringfit"
  320 if (ison==5) then
        iflag = i0; ke = 1
        indx = 1; nu = i0
        ddx1 = dfd
        ddx2 = ddx1; itin = i0
        do i=1,10; test(i) = z0; enddo
        do i=1,5
          xx(i) = z0
          yy(i) = z0
        enddo
        xx(1) = zjde
      endif
      go to 340
  330 zjde = xx(indx)
      call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)
  340 time = (zjde - zjd0)/tcen
      tau  = (zjde - zjd0)/tmil
      inum(1) = inum(1) + 1

!.....Variante 1 (VSOP87D, Kurzversion aus "Meeus")
      if (imod==1) then
        do i=1,12; call vsop1(i,tau,resu); re(i) = resu; enddo
        if (ilin<=2) then
          call kartko(ison)
          do i=1,9; rk(i) = xyr(i); enddo
        endif
      endif

!.....Variante 2 (VSOP87A/C, Vollversion)
  350 if (imod==2) then
        do i=il(1),il(2),il(3); ii = 3*(i-1)
          call vsop2(zjde,ivers,i,md,ix,prec,lu,r,ierr,rku)
          do j=1,3
            re(ii+j) = rku(j)
            if (ilin<=2) rk(ii+j) = r(j)
          enddo
        enddo
      endif

!.....Variante 3 (Keplersche Gleichung, Polynome 3. Grades nach VSOP82)
      if (imod==3) then
        do i=1,4; ii = 6*i
          call vsop3(lv,i,ix,ir,time,res)
          if (ir/=i0) go to 500
          re(25+ii) = res(1);  re(28+ii) = res(5)
          re(26+ii) = res(2);  re(29+ii) = res(4)
          re(27+ii) = res(3);  re(30+ii) = res(6)
          if (i<=4) re(3*i-2) = res(11)
        enddo
      endif

!.....Korrelation der Positionen pruefen
      ic = i0
      iwo = i0
      df(1) = re(1)-re(4);   df(2) = re(1)-re(7)
      df(3) = re(1)-re(10);  df(4) = re(4)-re(7)
      df(5) = re(4)-re(10);  df(6) = re(7)-re(10)
      do i=1,6; call reduz(df(i),i0,i0); enddo
```



```
      if (ilin==3) difm = dmax1(dabs(df(1)),dabs(df(2)),dabs(df(4)))
      if (ilin==4) difm = dmax1(dabs(df(1)),dabs(df(2)),dabs(df(3)), &
          dabs(df(4)),dabs(df(5)),dabs(df(6)))
      if (isep==1) then
        if (itransit==1) difm = df(2)
        if (itransit==2) difm = df(4)
      else
        if (itransit==1 .or.itransit==2) then
          call sepa(itransit,2,rk,sep1)
          difm = dabs(sep1)
        endif
      endif
      if (ison==5) yy(indx) = difm
! . . Test-Ausdruck (--> !t)
!t    difr = re(7)-re(1)
!t    call reduz(difr,i0,i0)
!t    do iu=ix,6,5; write(iu,'(''imod,ifit,dt,Le-Lm,jde,difm = '',2i2,&
!t     &f5.1,f6.1,f18.7,f13.7)')imod,ifitrun,step,difr,zjde,difm;enddo

!.....Hauptbedingung pruefen. . . . . . . . . . . . . . . . . . . . . .
      if (difm>dwi.and.ifitrun/=1) go to 370
! . . Ruecksprung fuer ikomb = 1
      if (ikomb==1 .and.imod==1 .and.ilin>=3) then
        ifitrun = 1; imod = 2
        dwi = dwikomb
        go to 350
      endif

! . . Minimierung des Gesamtwinkels difm mit "fitmin" fuer ison = 5
!     (Das heisst, "ison" hat hier eine andere Funktion und bedeutet
!     Minimumsuche.)
      if (ison==5) then
        ifitrun = 1; step = 1.d0
        if (ilin>=3 .and.itransit==i0) then
          call fitmin(imod,1,iaph,ke,xx,yy,e(1),step,nu, &
          iflag,ddx1,ddx2,test,itin,indx,ix); zjde = xx(indx)
        endif
        if (itransit==1 .or.itransit==2) then
          if (isep==1) then
            xj2 = xx(indx); yy2 = yy(indx); indx = 2
            call ringfit(xj1,xj2,xj3,yy1,yy2,yy3, &
              1.d-6,1.d-2,nu,50,ix,ke)
            xx(2) = xj2; zjde = xj2
          else
            eep = e(1)
            if (ikomb==1 .and.imod==1 .and.isep>=3) eep=1.d2*e(1)
            imodus = 1
            if (ddx1<dfe.or.ddx2<dfe) imodus = 2
             call fitmin(imod,imodus,iaph,ke,xx,yy,eep,dfd,nu, &
              iflag,ddx1,ddx2,test,itin,indx,ix)
              zjde = xx(indx)
          endif
        endif
        if (ke==1 .or.(isep==1 .and.ke==5)) go to 330
      endif

! . . Spezialtest fuer ikomb = 0 (imod = 1, 3)
!     Anmerkung: Aufgrund der Zeitschritte (1 Tag) ist es moeglich,
!     dass das Minimum des Winkelintervalls (difm) fuer die eklipti-
```

```
!     kalen Laengen der Planeten genau zwischen zwei Zeitpunkten er-
!     reicht wird. Falls die Schwelle (dwi0) so knapp unterschritten
!     wird, dass sie an den Zeitpunkten davor und danach schon wieder
!     ueberschritten wird, wuerde das Ereignis verloren gehen. Des-
!     halb wird die Schwelle (dwi) zuvor um 1 Grad erhoeht, dann das
!     Winkelintervall minimiert und anschliessend geprueft, ob die
!     urspruengliche Schwelle (dwi0) unterschritten wurde.
      if (ikomb==i0.and.ilin>=3) then
        if (difm<=dwi0) go to 360
        go to 370
      endif
! . . Gegebenenfalls Sprung von der oberen zur unteren Konjunktion.
!     Bei Minimierung der Winkelseparation (isep 2,3,4) wuerden ab
!     einem gewissen Zeitpunkt nur noch obere Konjunktionen berech-
!     net werden. Das wird durch die folgende if-Abfrage behoben.
  360 if (isep>=2 .and.((itransit==1 .and.dabs(df(2))>170.d0) &
          .or.(itransit==2 .and.dabs(df(4))>170.d0))) then
          zjde = zjde + tsy*0.5d0
          go to 320
      endif
      if (ikomb/=1 .or.(ikomb==1 .and.(difm<=dwikomb.or. &
          ilin<=2))) then
        if (itransit==i0.and.nurtr==1) inum(2) = inum(2) + 1
        ic = 1
        if (ic==1 .and.icv==0 .and.ison/=5 .and.ilin>=3) then
          inum(3) = inum(3) + 1
          do iu=ix,6,5
            write(iu,'(i12,''. syzygy'')') inum(3)
          enddo
        endif
        call konst(ik,kon)
! . . . Pruefen des Transits (nur bei imod = 1, 2)
        if (itran==1 .and.ison==5) then
          if (itransit==i0.or.ilin<=2) call memo(zjde,zjahr, &
              delt,df(1),df(2),df(3),difm,zmem,iak,imem)
          if (itransit==1 .or.itransit==2) then
            call transit(itransit,ikomb,imod,ipla,ilin,iaph,ivers, &
             isep,ical,iuniv,tr,sep1,itt,sep,zjde,id5,da5,dmo5, &
             zjahr,rk,md,ddx1,ddx2,dfd,test,itin,is,irs,ix,pan,sd,sl,&
             iop0,inum)
            tra(itransit) = tr
          endif
! . . . . Ereignis mit Transit und Output
          if ((ilin>=3 .and.itransit==2).or. &
              (ilin<=2 .and.tr/=' ')) then
            if (ikomb==1 .and.imod==1 .and.ilin<=2) then
              imod = 2; go to 320
            endif
            if (nurtr==1 .or.(nurtr==2 .and. &
              (tra(1)/=' '.or.tra(2)/=' '))) then
              if (ilin<=2 .or.nurtr==2) inum(2) = inum(2) + 1
              iwo = 1
              if (ilin>=3) then
                do iu=ix,6,5
                  if (dabs(zmem(5))<1.d-4) then
                    zmem(5) = dabs(zmem(5))
                    write(iu,456)kon,' ',tra(1),tra(2),imem, &
                      (zmem(i),i=1,7)
                  elseif (dabs(zmem(6))<1.d-4) then
```



```
                    zmem(6) = dabs(zmem(6))
                    write(iu,457)kon,' ',tra(1),tra(2),imem, &
                      (zmem(i),i=1,7)
                  else
                    write(iu,455)kon,' ',tra(1),tra(2),imem, &
                      (zmem(i),i=1,7)
                  endif
                enddo
              else
                if (iop0==-803 .and.(zjahr<=-13000.d0 .or. &
                    zjahr>=17000.d0)) go to 390; ts = ' '
                if (tra(ilin)/='M'.and.tra(ilin)/='V') ts=tra(ilin)
                if (iuniv==2) call delta_T(zjde)
                call jdedate(zjde,ical,ida,da,dmo)
                if (ida(3)>=izmin) then
                 do iu=ix,6,5
                   if (izp<=3) call zwizeile(iu,io,zmem(1), &
                     ilin,imod,isep,ical,izp)
                   if ((isep<=3 .and. zmem(1)<-1566122.5d0).or. &
                       (isep==4 .and.(zmem(1)<-1931365.5d0 .or. &
                                      zmem(1)>=5373484.5d0))) then
                    if (isep<=2) then
                      write(iu,458)kon,ts,imem,da(7),dmo,ida(3), &
                      (ida(i),dp,i=4,5),ida(6),(zmem(i),i=3,6),sep,irs
                    else
                     if (isep==3) then
                      if (itt==3) &
                       write(iu,459)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       ((id5(l,i),dp,i=4,5),id5(l,6),l=1,5),sep,sl,irs
                      if (itt==2) &
                       write(iu,461)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       ((id5(l,i),dp,i=4,5),id5(l,6),str2,l=1,3,2), &
                       (id5(5,i),dp,i=4,5),id5(5,6),sep,sl,irs
                      if (itt==1) &
                       write(iu,471)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       str2,str2,(id5(3,i),dp,i=4,5),id5(3,6), &
                       str2,str2,sep,sl,irs
                     else
                      if (itt==3) &
                       write(iu,659)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       ((id5(l,i),dp,i=4,5),id5(l,6),l=1,5),sep,sl, &
                       (pan(i),i=1,5),sd(1),sd(2),irs
                      if (itt==2) &
                       write(iu,661)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       ((id5(l,i),dp,i=4,5),id5(l,6),str2,l=1,3,2), &
                       (id5(5,i),dp,i=4,5),id5(5,6),sep,sl,pan(1), &
                       str3,pan(3),str3,pan(5),sd(1),sd(2),irs
                      if (itt==1) &
                       write(iu,671)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                       str2,str2,(id5(3,i),dp,i=4,5),id5(3,6), &
                       str2,str2,sep,sl,str3,str3,pan(3), &
                       str3,str3,sd(1),sd(2),irs
                     endif
                     if (itt==i0.and.iu==6) inum(2) = inum(2) - 1
                    endif
                   else
                    if (isep<=2) then
                      write(iu,558)kon,ts,imem,da(7),dmo,ida(3), &
                      (ida(i),dp,i=4,5),ida(6),(zmem(i),i=3,6),sep,irs
```

```
                          else
                           if (isep==3) then
                            if (itt==3) &
                             write(iu,559)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             ((id5(l,i),dp,i=4,5),id5(l,6),l=1,5),sep,sl,irs
                            if (itt==2) &
                             write(iu,561)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             ((id5(l,i),dp,i=4,5),id5(l,6),str2,l=1,3,2), &
                             (id5(5,i),dp,i=4,5),id5(5,6),sep,sl,irs
                            if (itt==1) &
                             write(iu,571)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             str2,str2,(id5(3,i),dp,i=4,5),id5(3,6), &
                             str2,str2,sep,sl,irs
                           else
                            if (itt==3) &
                             write(iu,759)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             ((id5(l,i),dp,i=4,5),id5(l,6),l=1,5),sep,sl, &
                             (pan(i),i=1,5),sd(1),sd(2),irs
                            if (itt==2) &
                             write(iu,761)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             ((id5(l,i),dp,i=4,5),id5(l,6),str2,l=1,3,2), &
                             (id5(5,i),dp,i=4,5),id5(5,6),sep,sl,pan(1), &
                             str3,pan(3),str3,pan(5),sd(1),sd(2),irs
                            if (itt==1) &
                             write(iu,771)kon,ts,da5(3,7),dmo5(3),id5(3,3), &
                             str2,str2,(id5(3,i),dp,i=4,5),id5(3,6), &
                             str2,str2,sep,sl,str3,str3,pan(3), &
                             str3,str3,sd(1),sd(2),irs
                           endif
                           if (itt==i0.and.iu==6) inum(2) = inum(2) - 1
                          endif
                         endif
                         if (isep<=2 .and.iu==6) then
                          if (ts=='m'.or.ts=='v') inum(3) = inum(3) + 1
                          if (ts=='C'.or.ts=='c') inum(4) = inum(4) + 1
                         endif
                       enddo
                      else
                        ic = i0; iwo = i0; inum(2) = inum(2) - 1
                      endif
                    endif
                  endif
                endif
                if (itransit==i0.or.ilin<=2) zjde0 = zjde
!t              read(*,*)  !t
! . . . Ereignis ohne Transit-Pruefung (z.B. imod = 3), Output
              else
                do iu=ix,6,5
                  if (dabs(df(2))<1.d-4) then
                    write(iu,456)kon,' ',tra(1),tra(2),ik, &
                      zjde,zjahr,delt,(df(i),i=1,3),difm
                  elseif (dabs(df(3))<1.d-4) then
                    write(iu,457)kon,' ',tra(1),tra(2),ik, &
                      zjde,zjahr,delt,(df(i),i=1,3),difm
                  else
                    write(iu,455)kon,' ',tra(1),tra(2),ik, &
                      zjde,zjahr,delt,(df(i),i=1,3),difm
                  endif
                enddo
```



```
                call memo(zjde,zjahr,delt,df(1),df(2),df(3),difm,zmem, &
                  iak,imem)
              endif
            endif
! . . Ruecksprung fuer Transit-Pruefung
  370       if (itran==1 .and.ison==5 .and.ilin>=3) then
              if (itransit/=i0) zjde = zjde0
              if (ison==5 .and.ic==1 .and.ilin>=3) itransit=itransit+1
              if (itransit==i0.or.itransit>2 .or.ilin<=2) go to 380
              go to 320
            endif

! . . Bedingung fuer Zeitsprung zur Verkuerzung der Rechenzeit
  380       if (ilin>=3 .and.dwin<=21.d0) then
              iflag2 = iflag1; iflag1=i0; if (dabs(df(4))<=dwin) iflag1=1
            endif; ifitrun = i0

! . . Weiterer Output
            do iu=ix,6,5
              if (((ilin<=2 .and.(tra(1)/=' '.or.tra(2)/=' ').and. &
                  ((isep<=2 .or.(isep>=3 .and.itt/=0)).and. &
                  ida(3)>=izmin)).or.(ic==1 .and.ilin>=3)).and. &
                  io==2 .and.iwo==1) then
                if (imod/=3) then
                  call linie(iu,2+ipar); write(iu,57) (zmem(i),i=11,19)
                  write(iu,57) (zmem(i),i=20,22); call linie(iu,2)
                endif
                if (ic==1 .and.imod==3 .and.io==2) call linie(iu,2)
                immin = 1; if (imod==3) immin = 5
                immax = 4; if (ilin>=3) immax = 8
                if (immin<=immax) then
                  do i=immin,immax; ii = 6*i
                    call vsop3(lv,i,ix,ir,time,res); if (ir/=i0) go to 500
                    zmem(25+ii) = res(1);  zmem(28+ii) = res(5)
                    zmem(26+ii) = res(2);  zmem(29+ii) = res(4)
                    zmem(27+ii) = res(3);  zmem(30+ii) = res(6)
                  enddo
                endif
                if (ilin<=2) then
                  if (ivers==3) then
                    write(iu,'(''  ascending node (M/V/E/Ma): '',2f12.6, &
                     & ''      ---   '',f12.6)')zmem(34),zmem(40),zmem(52)
                  else
                    write(iu,'(''  ascending node (M/V/E/Ma): '',4f12.6)') &
                      (zmem(28+6*i),i=1,4)
                  endif
                  write(iu,'(''  inclination i  (M/V/E/Ma): '',4f12.6)') &
                    (zmem(29+6*i),i=1,4)
                  write(iu,'(''  perihelion pi  (M/V/E/Ma): '',4f12.6)') &
                    (zmem(30+6*i),i=1,4)
                else
                  do i=31,78; re(i) = zmem(i); enddo
                  call elements(iu,ivers,pla)
                endif
                call linie(iu,1+ipar)
              endif
            enddo
  390       if (ikomb==1 .and.imod==2) then; imod = 1; dwi = dwi0; endif
```

```
      ! . . Bedingter groesserer Zeitsprung
            if (ilin<=2 .or.(dwin<=21.d0 .and.((iflag2==1 .and.iflag1==i0) &
               .or.(ison==5 .and.ifitrun==i0.and.(ke==i0.or.ke==3))))) then
              zjde = zjde + tsprung; iflag1 = i0
            else
              zjde = zjdestep
              if (ison==5 .or.(ison/=5 .and.dabs(difm)>dwin*sz)) then
                stepl = difm*zfact + zstep
                if (ic==1) stepl = 0.9d0*ymer
                zjde = zjde + stepl
              else
                zjde = zjde + step
              endif
            endif
            icv = ic
            if (zjde<=zjdemax) go to 310
      ! . . Ergaenzung (Tabellenkopf fuer Transit-Test mit inum(2)=0)
            if (ilin<=2 .and.inum(2)==0) then
              do iu=ix,6,5
                call zwizeile(iu,io,zmem(1),ilin,imod,isep,ical,izp)
              enddo
            endif

      !-----Ende der 3. Hauptschleife (Linearkonstellation, Transit)---------

      !====================================================================
      !---------------------- Ende der Hauptschleifen ----------------------
      !====================================================================

        400 do iu=ix,6,5; if (io/=2) call linie(iu,1+ipar); enddo

      ! . . Ruecksprung bei Option -803 und Speichern von "inser-2.t"
            if (iop0==-803) then
              if (ilin==1) then
                ilin = 2
                zmin = -30000; zmax = 30000
                go to 10
              endif
              call save_ser
            endif

      !-----Endzeilen
            call cpu_time(zib)
            call comtime(zia,zib,ihour,imin,sec)
            do iu=ix,6,5
              call endzeile(ipla,imod,ilin,iaph,isep, &
                ison,ijd,ipos,iu,inum,ihour,imin,sec,is12,iop0)
      !h      if (ipla<=2.and.imod<=2.and.ison>=3) then                 !h
      !h        write(iu,'(7x,a24,a33)') 'Frequency of deviations ', &  !h
      !h          & ' Fpos(0 to 5%) in steps of 0.05%:'                 !h
      !h        call linie(iu,1)                                        !h
      !h        do i=0,4                                                !h
      !h          write(iu,'(2(3x,10i3))') (ihis(j+i*20),j=1,20)        !h
      !h        enddo; call linie(iu,1); write(iu,*)                    !h
      !h      endif                                                     !h
              close(iu)
            enddo
        500 continue
```



```
      !-----Ende des Hauptprogramms-------------------------------------------
            stop
         54 format(1x,3f9.6,3(a1,f7.6),3f9.6)
         55 format(1x,a2,i7,f14.5,f10.3,f8.3,4f8.3,f6.1)
         56 format(1x,a2,i7,f15.5,f11.3,f9.3,4f8.3,a2)
         57 format(1x,3(f9.4,f8.4,f9.6))
         65 format(1x,a2,i7,f10.3,3f8.3,3f7.1,f5.1,a2,f7.3)
         67 format(1x,a2,i7,f10.3,3f8.3,2f7.1,a7,f5.1,a2,f7.3)
         85 format(1x,a2,i7,f10.3,3f8.3,3f7.2,f5.2,a2,f7.3)
        165 format(1x,a2,i7,f10.3,2f8.3,i3,i4,3f7.1,f6.1,a2,f7.3)
        184 format(1x,a2,i7,f10.3,2f8.3,i3,i4,3f7.2,f6.2,a2,f7.3)
        255 format(1x,f14.5,f7.1,f7.2,f7.3,f7.2,3f7.1,f6.1,a2,f7.3)
        256 format(1x,a2,i7,f10.3,f8.2,f7.3,f7.2,4f7.1,a2,f7.3)
        275 format(1x,f14.5,f7.1,f7.2,f7.3,f7.2,3f7.2,f6.2,a2,f7.3)
        276 format(1x,a2,i7,f10.3,f8.2,f7.3,f8.2,3f7.2,f6.2,a2,f7.3)
        365 format(1x,a3,i8,f13.3,f12.0,i6,1x,3f7.1,f5.1,a2,f7.3)
        366 format(1x,a3,i8,f13.3,f12.0,i2,i4,3f7.1,f6.1,a2,f7.3)
        367 format(1x,a3,i8,f13.3,f12.0,i6,1x,2f7.1,a7,f5.1,a2,f7.3)
        384 format(1x,a3,i8,f13.3,f12.0,i6,1x,3f7.2,f5.2,a2,f7.3)
        386 format(1x,a3,i8,f13.3,f12.0,i2,i4,3f7.2,f6.2,a2,f7.3)
        405 format(1x,a2,i7,f11.3,f8.3,f9.3,f9.4,f8.1,2f7.1,1x,a2,f7.3)
        406 format(1x,a2,i7,f11.3,f8.3,f9.3,f9.4,f8.2,2f7.2,1x,a2,f7.3)
        407 format(1x,a2,i7,f15.5,f11.3,i3,i4,f8.1,2f7.1,f6.2,a2,f6.3)
        408 format(1x,a2,i7,f15.5,f11.3,i3,i4,f8.2,2f7.2,f6.2,a2,f6.3)
        455 format(1x,a2,3a1,i7,f15.5,f11.3,5f8.3)
        456 format(1x,a2,3a1,i7,f15.5,f11.3,2f8.3,f6.1,f10.3,f8.3)
        457 format(1x,a2,3a1,i7,f15.5,f11.3,3f8.3,f6.1,f10.3)
        458 format(1x,a2,a1,i7,f5.0,a5,i6,i3,2(a1,i2),4f8.3,f7.1,i5)
        459 format(1x,a2,a1,f4.0,a5,i6,i3,2(a1,i2),4(i4,2(a1,i2)),f7.1,a1,i4)
        461 format(1x,a2,a1,f4.0,a5,i6,i3,2(a1,i2),2(a10,i4,2(a1,i2)), &
                   f7.1,a1,i4)
        471 format(1x,a2,a1,f4.0,a5,i6,1x,a8,2x,a8,i4,2(a1,i2),2(2x,a8), &
                   f7.1,a1,i4)
        558 format(1x,a2,a1,i7,f5.0,a5,i5,i4,2(a1,i2),4f8.3,f7.1,i4)
        559 format(1x,a2,a1,f4.0,a5,i5,5(i4,2(a1,i2)),f7.1,a1,i3)
        561 format(1x,a2,a1,f4.0,a5,i5,i4,2(a1,i2),2(a10,i4,2(a1,i2)), &
                   f7.1,a1,i3)
        571 format(1x,a2,a1,f4.0,a5,i5,2a10,i4,2(a1,i2),2a10,f7.1,a1,i3)
        659 format(1x,a2,a1,f4.0,a5,i6,5(i4,a1,i2,a1,i2),f7.1,1x,a1, &
                   3x,5f8.2,4x,2f8.2,i6)
        661 format(1x,a2,a1,f4.0,a5,i6,i4,2(a1,i2),2(a10,i4,2(a1,i2)), &
                   f7.1,1x,a1,3x,f8.2,a8,f8.2,a8,f8.2,4x,2f8.2,i6)
        671 format(1x,a2,a1,f4.0,a5,i6,2a10,i4,2(a1,i2),2a10,f7.1,1x,a1, &
                   3x,2a8,f8.2,2a8,4x,2f8.2,i6)
        759 format(1x,a2,a1,f4.0,a5,i5,1x,5(i4,a1,i2,a1,i2),f7.1,1x,a1, &
                   3x,5f8.2,4x,2f8.2,i6)
        761 format(1x,a2,a1,f4.0,a5,i5,1x,i4,2(a1,i2),2(a10,i4,2(a1,i2)), &
                   f7.1,1x,a1,3x,f8.2,a8,f8.2,a8,f8.2,4x,2f8.2,i6)
        771 format(1x,a2,a1,f4.0,a5,i5,1x,2a10,i4,2(a1,i2),2a10,f7.1,1x,a1, &
                   3x,2a8,f8.2,2a8,4x,2f8.2,i6)

      ! . . Ausgabe einer groesseren Stellenanzahl zur Feinabstimmung bzw.
      !     Minimierung von F[%] fuer die Schnellstart-Optionen 4 und 9.
      !     Dies wurde verwendet fuer Buch 1.
      !     Suche in der Umgebung des Merkur-Aphels bzw. Merkur-Perihels
      !f255 format(1x,f14.5,f8.2,f7.2,f8.4,f6.1,3f7.1,f5.1,a2/65x,f14.8)
      !f275 format(1x,f14.5,f8.2,f7.2,f7.3,f7.2,3f7.2,f5.1,a2/65x,f14.8)
            end program P4
```

```
      subroutine inputdata(ipla,ilin,imod,imo4,ikomb,io,lv,ivers, &
        itran,isep,iuniv,ical,ika,iaph,iamax,step,ison,ihi,irb,ijd, &
        zmin,zmax,ak,zjdel,dwi,dwikomb,dwi2,dwi3,nurtr,iek,iop0,iout)
!-----Inputdaten und Programmstart------------------------------------
      implicit double precision (a-h,o-z)
      character(36) :: com
      data ita/0/ ! pre-init.
      iy = 6; ipla = 1; itran = 1
      io = 0; ire = 0; z0 = 0.d0
      write(iy,'(///29x,23(''-''))')
      write(iy,'(30x,''PLANETARY CORRELATION'')')
      write(iy,'(30x,''Program P4, June 2015'')')
      write(iy,'(29x,23(''-''))')

! . . Schnellstart-Menue
      write(iy,'(/7x,a16,9x,a18,7x,a16/5x,70a1/5(6x,2(a19,6x),a18/), &
      & 5x,70a1)') &
     'pyramids of Giza','chambers, Great P.','transits, syzygy', &
      ('-',i=1,70), &
     '3D Mer. at aph. (1)','3D Mer. at per. (6)','Mercury tr.   (11)',&
     '2D Mer. at aph. (2)','Keplers equ.    (7)','Venus tr.     (12)',&
     'const. 12, 3088 (3)','const. 12, 3088 (8)','syzygy, 3 pl. (13)',&
     '1.5 days, 3088  (4)','1.5 days, 3088  (9)','syzygy, 4 pl. (14)',&
     'near aphelion   (5)','F minimized    (10)','TYMT-test     (15)',&
      ('-',i=1,70)
      do
        do
          write(iy,'(8x,a10,3x,a20,3x,a26)',advance='no')'info (111)',&
           'detailed options (0)','(0..15 or book options) : '
          read(*,*,iostat=iox) iop0
          if (iox==0) exit
          call emes(ire,com,dm)
        enddo; iop=iop0
        if (iop==0) then; write(iy,*); go to 10; endif
        if (iop==111) then; call info; iout=4; return; endif

! . . . Verborgene Optionen fuer Tabellen aus beiden oben genannten
!       Buechern, s.a. im Programmkopf unter "Neue Optionen, b)"
        if ((iop>=0 .and.iop<=15).or. &
!         1. "Pyramiden und Planeten", Tab. 39-51
          (iop>=390 .and.iop<=392).or.(iop>=400 .and.iop<=402).or. &
          (iop>=410 .and.iop<=432).or.(iop>=440 .and.iop<=442).or. &
          iop==450 .or. &
          (iop>=460 .and.iop<=461).or.(iop>=470 .and.iop<=471).or. &
          (iop>=480 .and.iop<=481).or.(iop>=490 .and.iop<=492).or. &
          (iop>=500 .and.iop<=502).or.(iop>=510 .and.iop<=512).or. &
          (iop>=517 .and.iop<=519).or. &
!         2. Buch 2, Tab. 17-38 ausser 34
          iop==170 .or. &
          iop==180 .or.(iop>=190 .and.iop<=192).or.iop==200 .or. &
          iop==210 .or.iop==220 .or.iop==230 .or.iop==240 .or. &
          iop==250 .or.iop==260 .or.iop==270 .or.iop==280 .or. &
          iop==290 .or.iop==300 .or.iop==310 .or.iop==320 .or. &
          iop==330 .or.iop==350 .or.iop==351 .or.iop==360 .or. &
          iop==361 .or.(iop>=370 .and.iop<=372).or.iop==380 .or. &
          iop==381 .or.iop==385 .or.iop==999 .or.iop==-803) exit
        ire = 1; call emes(ire,com,dm)
      enddo
```



```
! . . Auswertung der eingegebenen Option
      if (iop<0 .or.iop>15) then
        id = mod(iop,10); ita = (iop-id)/10

!       Buch 1 (Parameter fuer Datei "inparm.t")
        if (ita==39) iop = 16 + id
        if (ita==40) iop = 19 + id
        if (ita==41 .or.ita==42) then
          if (id<=6) iop = 22 + id
          if (id==7) iop = 3
          if (id>=8) iop = 21 + id
        endif
        if (ita==43) iop = 31 + id
        if (ita==44) iop = 23 + 3*id
        if (ita==45) iop = 2
        if (ita==46 .or.ita==47) iop = 34 + id
        if (ita==48) iop = 36 + id
        if (ita==49 .and.id==0) iop = 3
        if (ita==49 .and.id>=1) iop = 28 + id
        if (ita==50 .and.id==0) iop = 1
        if (ita==50 .and.id>=1) iop = 37 + id
        if (ita==51 .and.id<=2) iop = 40 + id
        if (ita==51 .and.id>=7) iop = 64 + id

!       Buch 2 (Parameter fuer Datei "inparm.t")
        if (ita==17 .or.ita==18) iop = 26 + ita
        if (ita==19) iop = 45 + id
        if (ita==20 .or.ita==21) iop = 28 + ita
        if (ita==22) iop = 50
        if (ita==23 .or.ita==24) iop = 28 + ita
        if (ita==25) iop = 8
        if (ita==26) iop = 3
        if (ita==27 .or.ita==28) iop = 26 + ita
        if (ita==29) iop = 14
        if (ita>=30 .and.ita<=33) iop = 25 + ita
        if (ita==35) iop = 59 + id
        if (ita==36) iop = 61 + id
        if (ita==37) iop = 63 + id ! Bei iop0=371, 372 s.a. "aphelko".
        if (ita==38 .and.id<=1) iop = 66 + id
        if (ita==38 .and.id==5) iop = 68
        if (iop0==-803) iop = 69 ! Erzeugung der Datei "inser-2.t"
      endif

! . . Einlesen der Parameter aus "inparm.t"
      call inputfile(ipla,ilin,imod,imo4,ikomb,io,lv,ivers, &
        itran,isep,iuniv,ical,ika,iaph,iamax,step,ison,ihi,irb,ijd, &
        zmin,zmax,ak,zjdel,dwi,dwikomb,dwi2,dwi3,nurtr,iek,iop,1,iout)
      return

!.....Menues fuer Einzeleingabe der Parameter.........................
! . . Planetenpositionen
   10 do
        write(iy,'(''  Constell.  pyr.(1),  chamb.(2),  lin.(3)  : ''&
          & )',advance='no')
        read(*,*,iostat=iox) ipla
        if (ipla>=1 .and.ipla<=3 .and.iox==0) exit
        call emes(ire,com,dm)
      enddo
```

```
      ! . . Linearkonstellation, Transite
            ilin = 4
            if (ipla==3) then
              do
               write(iy,'(''  Tr. Mer.(1), Ven.(2), 3-co.(3), 4-co.(4)  : ''&
                 & )',advance='no')
               read(*,*,iostat=iox) ilin
               if (ilin>=1 .and.ilin<=4 .and.iox==0) exit
               call emes(ire,com,dm)
              enddo
            endif

      ! . . VSOP, Theorie-Variante
      !     Es erfolgt hier eine Aenderung des Parameters 'imod' (s.u.).
      !     Eingabe :  VSOP87  Kombi.(1), Kurzv.(2), Kepl.(3), Vollv.(4)
      !     intern  :  VSOP87  Kurzv.(1), Vollv.(2), Kepl.(3)
            ikomb = 0
            do
              if (ipla/=3) then
               write(iy,'(''  VSOP87     combi. (1), short version (2),''/ &
                         &''         Kepl. equ. (3),  full version (4)  : ''&
                 & )',advance='no')
               read(*,*,iostat=iox) imod
               if (imod>=1 .and.imod<=4 .and.iox==0) exit
              else
               if (ilin>=3) then
                write(iy,'(''  VSOP87  combi.(1), short v.(2), '', &
                  & ''Kepl.(3)  : '')',advance='no')
                read(*,*,iostat=iox) imod
                if (imod>=1 .and.imod<=3 .and.iox==0) exit
               else
                 write(iy,'(''  VSOP87-version  full v.(1),  '', &
                  & ''short v.(2)  : '')',advance='no')
                read(*,*,iostat=iox) imod
                if (imod>=1 .and.imod<=2 .and.iox==0) exit
               endif
              endif
              call emes(ire,com,dm)
            enddo
      !     Aendern des Parameters "imod"
      !     (imo4 wird eingefuehrt, da imod wechselt, falls ikomb = 1 ist.)
            imo4 = 0
            if (imod==1) ikomb = 1
            if (imod==2) imod = 1
            if (imod==4) then
              imod = 2; imo4 = 1
            endif

      ! . . Version von VSOP87
      !     (Bei Transits u. J2000: geringe Abw. zu Meeus => keine Option
      !     bzw. ipla <= 2.)
            lv = 1; ivers = 3
            if (imod/=1 .or.(imod==1 .and.ikomb==1 .and.ipla<=2)) then
              do
               write(iy,'(''  System   ecl. of epoch (1),  J2000.0 (2)  : ''&
                 & )',advance='no')
               read(*,*,iostat=iox) lv
               if ((lv==1 .or.lv==2).and.iox==0) exit
               call emes(ire,com,dm)
```



```
              enddo
              if (lv==2) ivers = 1
            endif

      ! . . Merkur- und Venustransite vor Sonne pruefen bei VSOP-Vollversion
      !     (Diese Option wird nicht mehr abgefragt, da nach Optimierung der
      !     VSOP87-Routine der Geschwindigkeitsvorteil durch Weglassen der
      !     Transit-Pruefung nur noch gering ist.)
      !c    if (ipla==3.and.ikomb==1.and.ilin>=3) then
      !c      do
      !c       write(iy,'(''  Check planetary transit  yes (1), no (2)  : ''&
      !c         & )',advance='no')
      !c       read(*,*,iostat=iox) itran
      !c       if ((itran==1.or.itran==2).and.iox==0) exit
      !c       call emes(ire,com,dm)
      !c      enddo
      !c      if (itran==2) io = 1
      !c    endif

      ! . . Transit-Pruefung bei gleicher ekl. Laenge, minimaler Separation
      !     oder Berechnung der Phasen, optional mit Positionswinkeln
            isep = 1
            if (itran==1 .and.ilin<=2) then
              do
               write(iy,'(''  Date  equ.L.(1), nearest (2), phases (3)''/ &
                         & ''            phases and position angles (4)  : ''&
                 & )',advance='no')
               read(*,*,iostat=iox) isep
               if (isep>=1 .and.isep<=4 .and.iox==0) exit
               call emes(ire,com,dm)
              enddo
            endif

      ! . . Julian/Gregorian calendar: Automatic choice of calender or
      !     only Gregorian calendar
            ical = 0
            do
              write(iy,'(''  Calendar  only Greg. (1), Jul./Greg. (2)  : ''&
                & )',advance='no')
              read(*,*,iostat=iox) ical
              if ((ical==1 .or.ical==2).and.iox==0) exit
              call emes(ire,com,dm)
            enddo

      ! . . Terrestrial Time bzw. Universal Time
            iuniv = 1
            if (itran==1 .and.ilin<=2 .and.isep>=3) then
              do
               write(iy,'(''  Time system        JDE/ TT (1),  UT (2)  : ''&
                 & )',advance='no')
               read(*,*,iostat=iox) iuniv
               if ((iuniv==1 .or.iuniv==2).and.iox==0) exit
               call emes(ire,com,dm)
              enddo
            endif

      ! . . Zuordnung der Planeten Erde (E), Venus (V) und Merkur (M) zu
      !     Koenigs-, Koeniginnen- und Felsenkammer in dieser Reihenfolge
            ika = 0
```

```
      if (ipla==2 .and.imod/=3) then
        do
         write(iy,'(''  Planets  E-V-M (1), E-M-V (2), V-E-M (3),''/ &
                & ''           V-M-E (4), M-E-V (5), M-V-E (6)  : ''&
          & )',advance='no')
         read(*,*,iostat=iox) ika
         if (ika>=1 .and.ika<=6 .and.iox==0) exit
         call emes(ire,com,dm)
        enddo
      endif

! . . Zeitpunkte im/um Aphel bzw. Perihel oder freier Zeitpunkt
      iaph = 1
      iamax = 0
      step = 24.d0
      if (ipla/=3) then
       do
        if (imod<=2 .and.ikomb==0 .and.imo4==0) then
         write(iy,'(''  Passage aph./per. area of aph./per. free''/ &
                & ''          (1)  (2)          (3)  (4)   (5)  : ''&
          & )',advance='no')
         read(*,*,iostat=iox) iaph
         if (iaph>=1 .and.iaph<=5 .and.iox==0) exit
        elseif (imod<=2 .and.ikomb==1 .and.imo4==0) then
         write(iy,'(''  Passage   aph. (1),  per. (2),  free (5)  : ''&
          & )',advance='no')
         read(*,*,iostat=iox) iaph
         if ((iaph==1 .or.iaph==2 .or.iaph==5).and.iox==0) exit
        elseif (imod<=2 .and.ikomb==0 .and.imo4==1) then
         write(iy,'(''  Passage    aph./ per.  area of aph./ per.''/ &
                & ''             (1)   (2)           (3)   (4)  : ''&
          & )',advance='no')
         read(*,*,iostat=iox) iaph
         if (iaph>=1 .and.iaph<=4 .and.iox==0) exit
        else
         write(iy,'(''  Passage    aphelion (1),  perihelion (2)  : ''&
          & )',advance='no')
         read(*,*,iostat=iox) iaph
         if ((iaph==1 .or.iaph==2).and.iox==0) exit
        endif
        call emes(ire,com,dm)
       enddo
       if (iaph==3 .or.iaph==4) then
        do
         write(iy,'(''  Steps per Mercury passage : '')',advance='no')
         read(*,*,iostat=iox) iamax
         if (iamax>0 .and.iamax<=200000 .and.iox==0) exit
         call emes(ire,com,dm)
        enddo
        do
         write(iy,'(''  Step width  (hours, real) : '')',advance='no')
         read(*,*,iostat=iox) step
         if (step>z0.and.iox==0) exit
         call emes(ire,com,dm)
        enddo
        if (imod==2) io = 1
       endif
      endif
```



```
! . . Sonnenposition
      ison = 1
      if (ipla/=3) then
       do
        if (ipla==1 .and.iaph<=2) then
         if (imod<=2) then
         write(iy,'(''  Sun pos.  Myk.(1),  Chefr.(2),  free (3)  : ''&
          & )',advance='no')
         else
         write(iy,'(''  Sun pos. south of   Myk.(1),   Chefr.(2)  : ''&
          & )',advance='no')
         endif
         read(*,*,iostat=iox) ison
        else
         if (imod<=2) ison = 3
        endif
        if (((imod<=2 .and.ison>=1 .and.ison<=3).or. &
            (imod==3 .and.(ison==1 .or.ison==2))).and.iox==0) exit
        call emes(ire,com,dm)
       enddo
      endif

! . . Freie Sonnenposition, Berechnung 2- oder 3-dimensional
      if (iaph==5) ison = 5
      if (ison==3) then
       do
        if (ipla==1) then
         write(iy,'(''  Sun     2D (1), 3D/SLE (2), 3D/FITEX (3)  : ''&
           & )',advance='no')
        else
         write(iy,'(''  Sun (three-dim.):    SLE (2),  FITEX (3)  : ''&
           & )',advance='no')
        endif
        read(*,*,iostat=iox) ison2
        if (((ipla==1 .and.ison2>=1 .and.ison2<=3).or. &
            (ipla==2 .and.(ison2==2 .or.ison2==3))).and.iox==0) exit
        call emes(ire,com,dm)
       enddo
       if (ison2==2) ison = 4
       if (ison2==3) ison = 5
      endif

! . . Hoehenlage der Pyramiden-Grundflaechen bzw. der -Schwerpunkte
      ihi = 0
      if (ipla/=3 .and.ison>=4) then
       do
        if (ipla==1) then
         write(iy,'(''  z-coord.    base (1),  C-M (2),  top (3)  : ''&
           & )',advance='no')
        else
         write(iy,'(''  Wall      east (1), middle (2), west (3)  : ''&
           & )',advance='no')
        endif
        read(*,*,iostat=iox) ihi
        if (ihi>=1 .and.ihi<=3 .and.iox==0) exit
        call emes(ire,com,dm)
       enddo
      endif
```

```
  ! . . Grundebene Ekliptik, Merkur- oder Venusbahn
        irb = 1
        if (ipla/=3 .and.imod<=2 .and.ison==1) then
         do
          write(iy,'(''  Coord.   ecl.(1),   Mer.(2-4),   Ven.(5)  : '' &
            & )',advance='no')
          read(*,*,iostat=iox) irb
          if (irb>=1 .and.irb<=5 .and.iox==0) exit
          call emes(ire,com,dm)
         enddo
        endif

  ! . . Angabe bzw. Berechnung von JDE
        ijd = 15
        if (ipla/=3 .and.ikomb==0 .and.iaph/=5) then
         do
          if (imod==2 .and.iaph<=2) then
           write(iy,'(''  Constell. (1..14),  k-No. (15),  JDE (0)  : ''&
             & )',advance='no')
          else
           write(iy,'(''  Constell. (1..14),  years (15),  JDE (0)  : ''&
             & )',advance='no')
          endif
          read(*,*,iostat=iox) ijd
          if (ijd>=0 .and.ijd<=15 .and.iox==0) exit
          call emes(ire,com,dm)
         enddo
        endif
        ak = z0
        zmin = z0
        zmax = z0
        if (ijd==15) then
          if (imod==2 .and.iaph<=2 .and.ipla/=3) then
            do
             write(iy,'(''     k (real): '')',advance='no')
             call pcheck(1,ak,2,dm,imod,ire)
             if (ire==0) exit
            enddo
          else
            do
             write(iy,'(''     from  year (real): '')',advance='no')
             call pcheck(1,zmin,1,dm,imod,ire)
             if (ire==0) exit
            enddo
            do
             write(iy,'(''     until year (real): '')',advance='no')
             call pcheck(1,zmax,1,dm,imod,ire)
             if (zmin>=zmax.and.ire==0) then
              call emes(ire,com,dm)
              ire = 1
             endif
             if (ire==0) exit
            enddo
          endif
        endif
        if (ipla==3) then
         step = z0
         if (ilin>=3 .and.ikomb==0) then
          do
```



```
           write(iy,'(''  Step width [hrs] (min.-search 0.) (real)  : ''&
             & )',advance='no')
           read(*,*,iostat=iox) step
           if (step>=z0.and.iox==0) exit
           call emes(ire,com,dm)
          enddo
         endif
        endif
        if (step==z0) ison = 5
        if (ipla==3 .and.step/=z0) io = 1
        zjdel = z0
        if (ijd==0) then
          do
           write(iy,'(''     JDE (real) : '')',advance='no')
           call pcheck(1,zjdel,3,dm,imod,ire)
           if (ire==0) exit
          enddo
        endif

  ! . . Winkelintervall bzw. relativer Fehler
        dwi = z0
        dwi2 = z0
        dwi3 = z0
        dwikomb = z0; dm = 99.99d0
        if (ipla/=3 .and.ijd==15 .and.(imod/=2 .or. &
           (imod==2 .and.(iaph==3 .or.iaph==4)))) then
         if (ikomb==0 .and.iaph/=5) then
          do
           if (ison<=2) then
            if (imod/=3) dm = 10.d0
            write(iy,'(''  Tolerance ecl. long. Venus, Earth (real)'', &
              & ''  : '')',advance='no')
           else
            write(iy,'(''  Max. F-pos at aphelion/ per.  (real) [%]'', &
              & ''  : '')',advance='no')
           endif
           call pcheck(2,dwi,1,dm,imod,ire)
           if (ire==0) exit
          enddo
         else
          do
           if (ison<=2) then
            if (imod/=3) dm = 10.d0
            write(iy,'(''  Tolerance ecl. long. VSOP short   (real)'', &
              & ''  : '')',advance='no')
           else
            if (iaph/=5 .or.(iaph==5 .and.ikomb==1)) then
             write(iy,'(''  Max. F-pos    VSOP short ver. (real) [%]'', &
               & ''  : '')',advance='no')
            else
             write(iy,'(''  Max. F-pos, VSOP short, start fitmin [%]'', &
               & ''  : '')',advance='no')
            endif
           endif
           call pcheck(2,dwi,1,dm,imod,ire)
           if (ire==0) exit
          enddo
          do
           if (ison<=2) then
```

```
                if (imod/=3) dm = 10.d0
                write(iy,'(''       "       "    VSOP full    (real)'', &
                  & ''  : '')',advance='no')
               else
                if (iaph/=5 .or.(iaph==5 .and.ikomb==1)) then
                 write(iy,'(''   "     "      VSOP  full ver. (real) [%]'', &
                   & ''  : '')',advance='no')
                else
                 write(iy,'(''   "     "    VSOP short, final range  [%]'', &
                   & ''  : '')',advance='no')
                endif
               endif
               call pcheck(2,dwikomb,1,dm,imod,ire)
               if (ire==0) exit
              enddo
             endif
             if (iaph==3 .or.iaph==4) then
              do
               write(iy,'(''   "     "   consider without printing [%]'', &
                & ''  : '')',advance='no')
               call pcheck(2,dwi2,1,dm,imod,ire)
               if (ire==0) exit
              enddo
              do
               write(iy,'(''   "     "   print beyond aphelion/per.[%]'', &
                & ''  : '')',advance='no')
               call pcheck(2,dwi3,1,dm,imod,ire)
               if (ire==0) exit
              enddo
             endif
            endif
            if (ipla==3 .and.ilin>=3) then
             if (ikomb==0) then
              do
               write(iy,'(''  Ang. range of eclipt. longitude   (real)'', &
                 & ''  : '')',advance='no')
               call pcheck(2,dwi,1,dm,imod,ire)
               if (ire==0) exit
              enddo
             else
              do
               write(iy,'(''  Ecl. angular range, VSOP short v. (real)'', &
                 & ''  : '')',advance='no')
               call pcheck(2,dwi,1,dm,imod,ire)
               if (ire==0) exit
              enddo
              do
               write(iy,'(''   "      "      "  , VSOP  full v. (real)'', &
                 & ''  : '')',advance='no')
               call pcheck(2,dwikomb,1,dm,imod,ire)
               if (ire==0) exit
              enddo
             endif
            endif

      ! . . Dreier- oder Viererkonjunktion nur mit Transit
            nurtr = 1
            if (ipla==3 .and.ilin>=3 .and.ison==5 .and.imod/=3 &
                .and.itran==1) then
```



```
              do
               write(iy,'(''  All conjunctions (1),  only transits (2)'', &
                 & ''  : '')',advance='no')
               read(*,*,iostat=iox) nurtr
               if ((nurtr==1 .or.nurtr==2).and.iox==0) exit
               call emes(ire,com,dm)
              enddo
             endif

      ! . . Blickrichtung auf die Planetenbahnen
             iek = 1
             if (ipla/=3) then
              do
               if (ison<=2 .and.(ijd==15 .or.ijd==0)) then
                if ((imod==2 .and.iaph<=2).or.ijd==0) then
                 write(iy,'(''  View from ecliptic  North (1), South (2)'', &
                   & ''  : '')',advance='no')
                 read(*,*,iostat=iox) iek
                 if (iek>=1 .and.iek<=2 .and.iox==0) exit
                else
                 write(iy,'(''  View from eclipt.  N (1), S (2), N/S (3)'', &
                   & ''  : '')',advance='no')
                 read(*,*,iostat=iox) iek
                 if (iek>=1 .and.iek<=3 .and.iox==0) exit
                endif
                call emes(ire,com,dm)
               else
                iek = 1
                if ((ijd>=6 .and.ijd<=11).or.ijd==13 .or.ijd==14) iek=2; exit
               endif
              enddo
             endif

      ! . . Ausgabe
             if (io==0) then
               io = 2; if (iaph==5) io = 1
               if (imo4==0 .and.iaph/=5) then
                do
                 write(iy,'(''  Output         normal (1),  extended (2)'', &
                   & ''  : '')',advance='no')
                 read(*,*,iostat=iox) io
                 if ((io/=2 .or.io==2).and.iox==0) exit
                 call emes(ire,com,dm)
                enddo
               endif
             endif

      ! . . Ausgabegeraet
             do
               if (imod<=2 .and.ipla<=2 .and.ison==5) then
                write(iy,'(''  Mon.(1), file (2), special (3), exit (4)'', &
                  & ''  : '')',advance='no')
                read(*,*,iostat=iox) iout
                if (iout>=1 .and.iout<=4 .and.iox==0) exit
               else
                write(iy,'(''  Monitor (1),  mon. + file (2),  exit (4)'', &
                  & ''  : '')',advance='no')
                read(*,*,iostat=iox) iout
                if ((iout==1 .or.iout==2 .or.iout==4).and.iox==0) exit
```

```
          endif
          call emes(ire,com,dm)
        enddo
        end subroutine

        subroutine inputfile(ipla,ilin,imod,imo4,ikomb,io,lv,ivers, &
         itran,isep,iuniv,ical,ika,iaph,iamax,step,ison,ihi,irb,ijd, &
         zmin,zmax,ak,zjde1,dwi,dwikomb,dwi2,dwi3,nurtr,iek,iop,irw,iout)
!-----Einlesen der Inputdaten bei Schnellstart-------------------------
!       irw=1: lesen aus "inparm.t",  irw=2: schreiben in "inedit.t"
!       Mit Hilfe von inedit.t kann inparm.t manuell editiert werden.
        implicit double precision (a-h,o-z)
        if (irw==1) then
          if (iop/=999) then
            open(unit=10,file='inparm.t')
            do i=1,10*iop+1; read(10,*); enddo
          else
            open(unit=10,file='inedit.t')
            do i=1,24; read(10,*); enddo
          endif
          read(10,*) ipla,ilin,imod,imo4,ikomb
          read(10,*) lv,ivers,itran,isep,iuniv
          read(10,*) ical,ika,iaph,iamax,step
          read(10,*) ison,ihi,irb,ijd
          read(10,*) zmin,zmax,ak,zjde1
          read(10,*) dwi,dwikomb,dwi2,dwi3
          read(10,*) nurtr,iek,io,iout
        elseif (irw==2) then
          open(unit=10,file='inedit.t')
          do i=1,34; read(10,*); enddo
          write(10,'(5i3)') ipla,ilin,imod,imo4,ikomb
          write(10,'(5i3)') lv,ivers,itran,isep,iuniv
          write(10,'(3i3,i6,f10.5)') ical,ika,iaph,iamax,step
          write(10,'(3i3,i4)') ison,ihi,irb,ijd
          write(10,'(3f13.5,f15.5)') zmin,zmax,ak,zjde1
          write(10,'(4f8.3)') dwi,dwikomb,dwi2,dwi3
          write(10,'(4i3)') nurtr,iek,io,iout
          write(10,*) ('-',i=1,59)
          write(10,*) ('*',i=1,27),' END ',('*',i=1,27)
        endif
        close(10)
        end subroutine

        subroutine chambers(ig,rx)
!-----Aenderung der Planeten-Kammer-Zuordnung-------------------------
!       Reihenfolge Koenigs-, Koeniginnen- u. Felsenkammer mit Planeten:
!       ig: 1. E-V-M, 2. E-M-V, 3. V-E-M, 4. V-M-E, 5. M-E-V, 6. M-V-E
        implicit double precision (a-h,o-z)
        dimension :: rx(3,4),x(5),y(5)
        if (ig==3 .or.ig==5) call pchange(1,1,2,rx,x,y,indx)
        if (ig==2 .or.ig==4 .or.ig==5) call pchange(1,2,3,rx,x,y,indx)
        if (ig==4) call pchange(1,1,2,rx,x,y,indx)
        if (ig==6) call pchange(1,1,3,rx,x,y,indx)
        end subroutine

        subroutine pchange(imodus,iz,jz,rxx,x,y,indx)
!-----Vertauschen von Input-Zeilen oder Zahlen in "fitmin"-------------
        implicit double precision (a-h,o-z)
        dimension :: rxx(3,4),x(5),y(5)
```



```
        if (imodus==1) then; do i=1,4
          rpc=rxx(iz,i); rxx(iz,i)=rxx(jz,i); rxx(jz,i)=rpc; enddo
        elseif (imodus==2) then
          z=x(iz); x(iz)=x(jz); x(jz)=z; z=y(iz); y(iz)=y(jz); y(jz)=z
          if (indx==iz) then; indx = jz; return; endif
          if (indx==jz) indx = iz
        endif
        end subroutine

        subroutine pcheck(i,p,n,dm,imod,ire)
!-----Read and check of input parameter p-----------------------------
!       modus i: read + check time (1), tolerance (2)
!       time  n: year (1), k-number (2), JDE (3)
!       p: input parameter, dm: maximum allowed value
!       error code ire (ire = 0 means "no error".)
        implicit double precision (a-h,o-z)
        character(36) :: com
        ire = 0; read(*,*,iostat=iox) p; if (iox/=0) ire = 1
        if (i==1 .and.ire==0) then
          ire = 2
          if (imod/=3) then
            if (n==1 .and.(p<-13000.00001d0 .or.p>17000.00001d0)) then
              com = ' (-13 000. <= year <= 17 000.)      '
            elseif (n==2 .and.(p<-63000.001d0 .or.p>63000.001d0)) then
              com = ' (-63 000. <=  k  <=  63 000.)      '
            elseif (n==3 .and.(p<-3030000.1d0 .or.p>7940000.1d0)) then
              com = ' (-3 030 000. <= JDE <= 7 940 000.) '
            else
              ire = 0
            endif
          else
            if (n==1 .and.(p<-30000.00001d0 .or.p>30000.00001d0)) then
              com = ' (-30 000. <= year <= 30 000.)      '
            elseif (n==2 .and.(p<-133000.01d0 .or.p>117000.01d0)) then
              com = ' (-133 000. <=  k  <=  117 000.)    '
            elseif (n==3 .and.(p<-9240000.1d0 .or.p>12680000.1d0)) then
              com = ' (-9 240 000. <= JDE <= 12 680 000.)'
            else
              ire = 0
            endif
          endif
        elseif (i==2 .and.ire==0) then
          if (p<=0.d0) ire = 1; if (p>dm) ire = 3
        endif
        if (ire/=0) call emes(ire,com,dm)
        end subroutine

        subroutine emes(ire,com,dm)
!-----Error message---------------------------------------------------
        implicit double precision (a-h,o-z)
        character(36) :: com
        iy = 6
        if (ire<=1) write(iy,'(/''  ---->  incorrect input.''/)')
        if (ire==2) write(iy,'(/''  ---->  incorrect input. '', &
          & a36/)')com
        if (ire==3) write(iy,'(/''  ---->  number too large '', &
          & ''(max.'',f6.2,'').''/)') dm
        end subroutine
```

```
      subroutine konst(ik,kon)
!-----Automatische Erkennung der Planetenkonst. 1 bis 14 --> kon-------
!     Suchtoleranz (+/-) fuer Konst.: 53 Tage, fuer "->": 880 Tage
      use base, only : akon
      implicit double precision (a-h,o-z)
      character(2) :: kon,tkon(14)
      data tkon/' 1',' 2',' 3',' 4',' 5',' 6',' 7', &
                ' 8',' 9','10','11','12','13','14'/
      ye = 10.d0; kon = '  '
      ep = 0.6d0; ako = dfloat(ik)
      do i=1,14
        a1 = dabs(ako-akon(i))
        a2 = dabs(ako-(akon(i)-1.d0))
        if (a1<ye.or.a2<ye) kon = '->'
        if (a1<ep.or.a2<ep) kon = tkon(i)
      enddo
      end subroutine

      subroutine ephim(i,iaph,ipla,ical,ak,iak,day,year,delt)
!-----Julian Ephemeris Day and Year (Merkur im Aphel)------------------
!     Input ist "ak" (Nummer des Apheldurchgangs), "day" oder "year".
!     i = 0:  ak   --> day, year, delt
!     i = 1:  day  --> ak, iak, year, delt
!     i = 2:  year --> day, ak, iak
      implicit double precision (a-h,o-z)
      if (i==0) call akday(0,iaph,ipla,ak,iak,day)

! . . Neue Werte (Buch 2)
!     Diese Zahlen verbessern nur die Genauigkeit der dezimalen Jah-
!     reszahl auf +/- 0,5 Tage, aendern jedoch nichts an den bishe-
!     rigen astronomischen Berechnungen und Datumsberechnungen. Alle
!     durch 400 teilbaren Jahreszahlen, wie z.B. -1200.0 oder 2000.0,
!     entsprechen jetzt exakt dem 1. Januar, 12 Uhr. Das heisst, das
!     dezimale Jahr 2000.0 bedeutet die Standard-Epoche J2000.0.
      if (ical==2 .and.((i<=1 .and.day>=0.d0 .and.day<2299160.5d0) &
       .or.(i==2 .and.year>=-4712.d0 .and.year<1582.7854097d0))) then
        A = 365.25d0; B = 0.d0; C =-4712.d0  ! (Julian. Kal.)
      else
        A = 365.2425d0; B = 2451545.d0; C = 2000.d0  ! (Gregor. Kal.)
      endif
! . . Vorherige Werte (Programm P3, Buch 1)
!c      A = 365.248d0; B = 0.d0; C = -4711.9986d0  ! (Programm P3)

! . . Umrechnung der Daten
      if (i<=1) year = (day - B)/A + C
      if (i==1) call akday(1,iaph,ipla,ak,iak,day)
      if (i<=1) then
        call akday(0,iaph,ipla,dnint(ak),iak,aiday)
        delt = day - aiday
      else
        day = A * (year - C) + B
        call akday(1,iaph,ipla,ak,iak,day)
      endif
      end subroutine

      subroutine akday(j,iaph,ipla,ak,iak,day)
!-----Julian Ephemeris Day---------------------------------------------
!     j = 0:  ak  --> day
!     j = 1:  day --> ak,iak
```



```
!     ymer = Umlaufzeit des Merkur in Tagen
      use base, only : pmer,ymer
      implicit double precision (a-h,o-z)
      if (j==0) then
        aak = ak
        if (iaph==1 .or.iaph==3 .or.(iaph==5 .and.ipla==1)) &
          aak = aak - 0.5d0
        day = pmer + ymer * aak
      endif
      if (j==1) then
        ak = (day - pmer)/ymer
        if (iaph==1 .or.iaph==3 .or.(iaph==5 .and.ipla==1)) &
          ak = ak + 0.5d0
        iak = idnint(ak)
      endif
! . . Apheldurchgang der Erde
!c    day = 2451547.507d0 + 365.2596358d0 * (ak + 0.5d0) &
!c          + 1.58d-8 * (ak + 0.5d0)**2
      end subroutine

      subroutine delta_T(zjd)
!-----Umrechnung: Terrestrial Time --> Universal Time-----------------
!     Gleichungen von Fred Espenak und Jean Meeus, entwickelt auf Ba-
!     sis des "Five Millennium Canon of Solar Eclipses", nach Artikeln
!     von Morrison/Stephenson (2004) und Stephenson/Houlden (1986).
!     (NASA Eclipse Web Site, Polynomial expressions for DELTA-T, 2005)
      implicit double precision (a-h,o-z)
      call ephim(1,1,1,1,ak,iak,zjd,y,delt)
      if (y>-500.d0 .and.y<=500.d0) then
         u = y/100.d0
         del = 10583.6d0 - 1014.41d0 * u + 33.78311d0  * u**2 &
               - 5.952053d0 * u**3 - 0.1798452d0 * u**4 &
               + 0.022174192d0 * u**5 + 0.0090316521d0 * u**6
      elseif (y>500.d0 .and.y<=1600.d0) then
         u = (y-1000.d0)/100.d0
         del = 1574.2d0 - 556.01d0 * u + 71.23472d0 * u**2 &
               + 0.319781d0 * u**3 - 0.8503463d0 * u**4 &
               - 0.005050998d0 * u**5 + 0.0083572073d0 * u**6
      elseif (y>1600.d0 .and.y<=1700.d0) then
         t = y - 1600.d0
         del = 120.d0 - 0.9808d0 * t - 0.01532d0 * t**2 &
               + t**3 / 7129.d0
      elseif (y>1700.d0 .and.y<=1800.d0) then
         t = y - 1700.d0
         del = 8.83d0 + 0.1603d0 * t - 0.0059285d0 * t**2 &
               + 0.00013336d0 * t**3 - t**4 / 1174000.d0
      elseif (y>1800.d0 .and.y<=1860.d0) then
         t = y - 1800.d0
         del = 13.72d0 - 0.332447d0 * t + 0.0068612d0 * t**2 &
               + 0.0041116d0 * t**3 - 0.00037436d0 * t**4 &
               + 0.0000121272d0 * t**5 - 0.0000001699d0 * t**6 &
               + 0.000000000875d0 * t**7
      elseif (y>1860.d0 .and.y<=1900.d0) then
         t = y - 1860.d0
         del = 7.62d0 + 0.5737d0 * t - 0.251754d0 * t**2 &
               + 0.01680668d0 * t**3 - 0.0004473624d0 * t**4 &
               + t**5/233174.d0
      elseif (y>1900.d0 .and.y<=1920.d0) then
         t = y - 1900.d0
```

```
           del = -2.79d0 + 1.494119d0 * t - 0.0598939d0 * t**2 &
                 + 0.0061966d0 * t**3 - 0.000197d0 * t**4
        elseif (y>1920.d0 .and.y<=1941.d0) then
           t = y - 1920.d0
           del = 21.20d0 + 0.84493d0 * t - 0.076100d0 * t**2 &
                 + 0.0020936d0 * t**3
        elseif (y>1941.d0 .and.y<=1961.d0) then
           t = y - 1950.d0
           del = 29.07d0 + 0.407d0 * t - t**2/233.d0 + t**3/2547.d0
        elseif (y>1961.d0 .and.y<=1986.d0) then
           t = y - 1975.d0
           del = 45.45d0 + 1.067d0 * t - t**2/260.d0 - t**3/718.d0
        elseif (y>1986.d0 .and.y<=2005.d0) then
           t = y - 2000.d0
           del = 63.86d0 + 0.3345d0 * t - 0.060374d0 * t**2 &
                 + 0.0017275d0 * t**3 + 0.000651814d0 * t**4 &
                 + 0.00002373599d0 * t**5
        elseif (y>2005.d0 .and.y<=2050.d0) then
           t = y - 2000.d0
           del = 62.92d0 + 0.32217d0 * t + 0.005589d0 * t**2
        elseif (y>2050.d0 .and.y<=2150.d0) then
           del = -20.d0 + 32.d0 * ((y-1820.d0)/100.d0)**2 &
                 - 0.5628d0 * (2150.d0 - y)
        else
           u = (y - 1820.d0)/100.d0
           del = -20.d0 + 32.d0 * u**2
        endif
        zjd = zjd - del/86400.d0  ! DELTA-T (del) in Sekunden

  ! . . Alternativ: Jean Meeus, "Transits", S. 73, der wiederum fol-
  !     gende Referenz zitiert: L.V. Morrison, F.R. Stephenson, Sun
  !     and Planetary System, Vol. 96, Reidel, Dordrecht, 1982, S. 73
  !c    zjd = zjd - ((zjd-2382148.d0)**2/41048480.d0 - 15.d0)/86400.d0
        end subroutine

        subroutine jdedate(zjd,ical,ida,da,dmo)
  !-----Berechnung von Datum und Uhrzeit (TT)----------------------------
  !     Programm zur Umrechnung von Julian Day in ein Kalenderdatum.
  !     Es basiert auf einem Algorithmus aus dem Buch von Jean Meeus:
  !     "Astronomical Algorithms", Copyright: 1991, Willmann-Bell,
  !     Inc., P.O.Box 35025, Richmond, Virginia 23235, USA (S. 63).
  !     Anmerkung: Der Algorithmus wurde geringfuegig modifiziert,
  !     so dass er jetzt fuer beide Kalender auch fuer JDE < 0 gilt.
  !     Indizes:
  !     1: dez.Tag, 2: Mon., 3: Jahr, 4: Std, 5: Min, 6: Sek, 7: int.Tag
        implicit double precision (A-H,O-Z)
        dimension :: ida(7),da(7)
        character(5) :: monat(12),dmo
        data monat/' Jan.',' Feb.',' Mar.',' Apr.',' May ',' June', &
                   ' July',' Aug.',' Sep.',' Oct.',' Nov.',' Dec.'/
        Z = sdint(zjd + 0.5d0)
        F = zjd + 0.5d0 - Z
        if (z>=0.d0 .and.Z<2299161.d0 .and.ical==2) then
          A = Z
        else
          alpha = sdint((Z - 1867216.25d0)/36524.25)
          A = Z + 1.d0 + alpha - sdint(alpha*0.25d0)
        endif
        B = A + 1524.d0
```



```
        C = sdint((B - 122.1d0)/365.25d0)
        D = sdint(365.25d0 * C)
        E = sdint((B - D)/30.6001d0)
        da(1) = B - D - sdint(30.6001d0*E) + F + 5.d-9
        if (E<14.d0) then
          da(2) = E - 1.d0
        else
          if (E==14.d0 .or.E==15.d0) then
            da(2) = E - 13.d0
          else
            da(2) = 999.d0
          endif
        endif
        M = idnint(da(2))
        if (M>2) then
           da(3) = C - 4716.d0
        else
          if (M==1 .or.M==2) then
            da(3) = C - 4715.d0
          else
            da(3) = 9999999999999.d0
          endif
        endif
        st  = da(1) - sdint(da(1))
        dst = st*24.D0
        da(4) = sdint(dst)
        da(5) = (dst - sdint(dst))*60.D0
        da(6) = (da(5) - sdint(da(5)))*60.D0
        da(7) = sdint(da(1))                    ! day
        ida(3) = idnint(da(3))                  ! year
        ida(4) = idnint(da(4))                  ! hours
        ida(5) = idnint(da(5)-0.5d0+1.d-10)     ! minutes
        ida(6) = idnint(da(6))                  ! seconds
        imo    = idnint(da(2))                  ! month

  !     Geringfuegige Korrektur der Darstellung
  !     (Beispiel: Uhrzeit 13:44:60 wird zu 13:45:00)
        do i=6,5,-1
          if (ida(i)>=60) then
            ida(i) = ida(i) - 60
            ida(i-1) = ida(i-1) + 1
          endif
        enddo
        if (ida(4)>=24) then
          ida(4) = ida(4) - 24
          da(1) = da(1) + 1.d0
          da(7) = sdint(da(1))
        endif
        if ((dabs(da(7)-32.d0)<=1.d-8.and.(imo==1.or.imo==3 &
           .or.imo==5.or.imo==7.or.imo==8.or.imo==10.or.imo==12)).or. &
            (dabs(da(7)-31.d0)<=1.d-8.and.(imo==4.or.imo==6.or.imo==9 &
           .or.imo==11)).or.(dabs(da(7)-30.d0)<=1.d-8.and.imo==2)) then
          do k=30,32
            q=dfloat(k)
            if (dabs(da(7)-q)<=1.d-8) da(1) = da(1)+1.d0-q
          enddo
          da(7) = sdint(da(1)); imo = imo + 1
          if (imo==13) then
            imo = 1
```

```
            da(3) = da(3) + 1.d0
            ida(3) = idnint(da(3))
          endif
        endif
        dmo = monat(imo)
        end subroutine

        double precision function sdint(x)
!-----Step function-----------------------------------------------------
!       replacing some integer-functions in the subroutine "jdedate"
!       in order to expand the domain of definition for JDE < 0
        real(8) :: x
        sdint = dint(x)
        if (x<0.d0 .and.dmod(x,1.d0)/=0.d0) sdint = sdint - 1
        end function

        subroutine weekday(ZJD,wd)
!-----Berechnung des Wochentages--------------------------------------
        implicit double precision(a-h,o-z)
        character(10) :: wday(0:6),wd
        data wday/'    Sunday','    Monday','   Tuesday',' Wednesday', &
                  '  Thursday','    Friday','  Saturday'/
        wd = wday(idnint(dmod(dint(ZJD + 700000001.5d0),7.d0)))
        end subroutine

        subroutine vsop1(l,tau,resu)
!-----Berechnung der ekliptikalen Koordinaten (VSOP87D-Kurzversion)----
        use base, only : gdpi,z0,lmax,jp; use astro, only : par1
        implicit double precision (a-h,o-z)
        resu = z0
        do j=1,lmax(l)
          sum0 = z0
          do i=1,jp(l,j)
            sum0 = sum0 + par1(1,i,j,l) * &
                     dcos(par1(2,i,j,l) + par1(3,i,j,l)*tau)
          enddo
          resu = resu + sum0*tau**(j-1)
        enddo
        resu = resu * 1.d-8
        if (l==1 .or.l==4 .or.l==7 .or.l==10) call reduz(resu,1,1)
        if (l/=3 .and.l/=6 .and.l/=9 .and.l/=12) resu = resu*gdpi
        end subroutine

        subroutine vsop2(zjde,ivers,ibody,md,ix,prec,lu,r,ierr,rku)
!-----Aufruf der VSOP-Subroutine (VSOP87A/C-Vollversionen)-------------
!       (Index von rku  1: L,  2: B,  3: r)
        implicit double precision (a-h,o-z)
        dimension :: r(6),rku(3),md(0:9)
        character(11) :: afile(9),cfile(8)
        data afile/    'VSOP87A.mer','VSOP87A.ven','VSOP87A.ear', &
         'VSOP87A.mar','VSOP87A.jup','VSOP87A.sat','VSOP87A.ura', &
         'VSOP87A.nep','VSOP87A.emb'/
        data cfile/    'VSOP87C.mer','VSOP87C.ven','VSOP87C.ear', &
         'VSOP87C.mar','VSOP87C.jup','VSOP87C.sat','VSOP87C.ura', &
         'VSOP87C.nep'/
        if (md(ibody)==1) then
          if (ivers==1) open(unit=10,file=afile(ibody))
          if (ivers==3) open(unit=10,file=cfile(ibody))
        endif
```



```
        call VSOP87X(zjde,ivers,ibody,prec,lu,r,ierr,md)
        if (md(ibody)==1) close(10)
        call kugelko(r(1),r(2),r(3),rku)
!c      write(6,'(/''  x, y, z  = '',3f14.10)') (r(i),i=1,3)
!c      write(6,'( ''  vx,vy,vz = '',3f14.10)') (r(i),i=4,6)
!c      write(6,'( ''  L, B, r  = '',3f14.10)') (rku(i),i=1,3)
        do iu=ix,6,5
          if (ierr/=0) write(iu,'(''  In VSOP87X: ierr = '',i2)')ierr
        enddo
        end subroutine

        subroutine vsop3(l,k,ix,ke,time,res)
!-----Bahn-Elemente, abgeleitet aus VSOP82 (nach Meeus)----------------
!       fuer J2000.0 und Ekliptik der Epoche; Berechnung der wahren
!       Anomalie (ekliptikale Laenge) mit der Keplerschen Gleichung.
!       (Index von res  1: L, 2: a, 3: e, 4: i, 5: Omega, 6: pi, 7: M,
!                       8: omega, 9: E, 10: nue, 11: eklipt. Laenge)
        use base, only : pidg,gdpi
        use astro, only : par3
        implicit double precision (a-h,o-z)
        dimension :: res(12)
        u360 = 360.d0; ke = 0
        eps = 1.d-13
        do j=1,6
          resu = 0.d0
          do i=1,4
            resu = resu + par3(i,j,k,l)*time**(i-1)
            if (j==1 .or.j>=5) call reduz(resu,0,1)
            res(j) = resu
          enddo
        enddo
        res(7) = res(1) - res(6)
        if (res(7)<0.d0) res(7) = res(7) + u360
        res(8) = res(6) - res(5)
        if (res(8)<0.d0) res(8) = res(8) + u360

! . . Loesung der Keplerschen Gleichung (Resultat: zen)
        ii = 0
        E  = res(3)
        zm = res(7)*pidg
        ze = zm
        itmax = 100  ! Maximalzahl der Iterationen

        meth = 1  ! Drei iterative Methoden zur Auswahl (meth = 1..3)
        if (meth<3) then
          do
           if (meth==1) then
!       1. Verfahren von Newton-Raphson (schnellste Methode)
             zen = ze + (zm + E*dsin(ze) - ze)/(1.d0 - E*dcos(ze))
           else
!       2. Fixpunktverfahren (Keplersche Gleichung)
             zen = zm + E*dsin(ze)
           endif
           if (dabs(zen-ze)<eps) exit
           if (ii>itmax) then; ke = 2; go to 20; endif
           ii = ii+1
           ze = zen
          enddo
        else
```

```
!     3. Sekantenverfahren (verwendet Sekantensteigung)
         ke = 1
         ze2 = zm
   10    fze2 = zm + E*dsin(ze2) - ze2
         call sekante(ze1,ze2,fze1,fze2,eps,0.1d0,ii,itmax,ix,ke)
         if (ke==1) go to 10
         if (ke==2) go to 20 ! ("Ringfit" hat hier keinen Zeitvorteil
         zen = ze2           ! gegenueber "sekante", da die Keplersche
      endif                  ! Gleichung deutlich weniger Rechenzeit
      go to 30               ! benoetigt als "Ringfit" selbst.)

! . . zu viele Iterationen
   20 do iu=ix,6,5
        write(iu,'(/''  ---->  error in "vsop3" '', &
          & ''(Keplers equation), ke ='',I2/)') ke
      enddo
      return
   30 res(9) = zen*gdpi
      if (res(9)<0.d0) res(9) = res(9) + u360

! . . Berechnung der wahren Anomalie
      res(10) = 2.d0 * datan(dsqrt((1.d0 + E)/(1.d0 - E)) &
                  * dtan(zen*0.5d0))*gdpi
      if (res(10)<0.d0) res(10) = res(10) + u360
      res(11) = res(10) + res(6)
      if (res(11)>u360) res(11) = res(11) - u360
      end subroutine

      subroutine transit(ip,ikomb,imod,ipla,ilin,iap,ivers,isep, &
        ical,iuniv,tr,sepmin,itt,sep,zjde,id5,da5,dmo5,zjahr, &
        rk,md,ddx1,ddx2,dfd,test,itin,is,ires,ix,pan,sd,sl,iop0,inum)
!-----Ueberpruefung der Transite von Merkur bzw. Venus-----------------
!     Die berechneten Zeitpunkte sind optional dieselbe ekliptikale
!     Laenge bei Erde und Merkur bzw. Venus, die minimale Separation
!     oder die genauen Phasen. "M" bedeutet "normaler", "C" (geozen-
!     trischer) zentr. Transit des Merkurs und "m"/"c", dass irgend-
!     wo auf der Erde der Transit partiell/zentral erscheint. Analog
!     stehen "V" und "v" fuer die Venus. Das Minuszeichen "-" bedeu-
!     tet, dass der Planet die Sonne knapp verfehlt und dass der
!     dichteste Abstand der "sichtbaren" Scheiben (Sonnen- und Plane-
!     tenrand) nicht mehr als etwa 1 Prozent des scheinbaren Sonnen-
!     radius' betraegt (verwendet nur bei Syzygy-Berechnungen). Die
!     Planetenscheibe ist in diesem Fall natuerlich nicht sichtbar.
!     Index (ip): 1 = Merkur, 2 = Venus
      use base
      implicit double precision (a-h,o-z)
      dimension :: zi(2),sd(2),tcorr(2),rem(78)
      dimension :: ida(7),da(7),id5(5,7),da5(5,7),pan(5)
      dimension :: r(6),rku(3),rk(12),md(0:9),inum(0:4)
      dimension :: xx(5),yy(5),xk(2),yk(2),test(10)
      character(5) :: dmo,dmo5(5)
      character(1) :: tr,tp(8),sl
      data tp/'M','m','V','v','-',' ','C','c'/
      data idr/0/,blim/0.d0/,ba/0.d0/,ang/0.d0/,shift/0.d0/! pre-init.
! . . Einige Konstanten
      T = (zjde-zjd0)/tcen
!     Axel D. Wittmann: we = Schiefe der Ekliptik der Epoche
      we = (23.4458042d0 - 0.856033d0 * &
            dsin(0.015306d0 * (T + 0.50747d0))) * pidg
```



```
      zi(1) = re(35); zi(2) = re(41)
      wfact = 3600.d0*gdpi; eps = 1.d-7
!     (Der folgende Korrekturfaktor "tcorr" zur Berechnung
!     der minimalen Separation ist nur eine Abschaetzung.)
      do j=1,2; tcorr(j) = tsyn(j)/tsid(j); enddo
      ee = dsqrt(R3a*R3a-R3p*R3p)/R3a
      R3 = R3p/(AE*dsqrt(1.d0-(ee*dsin(we))**2))
      a  = dasin(R0/(AE*re(9)))
      b3 = dasin(R3*re(3*ip)/(re(9)*(re(9)-re(3*ip))))
      bp = dasin(Ra(ip)/(AE*(re(9)-re(3*ip))))
      bmin1 = a-bp; bmin2 = a-bp-b3
      bmax1 = a+bp; bmax2 = a+bp+b3

!.....OPTIONEN 1/ 2: gleiche eklipt. Laenge u. minimale Separation
      if (isep==1) then
        din = dcos(zi(ip)*pidg*tcorr(ip))
        dre = (re(3*ip-1)-re(8))*pidg
        ba  = din*datan(re(3*ip)*dsin(dre)/(re(9)-re(3*ip)*dcos(dre)))
        bap = dabs(ba)
      else
        bap = sepmin
      endif
      if (ikomb==1 .and.imod==1) bmax2 = bmax2*1.8d0
      bout = bmax2*1.01d0; tr = tp(6)
      if (bap<=bmin2) tr = tp(2*ip-1)
      if (bap>bmin2.and.bap<=bmax2) tr = tp(2*ip)
      if (bap>bmax2.and.bap<=bout.and.ilin>=3) tr = tp(5)
      if (isep<=2 .and.ilin<=2) then
       if (bap<=bp+b3) tr = tp(8)
       if (bap<=bp) tr = tp(7)
      endif
!c    do iu=ix,6,5; write(iu,'(a15,a18,i3,5f8.5)')'ip,bmin2,bmin1,', &
!c      'bmax1,bmax2,bap = ',ip,bmin2,bmin1,bmax1,bmax2,bap; enddo

! . . Min. Separation (sep) zw. Sonne und Planet in
!     Bogensekunden. Bei "plus" passiert der Planet das
!     das Sonnenzentrum noerdlich, bei "minus" suedlich.
      if (isep==1) then
        sep = ba*wfact
      else
        sep = bap*wfact
        if (re(3*ip-1)<0.d0) sep = -sep
      endif
      if (isep<=2) then
        if (tr==' '.or.ilin>=3) return; go to 60
      endif

!.....OPTIONEN 3/ 4: Transitphasen ohne/mit Positionswinkeln
!     (Beginn, Ende und minimale Separation des geozentrischen Tran-
!     sits => Ein, drei oder fuenf Zeitpunkte werden berechnet.)
      if (bap>bmax2*1.005d0 .or.(ikomb==1 .and.imod==1)) then
        itt = 0
        return
      endif

! . . Weitere Parameter festlegen
      prec = z0; lu = 10
      itr = 1
      do j=1,78; rem(j) = re(j); enddo
```

```
        do j=1,5
          do k=1,7
            id5(j,k) = 0
            da5(j,k) = z0
          enddo
        enddo
        xj2 = zjde

  ! . . Mitte des Transits, minimale Separation mit Lichtlaufzeit
        if (itr==1) then
          idr = 3; ke = 1; indx = 1
          step = 5.d-2; iflag = 0
          ddx1 = dfd + 1.d0; nu = 0
          if (ilin<=2) ddx1 = 1; ddx2 = ddx1
          xx(1) = xj2; itin = 0; iex = 0
          do j=1,10; test(j) = z0; enddo
  !       Mittlere Laufzeit des Lichtes, optimierter Startwert [Tage]
          if (ip==1) del = 320.d0/86400.d0  ! Merkur
          if (ip==2) del = 150.d0/86400.d0  ! Venus
          if (imod==1) then; ept=3.d-14; else; ept=2.d-9; endif

  !       VSOP87-Berechnung mit Beruecksichtigung der Lichtlaufzeit
     10   if (imod==1) then
           call vsop1tr(ip,rk,(xj2-zjd0-del)/tmil,del,r3i,ept,inum,resu)
          else
           call vsop2tr(xj2-del,ivers,ip,md,ix,prec,lu,r,rk, &
             ierr,del,r3i,ept,inum,rku)
          endif
          if (iex==1) go to 20
  !       Bestimmung: auf- bzw. absteigender Knoten
          if (nu==1 .or.nu==2) then
            xk(nu) = xj2; yk(nu) = re(3*ip-1)
          endif
          if (nu==2) then
            sl = '/'; if ((yk(2)-yk(1))/(xk(2)-xk(1))<0.d0) sl = ' '
          endif
  !       Ende Knotenbestimmung
          call sepa(ip,2,rk,sep0i)
          yy(indx) = sep0i
          epv = 1.d-6; if (sep0i<30.d0) epv = 1.d-7
          call fitmin(imod,2,iap,ke,xx,yy,epv,step,nu,iflag, &
                ddx1,ddx2,test,itin,indx,ix)
          xj2 = xx(indx)
          if (ke==0 .and.isep==4 .and.iex==0) then
          iex = 1; go to 10; endif
          if (ke==1) go to 10

  !       Art des (streifenden) Transits
     20   if (sep0i<=bmin2) then; tr=tp(2*ip-1); itt=3; endif
          if (sep0i>bmin2.and.sep0i<=bmin1) itt=3
          if (sep0i>bmin1.and.sep0i<=bmax1) itt=2
          if (sep0i>bmax1.and.sep0i<=bmax2) itt=1
          if (sep0i>bmax2) then; itt = 0; return; endif
          if (sep0i>bmin2.and.sep0i<=bmax2) then
            inum(3) = inum(3) + 1
            tr=tp(2*ip)
          endif
          sep = sep0i*wfact
          if (re(3*ip-1)<0.d0) sep = -sep
```



```
            xjdt = xj2; zjde = xj2
            if (iuniv==2) call delta_T(xjdt)
            call jdedate(xjdt,ical,ida,da,dmo)
            call ephim(1,iaph,ipla,ical,ak,iak,zjde,zjahr,delt)

  !         Berechnung des Positionswinkels (minimale Separation)
            if (isep==4) call pos_angle(ip,zjde,rk,ang)

  !         Radien (semidiameter) von Sonne und Merkur/Venus
            if (isep>=3 .and.ilin<=2) then
              sd(1) = dasin(R0/(AE*re(9))) * wfact
              sd(2) = dasin(Ra(ip)/(AE*r3i)) * wfact
  !           Kennzeichnung des zentralen Transits
              csep = (r3*re(3*ip)/re(9)+Ra(ip)/AE)*wfact/(re(9)-re(3*ip))
              if (dabs(sep)<csep) then
                tr = tp(8)
                if (dabs(sep)<sd(2)) tr = tp(7)
                inum(4) = inum(4) + 1
              endif
  !           Mit der zeitlichen Verschiebung "shift" (in julian. Tagen)
  !           wird der spaeter folgende Startpunkt fuer "ringfit" bzw.
  !           "sekante" moeglichst nahe an die Nullstelle verlegt.
              wu = 1.d0-(sep/sd(1))**2
              if (wu<1.d-2) wu = 1.d-2
              if (ip==1) shift = 0.115d0 * dsqrt(wu)
              if (ip==2) shift = 0.17d0 * dsqrt(wu)
            endif
          endif

          if (itr==1) then
            if (itt==1) itr = 6
            go to 50
          endif

  ! . . Vorbereitung zur naechsten Berechnung im selben Transit
     30   iis = 0; ke = 1
          itr = itr + 1
  !       Kontaktpunkt I
          if (itr==2) then
            idr = 1; blim = bmax1
            xj2 = zjde - shift
          endif
  !       Kontaktpunkt II
          if (itr==3) then
            if (itt==2) itr = 5
            idr = 2; blim = bmin1
            xj2 = zjde - shift
          endif
  !       Kontaktpunkt III
          if (itr==4) then
            idr = 4; blim = bmin1
            xj2 = zjde + shift
          endif
  !       Kontaktpunkt IV
          if (itr==5) then
            idr = 5; blim = bmax1
            xj2 = zjde + shift
          endif
```

```
! . . Berechnung der Kontaktzeiten I bis IV
      if (imod==1) then; ept=1.d-12; else; ept=2.d-7; endif
   40 tau = (xj2 - zjd0)/tmil

!     VSOP87D Kurzversion (imod=1), VSOP87C Vollversion (imod=2)
      if (imod==1) then
        call vsop1tr(ip,rk,tau,del,r3i,ept,inum,resu)
      else
        call vsop2tr(xj2,ivers,ip,md,ix,prec, &
          lu,r,rk,ierr,del,r3i,ept,inum,rku)
      endif
!     "Sekante" wurde durch das etwas schnellere "ringfit" ersetzt.
      call sepa(ip,2,rk,sep0i)
      yy2 = sep0i-blim
      call ringfit(xj1,xj2,xj3,yy1,yy2,yy3,eps,1.d-3,iis,25,ix,ke)
      if (ke==1 .or.ke==5) go to 40
      if (ke==2) go to 60
      xjdt = xj2 + del
      if (iuniv==2) call delta_T(xjdt)
      call jdedate(xjdt,ical,ida,da,dmo)

! . . Berechnung des Positionswinkels (Planet am Sonnenrand)
      if (isep==4 .and.itr/=1) call pos_angle(ip,xj2,rk,ang)

! . . Ruecksprung
   50 do k=1,7; id5(idr,k) = ida(k); da5(idr,k) = da(k); enddo
      dmo5(idr) = dmo
      pan(idr) = ang
      if (itr<=4) go to 30
      do j=1,78; re(j) = rem(j); enddo

!.....Berechnung der Transitserie
   60 if (ikomb==0 .or.(ikomb==1 .and.imod==2)) &
        call tserie(ip,zjde,is,iop0,ires)
      end subroutine

      subroutine sepa(ip,iv,rk,sep0i)
!-----Berechnung der Separation Sonne-Merkur bzw. Sonne-Venus----------
!     Index ip:  1 = Merkur, 2 = Venus
      use base, only : pidg,re
      implicit double precision (a-h,o-z)
      dimension :: rk(12),rd(3)
      if (iv==1) then
! . . . 1. Variante - raeumliche Geometrie (Testvariante)
        cos0i = dsin(re(3*ip-1)*pidg) * dsin(re(8)*pidg) + &
                dcos(re(3*ip-1)*pidg) * dcos(re(8)*pidg) * &
                dcos((re(3*ip-2)-re(7))*pidg)
        sep0i = datan(re(3*ip)*dsqrt(1.d0-cos0i*cos0i)/ &
                       (re(9)-re(3*ip)*cos0i))
      else
! . . . 2. Variante - Vektoranalysis
        do j=1,3; rd(j) = rk(3*(ip-1)+j) - rk(6+j); enddo
        ab = -rk(7)*rd(1)-rk(8)*rd(2)-rk(9)*rd(3)
        a  = dsqrt(rk(7)**2 + rk(8)**2 + rk(9)**2)
        b  = dsqrt(rd(1)**2 + rd(2)**2 + rd(3)**2)
        sep0i = dacos(ab/(a*b))
      endif
      end subroutine
```



```
      subroutine pos_angle(ip,xjd,rk,ang)
!-----Positionswinkel des Planeten fuer beliebigen Zeitpunkt des Tran-
!     sits in Bezug auf die Richtung zum Himmelsnordpol (y-Achse auf
!     Sonnenscheibe) – vgl. scheinbare Bewegungsrichtung der Sonne.
!     ip        : 1 fuer Merkur, 2 fuer Venus
!     xjd       : Zeitpunkt der Ankunft des Lichtes auf der Erde
!     rk(1..9)  : rechtwinklige heliozentrische Koordinaten
!                 von Merkur, Venus und Erde (VSOP87C)
!     eeps      : Stellung Erdachse gegen Ekliptik in jener Epoche
!     rgeo(1..9): transformierte geozentrische Koordinaten von Sonne,
!                 Merkur und Venus (rechtwinklig, dann sphaerisch)
!     ang       : Positionswinkel des Planeten vor der Sonne
      use base, only : pidg,gdpi,zjd0,tcen
      implicit double precision (a-h,o-z)
      dimension :: rk(12),rgeo(9),rku(3),xx(3)
      do i=1,9; rgeo(i) = rk(i); enddo

!.....Die Berechnung des Positionswinkels erfolgt in 4 Schritten.
!     Schritte 1-3: Koordinatentransformation helio- zu geozentrisch.

!  1. Rotation um x-Achse um Winkel der Schiefe der Ekliptik (Epoche);
!     Axel D. Wittmann: "On the variation of the obliquity of the
!     ecliptic", Univ.-Sternwarte Goettingen, 1984, MitAG 62, S.203
      T = (xjd-zjd0)/tcen
      eeps = (23.4458042d0 - 0.856033d0 * &
              dsin(0.015306d0 * (T + 0.50747d0))) * pidg
      call rotmat(1,-eeps,0.d0,0.d0,rgeo)

!  2. Translation des heliozentrischen Koordinatenursprungs von der
!     Sonne zur Erde. Das ergibt neue Koordinaten fuer Sonne und
!     Merkur bzw. Venus.
      do i=1,3
        xx(i) = -rgeo(6+i); rgeo(6+i) = rgeo(3+i)
        rgeo(3+i) = rgeo(i); rgeo(i) = 0.d0
      enddo
      call translat(xx(1),xx(2),xx(3),rgeo)

!  3. Umrechnung in sphaerische Koordinaten
!     (Positionen von Sonne, Merkur und Venus)
      do i=0,2; ii = 3*i
        call kugelko(rgeo(ii+1),rgeo(ii+2),rgeo(ii+3),rku)
        do j=1,3; rgeo(ii+j) = rku(j); enddo
      enddo

!  4. Berechnung des Positionswinkels nach Andre Danjon: "Astronomie
!     Generale", S.36, Gl."3 bis". Siehe auch Jean Meeus: "Transits",
!     S.15 ("kartesische" Koordinaten x und y in Bogensekunden).
      sdec = rgeo(2) * pidg
      dra  = (rgeo(3*ip+1)-rgeo(1)) * pidg
      ddec = (rgeo(3*ip+2)-rgeo(2)) * pidg
      tdra = dsin(sdec) * dtan(dra) * dtan(dra*0.5d0)
      zk = 206264.8062d0/(1.d0 + dsin(sdec) * tdra)
      x = -zk * (1.d0 - dtan(sdec)*dsin(ddec)) * dcos(sdec)*dtan(dra)
      y =  zk * (dsin(ddec) + dcos(sdec) * tdra)
      ang = datan(-x/y)*gdpi
      if (y*dcos(ang*pidg)<0.d0) ang = ang + 180.d0
      call reduz(ang,0,1)
      end subroutine
```

```
      subroutine tserie(ip,zjde,is,iop0,ires)
!-----Bestimmung der Transit-Serie------------------------------------
!     Die Seriennummern entsprechen denen der "NASA Eclipse Web Site".
!     (Die Liste der Seriennummern "inserie.t" wird nur einmal verwen-
!     det, um die Startnummern, d.h. die Nummern zu bestimmen, die den
!     ersten gefundenen Transiten zugeordnet werden. Danach werden al-
!     le weiteren Seriennummern unabhaengig von der Liste berechnet.)
!     Index (ip):  1 = Merkur, 2 = Venus
      use astro, only : ser,ase,cc,t13BC,t17AD, &
        zstart,ise,ji,jj,isflag,ismax
      implicit double precision (a-h,o-z)
      if (dabs(zstart-99.99d0)<1.d-10) zstart = zjde
      if (iop0/=-803) then
        if (zjde<t13BC-365.d0 .or.zjde>t17AD+365.d0) then
          ires = 999; return
        endif

! . . . Seriennummer (is) fuer Startzeitpunkt suchen
        if (isflag==0) then
          do j=jj(2*ip-1),jj(2*ip)
            if (ser(j,ip)>zjde) then
              is = j
              isflag = 1
              exit
            endif
          enddo
        endif
      endif

! . . Aktuelle Seriennummer bestimmen
      kflag = 0
      do j=is-ji(ip),is
        zlim = dmax1(t13BC,zstart)
        if (zjde-zlim>cc(ip)+100.d0) then
          do k=jj(2*ip-1),is
            ise(k) = 1
          enddo
        endif
        a = (zjde-ser(j,ip))/cc(ip)
        x = dabs((a-dnint(a))*cc(ip))
        b = dabs(zjde-ase(j)-cc(ip))
!c      write(6,'(''a,x,b,ise(j),j,is,ismax ='',f9.3,f10.3,f16.6, &
!c       & i3,3i5)')a,x,b,ise(j),j,is,ismax
        if (x<=10.d0 .and.(b<=2.d0 .or.ise(j)==0)) then
          ires = j
          kflag = 1
          if (j>ismax) ismax = j
        endif
        if (j==is.and.kflag==1) go to 20
      enddo
      if (ismax==-10000 .or.is>ismax) ismax = is - 1
      is = ismax + 1
      ismax = is
      ser(is,ip) = zjde
      ires = is
   20 ase(ires) = zjde
      ise(ires) = 1
      end subroutine
```



```
      subroutine VSOP87X(tdj,ivers,ibody,prec,lu,r,ierr,md)
!-------------------------------------------------------------------
! >>
! >>  UPGRADE (by H. Jelitto): As proposed by Bretagnon and Francou
! >>  for rapidity of computation, the parameters in the VSOP87-files
! >>  are read only once at the first call for each planet. The main
! >>  data are copied into the 5-dimensional array "par2" for random
! >>  access, covering all planets of one VSOP87-version. For the
! >>  calculation of the transit phases (TYMT-test), this reduces the
! >>  computing time by a factor 20 to 30. Thus, the original subrou-
! >>  tine "VSOP87" is extended and renamed as "VSOP87X".
! >>
! >>  The new VSOP87X routine has been checked only for the use of
! >>  the theory versions VSOP87A and VSOP87C. Furthermore, the code
! >>  is converted to the Fortran 95 standard and the free source
! >>  form.
! >>
! >>  The following text belongs to the original VSOP87-subroutine.
! >>
!-------------------------------------------------------------------
!
!     Reference : Bureau des Longitudes - PBGF9502
!
!     Object :
!
!     Substitution of time in VSOP87 solution written on a file.
!     The file corresponds to a version of VSOP87 theory and to a body.
!
!     Input :
!
!     tdj      julian date (real double precision).
!              time scale : dynamical time TDB.
!
!     ivers    version index (integer).
!              0: VSOP87 (initial solution).
!                 elliptic coordinates
!                 dynamical equinox and ecliptic J2000.
!              1: VSOP87A.
!                 rectangular coordinates
!                 heliocentric positions and velocities
!                 dynamical equinox and ecliptic J2000.
!              2: VSOP87B.
!                 spherical coordinates
!                 heliocentric positions and velocities
!                 dynamical equinox and ecliptic J2000.
!              3: VSOP87C.
!                 rectangular coordinates
!                 heliocentric positions and velocities
!                 dynamical equinox and ecliptic of the date.
!              4: VSOP87D.
!                 spherical coordinates
!                 heliocentric positions and velocities
!                 dynamical equinox and ecliptic of the date.
!              5: VSOP87E.
!                 rectangular coordinates
!                 barycentric positions and velocities
!                 dynamical equinox and ecliptic J2000.
!
!     ibody    body index (integer).
```

```
!              0: Sun (not used here in VSOP87X)
!              1: Mercury
!              2: Venus
!              3: Earth
!              4: Mars
!              5: Jupiter
!              6: Saturn
!              7: Uranus
!              8: Neptune
!              9: Earth-Moon barycenter
!
!     prec     relative precision (real double precision).
!
!              if prec is = 0 then the precision is the precision
!                 p0 of the complete solution VSOP87.
!                 Mercury    p0 =  0.6 10**-8
!                 Venus      p0 =  2.5 10**-8
!                 Earth      p0 =  2.5 10**-8
!                 Mars       p0 = 10.0 10**-8
!                 Jupiter    p0 = 35.0 10**-8
!                 Saturn     p0 = 70.0 10**-8
!                 Uranus     p0 =  8.0 10**-8
!                 Neptune    p0 = 42.0 10**-8
!
!              if prec is not equal to 0, let us say in between p0 and
!              10**-2, the precision is :
!                 for the positions :
!                 - prec*a0 au for the distances.
!                 - prec rd for the other variables.
!                 for the velocities :
!                 - prec*a0 au/day for the distances.
!                 - prec rd/day for the other variables.
!                   a0 is semi-major axis of the body.
!                 Mercury    a0 =  0.3871 au
!                 Venus      a0 =  0.7233 au
!                 Earth      a0 =  1.0000 au
!                 Mars       a0 =  1.5237 au
!                 Jupiter    a0 =  5.2026 au
!                 Saturn     a0 =  9.5547 au
!                 Uranus     a0 = 19.2181 au
!                 Neptune    a0 = 30.1096 au
!
!     lu       logical unit index of the file (integer).
!              The file corresponds to a version of VSOP87 theory and
!              a body, and it must be defined and opened before the
!              first call to subroutine VSOP87.
!
!     Output :
!
!     r(6)     array of the results (real double precision).
!
!              for elliptic coordinates :
!                  1: semi-major axis (au)
!                  2: mean longitude (rd)
!                  3: k = e*cos(pi) (rd)
!                  4: h = e*sin(pi) (rd)
!                  5: q = sin(i/2)*cos(omega) (rd)
!                  6: p = sin(i/2)*sin(omega) (rd)
!                     e:     eccentricity
```



```
!                     pi:    perihelion longitude
!                     i:     inclination
!                     omega: ascending node longitude
!
!              for rectangular coordinates :
!                  1: position x (au)
!                  2: position y (au)
!                  3: position z (au)
!                  4: velocity x (au/day)
!                  5: velocity y (au/day)
!                  6: velocity z (au/day)
!
!              for spherical coordinates :
!                  1: longitude (rd)
!                  2: latitude (rd)
!                  3: radius (au)
!                  4: longitude velocity (rd/day)
!                  5: latitude velocity (rd/day)
!                  6: radius velocity (au/day)
!
!     ierr     error index (integer).
!                  0: no error.
!                  1: file error (check up ivers index).
!                  2: file error (check up ibody index).
!                  3: precision error (check up prec parameter).
!                  4: reading file error.
!
!----------------------------------------------------------------------

!     --------------------------------
!     Declarations and initializations
!     --------------------------------
      use astro, only : par2,it2,in2,iv2
      implicit double precision (a-h,o-z)
      character(7) :: bo,body(0:9)
      dimension :: r(6),t(-1:5),a0(0:9),md(0:9)
      data body/'SUN','MERCURY','VENUS','EARTH','MARS','JUPITER', &
                'SATURN','URANUS','NEPTUNE','EMB'/
      data a0/0.01d0,0.3871d0,0.7233d0,1.d0,1.5237d0,5.2026d0, &
              9.5547d0,19.2181d0,30.1096d0,1.d0/
      data dpi/6.283185307179586476d0/
      data t/0.d0,1.d0,5*0.d0/
      data t2000/2451545.d0/
      data a1000/365250.d0/
      k=0; ierr=3
      if (md(ibody)==1) then
        ideb=0
        do i=1,3; do j=0,5; it2(j,i,ibody) = -1; enddo; enddo
      endif
      do i=1,6; r(i)=0.d0; enddo
      t(1)=(tdj-t2000)/a1000
      do i=2,5; t(i)=t(1)*t(i-1); enddo
      if (prec<0.d0 .or.prec>1.d-2) return
      if (md(ibody)/=1) ierr = 0
      q=dmax1(3.d0,-dlog10(prec+1.d-50))

!     ----------------------------------------------------------------
!     File reading, for each planet only at first call to VSOP87X
!     ----------------------------------------------------------------
```

```
      if (md(ibody)==1) then
   10   read (lu,1001,end=20) iv,bo,ic,it,inn
        iv2(ibody) = iv
        it2(it,ic,ibody) = 1
        in2(it,ic,ibody) = inn
        if (ideb==0) then
          ideb=1
          ierr=1
          if (iv/=ivers) return
          ierr=2
          if (bo/=body(ibody)) return
          ierr=0
        endif
        if (inn==0) go to 10
        do n=1,inn
          read (lu,1002) (par2(i,n,it,ic,ibody),i=1,3)
        enddo
        go to 10
   20   md(ibody) = 2
      endif

!     ------------------------------------
!     Computation of planetary coordinates
!     ------------------------------------
      ic = 1; it = 0
      iv = iv2(ibody)
      if (iv==0) k=2
      if (iv==2 .or.iv==4) k=1
   30 inn = in2(it,ic,ibody)
      if (inn==0) go to 50
      p=prec/10.d0/(q-2)/(dabs(t(it))+it*dabs(t(it-1))*1.d-4+1.d-50)
      if (k==0 .or.(k/=0 .and.ic==5-2*k)) p=p*a0(ibody)
      do 40 n=1,inn
        a = par2(1,n,it,ic,ibody)
        b = par2(2,n,it,ic,ibody)
        c = par2(3,n,it,ic,ibody)
        if (dabs(a)<p) go to 50
        u = b + c*t(1)
        cu = dcos(b + c*t(1))
        r(ic) = r(ic) + a*cu*t(it)
!c      if (iv==0) go to 40
!c      su=dsin(u)    ! velocity of planet (not used)
!c      r(ic+3)=r(ic+3)+t(it-1)*it*a*cu-t(it)*a*c*su
   40 enddo
   50 if (it<=4) itnext = it2(it+1,ic,ibody)
      if (it<=4 .and.itnext/=-1) then
        it = it + 1; go to 30
      else
        if (ic<3) then
          it = 0
          ic = ic + 1; go to 30
        endif
      endif
      if (iv/=0) then
        do i=4,6; r(i)=r(i)/a1000; enddo
      endif
      if (k==0) return
      r(k)=dmod(r(k),dpi)
      if (r(k)<0.d0) r(k)=r(k)+dpi
```



```
      return

!     -------
!     Formats
!     -------
 1001 format (17x,i1,4x,a7,12x,i1,17x,i1,i7)
 1002 format (79x,f18.11,f14.11,f20.11)
      end subroutine

      subroutine kartko(ison)
!-----Umwandlung in kartesische Koordinaten, re(1..9) --> xyr(1..9)----
!     mit Merkur bei x-Achse
!     Indizes von "re" :  1: Lm'  2: Bm   3: rm   4: Lv'  5: Bv
!                         6: rv   7: Le'  8: Be   9: re
!     Indizes von "xyr":  1: xm   2: ym   3: zm   4: xv   5: yv
!                         6: zv   7: xe   8: ye   9: ze  10: leer
      use base
      implicit double precision (a-h,o-z)
      rr = re(1)
      if (ison==2) rr = re(4)
      if (ison==0) rr = 0.d0
      do i=3,9,3
       xyr(i-2) = re(i)*dcos(re(i-1)*pidg)*dcos((re(i-2)-rr)*pidg)
       xyr(i-1) = re(i)*dcos(re(i-1)*pidg)*dsin((re(i-2)-rr)*pidg)
       xyr(i)   = re(i)*dsin(re(i-1)*pidg)
      enddo
      end subroutine

      subroutine relpos(ipla,ison,ijd,iek,iekk,ika)
!-----Vergleich der Positionen Pyramiden/Kammern mit Planeten,---------
!     daraus Bestimmung der Genauigkeit Fpos bzw. xyr(36) in Prozent
!     und der Polaritaet "iek" bzw. "iekk". Weitere Indizes von "xyr":
!        11: xv-xm   12: xe-xm   13: xe-xv   14: yv-ym   15: ye-ym
!        16: ye-yv   17: zv-zm   18: ze-zm   19: ze-zv   20: leer
!        21: v - m   22: e - m   23: e - v   24: q1      25: q2
!        26: q3      27: alpha'  28: beta'   29: gamma'  30: leer
!        31: x-Son   32: y-Son   33: z-Son   34: delta-s 35: M
!        36: Fpos, F'pos, F"pos
!     Indizes 11 - 19 und 21 - 29 bei "pyr" und "xyr" entsprechen sich.
      use base
      implicit double precision (a-h,o-z)

! . . Pyramidenabstaende
      xyr(11) = xyr(4)-xyr(1);  xyr(12) = xyr(7)-xyr(1)
      xyr(13) = xyr(7)-xyr(4);  xyr(14) = xyr(5)-xyr(2)
      xyr(15) = xyr(8)-xyr(2);  xyr(16) = xyr(8)-xyr(5)
      xyr(17) = xyr(6)-xyr(3);  xyr(18) = xyr(9)-xyr(3)
      xyr(19) = xyr(9)-xyr(6)
      ax = xyr(11);  ay = xyr(14)
      bx = xyr(12);  by = xyr(15)
      cx = xyr(13);  cy = xyr(16)
      if (ison==3) then
        az = z0; bz = z0; cz = z0
      else
        az = xyr(17); bz = xyr(18); cz = xyr(19)
      endif

! . . Feststellen der Polaritaet (Blickrichtung auf die Ekliptik)
!     gemaess Vorzeichen der z-Komponente des Vektorproduktes a x c.
```

```
      if (ijd==15 .or.ijd==0) then
        if (iek/=3) iek  = 1
        if (iek==3) iekk = 1
        ez = ax*cy-ay*cx
        if ((ipla==1 .and.ez>=z0).or.(ipla==2 .and. &
          ((ez<z0.and.(ika==1 .or.ika==4 .or.ika==5)).or. &
          (ez>=z0.and.(ika==2 .or.ika==3 .or.ika==6))))) then
          if (iek/=3) iek  = 2
          if (iek==3) iekk = 2
        endif
      endif

! . . Berechnung der rel. Abweichung [%] --> xyr(36)
!     Sonnenposition auf Nordsuedachse
      if (ison<=2) then
        xyr(24) = bx/ax; xyr(25) = by/ay; xyr(26) = by/bx
        s = 1.d0
        if (iek==3 .and.iekk==2) s = -1.d0
        dx1 = (xyr(24) - pyr(24))/pyr(24)
        dx2 = (xyr(25) - pyr(25))/pyr(25)
        dx3 = (xyr(26)-s*pyr(26))/pyr(26)
        xyr(36) = 100.d0 * dsqrt((dx1*dx1 + dx2*dx2 + dx3*dx3)/3.d0)
        return
      endif

!.....Relative Abweichung, Sonnenposition frei (2- und 3-dimensional)
!     Anmerkung: Bei Berechnung von F"pos (Sonnenpos. frei) laesst
!     sich statt der Strecken Mykerinos- Chefren-Pyramide u. Mykerinos-
!     Cheops-Pyramide auch ein anderes Streckenpaar verwenden, wie z.B.
!     Mykerinos- Chefren-Pyramide und Chefren- Cheops-Pyramide. F"pos
!     hat dann eventuell etwas andere Werte, aber die Minimierung von
!     F"pos liefert dieselben Zeitpunkte. Das heisst, die wesentlichen
!     Ergebnisse bleiben identisch.
      xyr(21) = dsqrt(ax*ax + ay*ay + az*az)
      xyr(22) = dsqrt(bx*bx + by*by + bz*bz)
      xyr(23) = dsqrt(cx*cx + cy*cy + cz*cz)
      xyr(24) = xyr(22)/xyr(21)
!c    xyr(25) = xyr(23)/xyr(21)
!c    xyr(26) = xyr(23)/xyr(22)
      xyr(27) = dacos((ax*bx + ay*by + az*bz)/(xyr(21) * xyr(22)))
!c    xyr(28) = dacos((ax*cx + ay*cy + az*cz)/(xyr(21) * xyr(23)))
!c    xyr(29) = dacos((bx*cx + by*cy + bz*cz)/(xyr(22) * xyr(23)))
      dx1 = (xyr(24)-pyr(24))/pyr(24)
      dx2 = xyr(27)-pyr(27)
      xyr(36) = 100.d0 * dsqrt((dx1*dx1 + dx2*dx2)*0.5d0)
      end subroutine

      subroutine sonpos(ison,iek,ix,xp3,yp3,zp3, &
        rcm,dmi,iter,iw,ke,m,n,f,x,e,w,y,z)
!-----Bestimmung von Sonnenposition und Massstab --> xyr(31 - 35)------
!     Indizes von xyr wie in relpos
      use base
      implicit double precision (a-h,o-z)
      dimension :: D(3,3),xsta(n),ysta(m),rcm(3)
      dimension :: x(n),e(n),iw(100),f(m),y(m),z(m),w(1000)

!.....Zweidimensionale Berechnung der Sonnenpos. (x- und y-Koord.)
!     Projektion der Planetenpositionen in die Ekliptikebene.
!     Zusammengehoerige Pyramiden- und Planetenabstaende werden paral-
```



```
!     lel ausgerichtet und in der Mitte zur Deckung gebracht. (Wegen
!     des gemeinsamen Massstabsfaktors "zmas" haben die entsprechenden
!     Strecken leicht unterschiedliche Laengen.)
      em = 1.d0
      if (iek==2) em = -1.d0
      if (ison<=3) then
        sax = (xyr(4)+xyr(1)) * .5d0
        say = (xyr(5)+xyr(2)) * .5d0
        sbx = (xyr(7)+xyr(1)) * .5d0
        sby = (xyr(8)+xyr(2)) * .5d0
        scx = (xyr(7)+xyr(4)) * .5d0
        scy = (xyr(8)+xyr(5)) * .5d0
        al1 = - em * pyr(31) - datan(ay/ax) + datan(say/sax)
        al2 = - em * pyr(32) - datan(by/bx) + datan(sby/sbx)
        al3 = - em * pyr(33) - datan(cy/cx) + datan(scy/scx)
        r1 = dsqrt(sax*sax + say*say)
        r2 = dsqrt(sbx*sbx + sby*sby)
        r3 = dsqrt(scx*scx + scy*scy)
        zmas = (pyr(21)/xyr(21)+pyr(22)/xyr(22)+pyr(23)/xyr(23))/3.d0
        xso1 = - r1 * zmas * dcos(al1) + pyr(34)
        xso2 = - r2 * zmas * dcos(al2) + pyr(36)
        xso3 = - r3 * zmas * dcos(al3) + pyr(38)
        yso1 = - r1 * zmas * dsin(al1) + pyr(35) * em
        yso2 = - r2 * zmas * dsin(al2) + pyr(37) * em
        yso3 = - r3 * zmas * dsin(al3) + pyr(39) * em
        xyr(31) = (xso1 + xso2 + xso3)/3.d0
        xyr(32) = (yso1 + yso2 + yso3)/3.d0
        if (iek==2) xyr(32) = - xyr(32)
        xyr(33) = z0
! . . . Fehlerabschaetzung fuer die Sonnenposition
        xyr(34) = dsqrt((xyr(31)-rcm(1))**2 + (xyr(32)-rcm(2))**2) &
          * xyr(36) * 1.d-2
! . . . Massstabsfaktor (nur fuer "Sonne" suedlich der
!       dritten Pyramide, zweidimensional gerechnet.)
        xyr(35)=AE*0.25d0*(dabs(xyr(11)/pyr(11))+dabs(xyr(12)/pyr(12))&
                        + dabs(xyr(14)/pyr(14))+dabs(xyr(15)/pyr(15)))
      endif

!.....Dreidimensionale Berechnung (x-, y- und z-Koordinate)
!     Loesung eines linearen inhomogenen Gleichungssystems bzgl. der
!     Planetenpositionen und Uebertragung des Ergebnisses auf die
!     Pyramidenpositionen.
! . . Erzeugung eines (schiefwinkligen) Vektordreibeins fuer die Pla-
!     neten (mit Hilfe des Vektorproduktes). Die 3 Vektoren bilden
!     dann die Spalten der Koeffizienten-Matrix.
      if (ison==4) then
        D(1,1) = ax
        D(2,1) = ay
        D(3,1) = az
        D(1,2) = bx
        D(2,2) = by
        D(3,2) = bz
        dx = by*az - ay*bz
        dy = ax*bz - bx*az
        dz = bx*ay - ax*by
        aba = dsqrt(ax*ax + ay*ay + az*az)
        abb = dsqrt(bx*bx + by*by + bz*bz)
        abd = dsqrt(dx*dx + dy*dy + dz*dz)
        dfakt = (aba + abb) * 0.5d0/abd
```

```
          D(1,3) = dx * dfakt
          D(2,3) = dy * dfakt
          D(3,3) = dz * dfakt
   ! . . . Inversion der Matrix D
          call invert(D)
   ! . . . Berechnung der Loesung mit x = Inv.(D) * (- Merkur-Koord.)
          x1 = - D(1,1) * xyr(1) - D(1,2) * xyr(2) - D(1,3) * xyr(3)
          x2 = - D(2,1) * xyr(1) - D(2,2) * xyr(2) - D(2,3) * xyr(3)
          x3 = - D(3,1) * xyr(1) - D(3,2) * xyr(2) - D(3,3) * xyr(3)
   ! . . . Koordinaten der Sonnenposition in Giza
          xyr(31) = x1 * pyr(11) + x2 * pyr(12) + x3 * pyr(7)
          xyr(32) = x1 * pyr(14) + x2 * pyr(15) + x3 * pyr(8)
          xyr(33) = x1 * pyr(17) + x2 * pyr(18) + x3 * pyr(9)
   ! . . . Massstabsfaktor
          xyr(35) = AE * dsqrt((xyr(12)**2 + xyr(15)**2 + xyr(18)**2)/ &
                               (pyr(12)**2 + pyr(15)**2 + pyr(18)**2))
        endif

   !.....Dreidimensionale Berechnung (x-, y- und z-Koordinate)
   !     mit Hilfe des Fit-Programms FITEX. Die Konstellation der Plane-
   !     ten wird durch Translation, Rotation und Groessenaenderung mit
   !     der Anordnung der Pyramiden bzw. der Kammern in der Cheops-Pyra-
   !     mide zur Deckung gebracht. Anschliessend wird die resultierende
   !     Transformation auf die Sonnenposition (Koordinatenursprung)
   !     angewendet.
        if (ison==5) then
          istart = 0; ke = 0
          if (iter/=0) then; do iu=ix,6,5; write(iu,*); enddo; endif

   ! . . . Koordinatentransformation --> y(i)
          do
            do i=1,m; y(i) = xyr(i); enddo
            call translat(x(1),x(2),x(3),y)
            call rotmat(5,x(4),x(5),x(6),y)
            call mastab(x(7),y)
            if (istart==0) then
              do i=1,n; xsta(i) = x(i); enddo
              do i=1,m; ysta(i) = y(i); enddo
            endif

   ! . . . . Die Fehlerquadrate dabs(F)**2
            w(4) = z0
            do i=1,m
              f(i) = y(i) - z(i)
              w(4) = w(4) + f(i)*f(i)
            enddo
            istart = istart + 1

   ! . . . . Ausgabe der Iterationen (Aufruf von FITEX)
            do iu=ix,6,5
             if (iter/=0) write(iu,152)iw(3),iw(4),w(3),w(4),(x(i),i=1,n)
            enddo
            call fitex(ke,m,n,f,x,e,w,iw)
            if (ke/=1) exit
          enddo

   ! . . . Ausgabe der Ergebnisse
          if (iter/=0) then
            do iu=ix,6,5
```



```
              write(iu,153) ke,iw(3),iw(4),w(3),w(4)
              j2 = n+n
              write(iu,154) x,(w(4+j),j=1,j2)
              if (w(5)==z0) go to 10
              j2=4+j2
   !c          do i=1,n
   !c            j1=j2+1; j2=j1+i-1
   !c            write(iu,154) (w(j),j=j1,j2)
   !c          enddo
      10      write(iu,*)
              write(iu,'(''  start   x(1..'',i1,''):'',7F13.3)') &
                n,(xsta(i),i=1,3),(xsta(i)*gdpi,i=4,6),xsta(7)
              write(iu,'(''      "    y(1..'',i1,''):'',9f13.3)') &
                m,(ysta(i),i=1,m)
              write(iu,'(''  results x(1..'',i1,''):'',7f13.3)') &
                n,(x(i),i=1,3),(x(i)*gdpi,i=4,6),x(7)
              write(iu,'(''      "    y(1..'',i1,''):'',9f13.3/)') &
                m,(y(i),i=1,m)
            enddo
          endif

   ! . . . Berechnung der Sonnenposition im Pyramidengelaende mit Hilfe
   !       der gerade bestimmten Parameter x(1)..x(7) durch Transforma-
   !       tion des Koordinatenursprungs (Sonne)
          do i=1,m; y(i) = z0; enddo
          call translat(x(1),x(2),x(3),y)
          call rotmat(5,x(4),x(5),x(6),y)
          call mastab(x(7),y)
          xyr(31) = y(1)
          xyr(32) = y(2)
          xyr(33) = y(3)
          xyr(35) = AE/x(7)
        endif

        if (ison>=4) then
   !.......Korrektur der Koordinaten (1/4 Hoehe oder ganze Hoehe der
   !       3. Pyramide bzw. Positionskoordinaten der Felsenkammer)
          xyr(31) = xyr(31) + xp3
          xyr(32) = xyr(32) + yp3
          xyr(33) = xyr(33) + zp3

   ! . . . Fehlerabschaetzung fuer die Sonnenposition
   !c      if (ison==4) then
            dcm = dsqrt((xyr(31)-rcm(1))**2 + (xyr(32)-rcm(2))**2 &
                      + (xyr(33)-rcm(3))**2)
            qu = dcm
            if (dcm<dmi) qu = dmi * ((dcm/dmi)**2 + 1.d0)*0.5d0
            xyr(34) = qu * xyr(36) * 1.d-2
   !c      else
   !c        xyr(34) = dsqrt(w(4))
   !c      endif
        endif

        return
    152 format(5x,2i5,1p,9e13.5)
    153 format(3i5,1p,8e23.15)
    154 format('  ',1p,6e13.5)
        end subroutine
```

```
      subroutine invert(a)
!-----Inversion der 3x3-Matrix a, d.h. a -> inv(a)---------------------
      implicit double precision (a-h,o-z)
      dimension :: a(3,3),b(3,3)
! . . Die Kofaktoren
      b(1,1) =   a(2,2)*a(3,3) - a(3,2)*a(2,3)
      b(1,2) = - a(2,1)*a(3,3) + a(3,1)*a(2,3)
      b(1,3) =   a(2,1)*a(3,2) - a(3,1)*a(2,2)
      b(2,1) = - a(1,2)*a(3,3) + a(3,2)*a(1,3)
      b(2,2) =   a(1,1)*a(3,3) - a(3,1)*a(1,3)
      b(2,3) = - a(1,1)*a(3,2) + a(3,1)*a(1,2)
      b(3,1) =   a(1,2)*a(2,3) - a(2,2)*a(1,3)
      b(3,2) = - a(1,1)*a(2,3) + a(2,1)*a(1,3)
      b(3,3) =   a(1,1)*a(2,2) - a(2,1)*a(1,2)
! . . Kehrwert der Determinante und Transponieren
      dei = 1.d0/(a(1,1)*b(1,1) + a(1,2)*b(1,2) + a(1,3)*b(1,3))
      do i=1,3; do j=1,3; a(i,j) = b(j,i)*dei; enddo; enddo
      end subroutine

      subroutine rotmat(iachse,w1,w2,w3,a)
!-----Erstellung der Dreh-Matrix und Multiplikation--------------------
!     3 Vektoren fuer Merkur bis Erde: a(1..9) --> a(1..9)
!     iachse = 1-3:  Drehung um x-, y- oder z-Achse (Winkel w1)
!
!                                  (  cos w1  sin w1   0  )
!                   z.B.   Dz(w1) = ( -sin w1  cos w1   0  )
!                                  (    0       0      1  )
!
!     iachse = 4:    Drehung um Knotenlinie (Winkel w1, w2)
!     iachse = 5:    Drehung um beliebige Achse (Winkel w1, w2
!                    und w3: die Eulerschen Winkel)
      implicit double precision (a-h,o-z)
      dimension :: a(9),b(9),D(3,3)
      z0 = 0.d0
      one = 1.d0
      s1 = dsin(w1); c1 = dcos(w1)
      if (iachse<=3) then
        do j=1,3; do i=1,3; D(i,j) = z0; enddo; enddo
        if (iachse==1) then
          D(1,1) = one           ! axis 1
          D(2,2) = c1
          D(2,3) = s1
          D(3,2) = - s1
          D(3,3) = c1
        else
          D(1,1) = c1
          if (iachse==2) then
            D(1,3) = s1          ! axis 2
            D(2,2) = one
            D(3,1) = - s1
            D(3,3) = c1
          else
            D(1,2) = s1          ! axis 3
            D(2,1) = - s1
            D(2,2) = c1
            D(3,3) = one
          endif
        endif
      else
```



```
        s2 = dsin(w2); c2 = dcos(w2)
        if (iachse==4) then
          D(1,1) = - s1 * s1 * (one - c2) + one    ! axis 4
          D(1,2) =   s1 * c1 * (one - c2)
          D(1,3) = - s1 * s2
          D(2,1) =   s1 * c1 * (one - c2)
          D(2,2) = - c1 * c1 * (one - c2) + one
          D(2,3) =   c1 * s2
        else
          s3 = dsin(w3); c3 = dcos(w3)
          D(1,1) =   c1 * c3 - s1 * c2 * s3        ! axis 5
          D(1,2) =   s1 * c3 + c1 * c2 * s3
          D(1,3) =   s2 * s3
          D(2,1) = - c1 * s3 - s1 * c2 * c3
          D(2,2) = - s1 * s3 + c1 * c2 * c3
          D(2,3) =   s2 * c3
        endif
        D(3,1) =   s1 * s2
        D(3,2) = - c1 * s2
        D(3,3) =   c2
      endif

! . . Ausfuehrung der Transformation (Merkur-, Venus- und Erdposition)
!c    do i = 1,3; write(6,'(3f13.8)')(D(i,j),j=1,3); enddo
      do i=1,9; b(i) = z0; enddo
      do k=0,6,3
        do i=1,3
          do j=1,3
            b(k+i) = b(k+i) + D(i,j)*a(j+k)
          enddo
        enddo
      enddo
      do i=1,9; a(i) = b(i); enddo
!c    write(6,'(a12,3f13.8)') '  Mercury : ',(a(j),j=1,3)
!c    write(6,'(a12,3f13.8)') '  Venus   : ',(a(j),j=4,6)
!c    write(6,'(a12,3f13.8)') '  Earth   : ',(a(j),j=7,9)
      end subroutine

      subroutine translat(a1,a2,a3,a)
!-----Translation der Positionen der 3 Planeten-----------------------
!     3 Vektoren a(1..9) --> a(1..9)
      implicit double precision (a-h,o-z)
      dimension :: a(9)
      do i=1,7,3
        a(i) = a(i)+a1; a(i+1) = a(i+1)+a2; a(i+2) = a(i+2)+a3
      enddo
      end subroutine

      subroutine mastab(zmas,a)
!-----Massstabsaenderung----------------------------------------------
!     3 Vektoren a(1..9) --> a(1..9)
      implicit double precision (a-h,o-z)
      dimension :: a(9)
      do i=1,9; a(i) = zmas * a(i); enddo
      end subroutine

      subroutine transfo(irb,rku)
!-----Transformation ins Merkurbahn-System (Venusbahn-System)----------
!     re(1..9) --> re(1..9),  xyr(1..9) --> xyr(1..9)
```

```
!     Die Transformationen A, B und C liefern dasselbe Ergebnis.
!     Die Eingabewinkel ao, ai, at sind im Modul "base" gespeichert.
      use base
      implicit double precision (a-h,o-z)
      dimension :: xyt(9),rku(3)
      pi2 = pi * 2.d0
      if (irb>=2 .and.irb<=4) then
        ao = (re(34) - re(1))*pidg
      else
        ao = (re(40) - re(1))*pidg
      endif
      if (ao<z0) ao = ao + pi2
      if (ao>pi2) ao = ao - pi2
!c    write(6,'(a10,f23.8)') '  re(4)   ',re(4)
!c    write(6,'(a10,f23.8)') '  re(40)  ',re(40)
      if (irb>=2 .and.irb<=4) then
        ai = dabs(datan(xyr(3)/(xyr(1)*dsin(ao))))
      else
        rxy = dsqrt(xyr(4)*xyr(4) + xyr(5)*xyr(5))
        aov = (re(40) - re(4))*pidg
        ai  = dabs(datan(xyr(6)/(rxy*dsin(aov))))
      endif
      at = dasin(dsin(ao)/dsqrt(1.d0-(dsin(ai)*dcos(ao))**2))+ao-pi
      a1 = ao; a2 = ai
      a3 = at
!c    write(6,'(a12,3f13.8)') '  Mercury : ',(xyr(j),j=1,3)
!c    write(6,'(a12,3f13.8)') '  Venus   : ',(xyr(j+3),j=1,3)
!c    write(6,'(a12,3f13.8)') '  Earth   : ',(xyr(j+6),j=1,3)
      do i=1,9; xyt(i) = xyr(i); enddo

!.....Transformation A  -->  Dz(at) * K(ao,ai)
!     (Reihenfolge der Matrizen von rechts nach links!)
      if (irb==2 .or.irb==5) then
! . . . Matrix K(ao,ai)
        call rotmat(4,a1,a2,z0,xyt)
! . . . Matrix Dz(at)
        if (irb==5) then
          at = datan(xyt(2)/xyt(1))
          a3 = at
        endif
        call rotmat(3,a3,z0,z0,xyt)
      endif

!.....Transformation B  -->  Dz(at-ao) * Dx(ai) * Dz(ao)
      if (irb==3) then
! . . . Matrix Dz(ao)
        call rotmat(3,a1,z0,z0,xyt)
! . . . Matrix Dx(ai)
        call rotmat(1,a2,z0,z0,xyt)
! . . . Matrix Dz(at-ao)
        call rotmat(3,a3-a1,z0,z0,xyt)
      endif

!.....Transformation C  -->  R(ao,ai,at-ao)
      if (irb==4) then
! . . . Matrix R(ao,ai,at-ao)
        call rotmat(5,a1,a2,a3-a1,xyt)
      endif
```



```
! . . Ruecktransformation in Kugelkoordinaten
      do i=1,9; xyr(i) = xyt(i); enddo
      do i=1,3
        k=3*(i-1)
        xy1 = xyr(k+1)
        xy2 = xyr(k+2)
        xy3 = xyr(k+3)
        call kugelko(xy1,xy2,xy3,rku)
        do j=1,3
          re(k+j) = rku(j)
        enddo
      enddo
      end subroutine

      subroutine kugelko(r1,r2,r3,rku)
!-----Umrechnung in Kugelkoordinaten rku(1)..rku(3)-------------------
!     (Index von rku  1: phi, 2: theta, 3: r)
      use base, only : gdpi
      implicit double precision (a-h,o-z)
      dimension :: rku(3)
      ra = dsqrt(r1*r1 + r2*r2)
      rku(1) = datan(r2/r1)*gdpi
      rku(2) = datan(r3/ra)*gdpi
      rku(3) = dsqrt(ra*ra + r3*r3)
      if (r1<0.d0) rku(1) = rku(1) + 180.d0
      if (rku(1)<0.d0) rku(1) = rku(1) + 360.d0
      end subroutine

      subroutine aphelko(imod,ivers,iaph,ipla, &
        ison,ijd,io,iop0,ix,dh3,x,y,rcm,dmi)
!-----Berechnung der "Merkur-Aphelposition" in Giza-------------------
!     fuer Konstell. 13, 14, sowie "quick start option" 371 und 372.
!     Die Berechnung kann mit VSOP87A (ivers=1) und VSOP87C (ivers=3)
!     durchgefuehrt werden. Die Ortsabweichungen im Pyramidengelaende
!     zwischen beiden Versionen liegen fuer Konst. 13 bzw. 14 bei ca.
!     10 cm und 5 mm, bei der "Schatten-Konstellation 12" bei ca. 4 mm.
!     Sollte sich an den Zeitpunkten dieser Konstellationen etwas aen-
!     dern, sind die astron. Aphelkoordinaten in "aphelm" anzupassen.
      use base
      implicit double precision (a-h,o-z)
      dimension :: aphelm(18),x(7),y(9),rcm(3)

!.....Sphaerische ekliptikale Koordinaten L, B und r des Merkur-Aphels
!     fuer Konst. 13 und 14 jeweils fuer J2000.0 und Ekl. der Epoche
!     und fuer "Schatten-Konstellation 12" mit J2000.0 (Option 372)
!     und Ekliptik der Epoche (Option 371).
!
! . . A. Berechnung mit Gl. (7.1) --> Konst. 13:  JDE = 5909973.28368
!                                     Konst. 14:  JDE =  671046.63581
!                               Optionen 371 und 372:  JDE = 2849071.14941
!     data aphelm/
!       272.2596751d0, -5.4263369d0, 0.4672908784d0,   (K.13, VSOP87A)
!        46.8137077d0, -6.4048699d0, 0.4670482474d0,   (K.13, VSOP87C)
!       249.5729904d0, -1.9354192d0, 0.4662991040d0,   (K.14, VSOP87A)
!       182.1787524d0, -1.3530604d0, 0.4662950222d0,.. (K.14, VSOP87C)
!
! . . B. r(Mer.) optimiert --> Konst. 13 (VSOP87A): JDE = 5909973.264
!        (r maximal fuer Aphel)          (VSOP87C): JDE = 5909973.255
!                              Konst. 14 (VSOP87A/C): JDE =  671046.632
```

```
      data aphelm/272.2054713d0, -5.4229877d0, 0.4672909313d0, &
                   46.7345218d0, -6.4007584d0, 0.4670483641d0, &
                  249.5625348d0, -1.9341303d0, 0.4662991059d0, &
                  182.1682931d0, -1.3518259d0, 0.4662950244d0, &
                  258.9945271d0, -3.6947988d0, 0.4667842406d0, &
                  274.2350325d0, -3.8355115d0, 0.4667842399d0/
      if ((ijd==13 .or.ijd==14 .or.iop0==371 .or.iop0==372).and. &
       imod<=2 .and.ison==5 .and.iaph==1 .and.ipla==1 .and.io==2) then
        if (ijd==13 .and.ivers==1) j = 1
        if (ijd==13 .and.ivers/=1) j = 4
        if (ijd==14 .and.ivers==1) j = 7
        if (ijd==14 .and.ivers/=1) j = 10
        if (iop0==371) j = 16
        if (iop0==372) j = 13
        do i=4,6; re(i) = aphelm(j+i-4); enddo
!       Umrechnung in kartesische Koordinaten
        call kartko(ison)
!       Koordinatentransformation: Weltraum --> Pyramidengelaende
        do i=4,6; y(i) = xyr(i); enddo
        call translat(x(1),x(2),x(3),y)
        call rotmat(5,x(4),x(5),x(6),y)
        call mastab(x(7),y)
        y(6) = y(6) + dh3
!       Fehler in Metern (dr)
        dcm = dsqrt((y(4)-rcm(1))**2 + (y(5)-rcm(2))**2 &
                  + (y(6)-rcm(3))**2)
        qu = dcm
        if (dcm<dmi) qu = dmi * ((dcm/dmi)**2 + 1.d0)*0.5d0
        dr = qu * xyr(36) * 1.d-2
!       Ausgabe des Ergebnisses
        do iu=ix,6,5
          write(iu,'(''  Mercury aphelion coordinates [m]:'', &
            & f13.2,2f10.2,f9.2)') y(4),y(5),y(6),dr
          call linie(iu,1)
        enddo
      endif
      end subroutine

      subroutine plako(diff,ipla,ijd,ik,ison,ipos, &
        rcm,x,y,ort,rp,dd,dn,dss,pla,plan,emp,text,tt,titab, &
        is12,dmi,zjda,zjde,ivers,md,ix,prec,lu,r,ierr,rku)
!-----Koordinaten fuer Merkur bis Neptun-------------------------------
!     und Berechnung der "Planetenpositionen" im Giza-Gelaende fuer
!     Konst. 1-14 mit ison = 5 (FITEX) und imod = 2 (VSOP87-Vollv.).
!     Zusaetzlich:
!     Spezialausgabe fuer Konst. 12 mit iuniv = 1 (TT) und iout = 3
!     (spezial). In diesem Fall sind nur noch folgende Parameter
!     variierbar: ipla (Pyr.- oder Kammerpositionen), imod (VSOP87
!     Voll- oder Kurzv.), ivers (VSOP87A oder VSOP87C, bei Vollv.)
!     und ihi (z-Koordinate)
      use base
      implicit double precision (a-h,o-z)
      dimension :: diff(9),r(6),rku(3),md(0:9),x(7),y(9),rcm(3)
      dimension :: ort(0:9,4),rp(3,4),zjda(4)
      character(2) :: dd,dn,dss
      character(3) :: pla(0:9),line
      character(7) :: emp
      character(10) :: plan(0:9)
      character(18) :: date(4)
```



```
      character(23) :: text(0:9),tt(2)
      character(49) :: titab
      data date/'date of chambers: ','date of syzygy:   ', &
               'date of transit:  ','date of pyramids: '/
      data line/'---'/

! . . Tabellenkopf
      do iu=ix,6,5
        if (is12==0) then
          write(iu,*); call linie(iu,1)
          write(iu,*)'pla.   x[AU]     y[AU]     z[AU]      L       ',&
            'B       r[AU]     Lm-L     dev.'
          call linie(iu,2)
        else
          write(iu,'(/27x,''Celestial positions in Giza'')')
          call linie(iu,1)
          write(iu,*)' body          x[m]      y[m]     z[m]', &
            '   dr[m]       latitude N    longitude E'
        endif
      enddo

!.....Positionen von Merkur bis Neptun und Sonne im Pyramiden-
!     gelaende und im System innerhalb der Cheops-Pyramide (nur
!     VSOP87-Vollversion)
      icm = 1; imax = 8; if (ivers==1) imax = 9
      if (is12/=0) imax = 4
      icmax = 1; if (is12/=0) icmax = 4
   10 if (is12/=0) then
        zjde = zjda(icm)
        do iu=ix,6,5
          call linie(iu,2)
          write(iu,'(4x,a18,''JDE ='',f14.5)') date(icm),zjda(icm)
          call linie(iu,2)
        enddo
      endif
      if (is12/=0 .and.icm==1) then
        if (ipla==1) then
          call geoko(ort(0,1),-ort(0,2),ipla,iB1,zB2,iL1,zL2)
        else
          call geoko(ort(0,1),ort(0,3),ipla,iB1,zB2,iL1,zL2)
        endif
        do iu=ix,6,5
          write(iu,102) plan(0),(ort(0,j),j=1,4),iB1,zB2,iL1,zL2
        enddo
      endif
      do 20 id=1,imax
        call vsop2(zjde,ivers,id,md,ix,prec,lu,r,ierr,rku)
        dif = re(1) - rku(1); call reduz(dif,0,0)
        err = dif-diff(id); call reduz(err,0,0)
        if (is12==0) then
          do iu=ix,6,5
            if (id/=4 .and.(id<=6 .or.id==9)) then
             write(iu,100)pla(id),(r(i),i=1,3),(rku(i),i=1,3),dif,err
            else
             write(iu,101)pla(id),(r(i),i=1,3),(rku(i),i=1,3),dif,emp
            endif
          enddo
        endif
```

```
!....."Planetenpositionen" im Giza-Gelaende (kartesische Koord.)
        if (((ijd>=1 .and.ijd<=14).or.(ik==4519 .and.ipla==1) &
           .or.(ik==4518 .and.ipla==2)).and.ison==5) ipos = 1
        if (ipos==1) then
          if (id==1) then
            do j=1,3; y(j) = rku(j); enddo
          endif
          do j=1,3; re(j+3) = rku(j); enddo
          call kartko(ison)
          do j=4,6; y(j) = xyr(j); enddo
          call translat(x(1),x(2),x(3),y)
          call rotmat(5,x(4),x(5),x(6),y)
          call mastab(x(7),y)
          do j=1,3; ort(id,j) = y(3+j) + rp(3,j); enddo
!         Genauigkeit der "Planetenpositionen"
          if (id<=3 .and.is12==0) then
            ort(id,4) = dsqrt((ort(id,1)-rp(4-id,1))**2 &
                            + (ort(id,2)-rp(4-id,2))**2 &
                            + (ort(id,3)-rp(4-id,3))**2)
          elseif (id==9 .and.is12==0) then
            ort(id,4) = dsqrt((ort(id,1)-rp(1,1))**2 &
                            + (ort(id,2)-rp(1,2))**2 &
                            + (ort(id,3)-rp(1,3))**2)
          else
            dcm = dsqrt((ort(id,1)-rcm(1))**2 &
                      + (ort(id,2)-rcm(2))**2 &
                      + (ort(id,3)-rcm(3))**2)
            qu = dcm
            if (dcm<dmi) qu = dmi * ((dcm/dmi)**2 + 1.d0)*0.5d0
            ort(id,4) = qu * xyr(36) * 1.d-2
          endif
!         Geographische Koordinaten (Laenge und Breite) der
!         transformierten Sonnen- und Planetenpositionen
          if (is12/=0) then
            if (ipla==1) then
              call geoko(ort(id,1),-ort(id,2),ipla,iB1,zB2,iL1,zL2)
            else
              call geoko(ort(id,1),ort(id,3),ipla,iB1,zB2,iL1,zL2)
            endif
            do iu=ix,6,5
             write(iu,102) plan(id),(ort(id,j),j=1,4),iB1,zB2,iL1,zL2
            enddo
          endif
        endif
   20 enddo
!     Ruecksprung zum naechsten Planeten
      icm = icm + 1
      if (icm<=icmax) go to 10

! . . Weitere Ergebnis-Ausgabe
      if (ipos==1 .and.is12==0) then
        text(2) = tt(ipla)
        do iu=ix,6,5
          call linie(iu,1)
          write(iu,'(''  Celestial pos. in Giza'',4x,a49)')titab
          call linie(iu,2)
          write(iu,'(''  Local coordinates'',9x,''Sun       '', &
            & f10.2,2f10.2,f9.2)') (ort(0,j),j=1,4)
        enddo
```



```
          do i=1,imax
            dd = dn
            if ((i>=1 .and.i<=3).or.i==9) dd = dss
            do iu=ix,6,5
              write(iu,'(a23,5x,a10,3f10.2,f9.2,a2)') &
              text(i),plan(i),(ort(i,j),j=1,4),dd
            enddo
          enddo
        endif
        do iu=ix,6,5; call linie(iu,1); enddo

        return
  100 format(1x,a3,3f10.6,f9.4,f8.4,f10.6,2f9.4)
  101 format(1x,a3,3f10.6,f9.4,f8.4,f10.6,f9.4,1x,a7)
  102 format(2x,a10,f9.2,f10.2,f9.2,f7.2,i8,f9.4,i6,f8.4)
! . . Groessere Stellenanzahl fuer Schnellstart-Optionen 3 und 8
!f100 format(2x,a3,f11.6,2f10.6/28x,f13.7,f11.7,f14.10/58x,f13.7,f8.3)
!f101 format(2x,a3,f11.6,2f10.6/28x,f13.7,f11.7,f14.10/58x,f13.7,a8)
      end subroutine

      subroutine geoko(x,y,ipla,iB1,zB2,iL1,zL2)
!-----Berechnung der geographischen Koordinaten-----------------------
!     (iB1,zB2 und iL1,zL2, jeweils in Grad und Minuten)
      use base, only : pi,pidg,R3a,R3p
      implicit double precision (a-h,o-z)

! . . Erdumfang ueber Pole. Anstelle von Ue = 40008 km folgt
!     Ellipsenumfang nach Srinivasa Ramanujan.
      zl = 3.d0*((R3a-R3p)/(R3a+R3p))**2
      Ue = pi*(R3a+R3p) * (1.d0 + zl/(10.d0 + dsqrt(4.d0-zl)))
!     Geographische Position des Koordinatenursprungs (Pyr./Kam.)
      if (ipla==1) then
        zB0 = 29.972530d0  !  Zentrum der Mykerinos-Pyramide
        zL0 = 31.128243d0  !  (Pyramiden-Koordinaten)
      else
        zB0 = 29.979200d0  !  Mittelachse der Ostwand
        zL0 = 31.134276d0  !  der Koeniginnenkammer
      endif

! . . Geographische Breite (zB)
      dBa = 360.d0 * x/Ue
      zBa = zB0 + dBa
      call geokar(zBa,ua,va)
      call geokar(zB0,u0,v0)
      xa  = dsqrt((ua-u0)**2 + (va-v0)**2)
      dB  = dBa * dabs(x/xa)
      zB  = zB0 + dB
      iB1 = idint(zB)
      zB2 = dmod(zB,1.d0)*60.d0

! . . Geographische Laenge (zL)
      zBm = 0.5d0*(zB + zB0)
      call geokar(zBm,um,vm)
      dL  = y/(pidg*um)
      zL  = zL0 + dL
      iL1 = idint(zL)
      zL2 = dmod(zL,1.d0)*60.d0
      end subroutine
```

```
      subroutine geokar(B,u,v)
!-----Abstand eines Punktes der geographischen Breite B----------------
!     zur Erdachse (u) und zur Aequatorebene (v) (kartesische Koord.)
      use base, only : pidg,R3a,R3p
      implicit double precision (a-h,o-z)
      u = R3a/dsqrt(1.d0 + (dtan(B*pidg)*R3p/R3a)**2)
      v = R3p*dsqrt(1.d0 - (u/R3a)**2)
      end subroutine

      subroutine reduz(a,i,j)
!-----Winkelreduzierung  a --> a  (z.B. 387 Grad --> 27 Grad)----------
!     i = 0/1: dezimale Grad/ Bogenmass
!     j = 0:   a --> -180...180 Grad
!     j = 1:   a -->    0...360 Grad
      use base, only : pidg,gdpi
      implicit double precision (a-h,o-z)
      u360 = 360.d0
      z1 = 1.d0
      if (a<0.d0) z1 = -1.d0
      if (i/=0) a = a*gdpi
      ab = dabs(a)
      if (ab>u360) ab = dmod(ab,u360)
      if ((j==0 .and.ab>180.d0).or. &
          (j==1 .and.a<0.d0)) ab = ab - u360
      a = z1 * ab
      if (i/=0) a = a * pidg
      end subroutine

      subroutine memo(zz1,zz2,zz3,zz4,zz5,zz6,zz7,zmem,ik,imem)
!-----Ergebnis-Parameter merken----------------------------------------
      use base, only : re
      implicit double precision (a-h,o-z)
      dimension :: zmem(78)
      zmem(1) = zz1
      zmem(2) = zz2
      zmem(3) = zz3
      zmem(4) = zz4
      zmem(5) = zz5
      zmem(6) = zz6
      zmem(7) = zz7
      do i=1,12; zmem(10+i) = re(i); enddo
      do i=31,78; zmem(i) = re(i); enddo
      imem = ik
      end subroutine

      subroutine info
!-----Information zu den Copyrights (aus der Datei "inpdata.t")--------
      character(70) :: itext(37)
      open(unit=10,file='inpdata.t')
      do i=1,105; read(10,*); enddo
      do i=1,37; read(10,*) itext(i); enddo
      close(10)
      write(6,'(///37(5x,a70/))') (itext(i),i=1,37)
      end subroutine

      subroutine titel1(iaph,ijd,ia,ison,ipla, &
        ilin,isep,nurtr,iuniv,is12,iop0)
!-----Haupttitel und Untertitel----------------------------------------
      implicit double precision (a-h,o-z)
```



```
      write(ia,*)
      if (iop0==350) then
        write(ia,'(20x,A20,A22)')'4 PLANETS IN A LINE ', &
             '(SYZYGY), 17. MAY 3088'
        go to 20
      elseif (iop0==351) then
        write(ia,'(17x,A16,A31)')'MERCURY TRANSIT ', &
             '(MIN. SEPARATION), 18. MAY 3088'
        go to 20
      elseif (iop0==360) then
        write(ia,'(18x,A14,A32)')'VENUS TRANSIT ', &
             '(MIN. SEPARATION), 18. DEC. 3089'
        go to 20
      elseif (iop0==361) then
        write(ia,'(19x,A20,A23)')'3 PLANETS IN A LINE ', &
             '(SYZYGY), 23. DEC. 3089'
        go to 20
      elseif (iop0==370) then
        write(ia,'(24x,A34)')'SEARCH FOR "SHADOW-CONSTELLATIONS"'
        go to 10
      elseif (iop0==371 .or.iop0==372) then
        write(ia,'(16x,A20,A29)')'PRECEDING "SHADOW-CO', &
             'NSTELLATION" 12, 22. MAY 3088'
        go to 20
      endif
      if (ipla==1) write(ia,*)'                 PLANETS IN ', &
          'ALIGNMENT WITH THE PYRAMIDS OF GIZA'
      if (ipla==2) write(ia,*)'          PLANETS IN ALIGNME', &
          'NT WITH THE CHAMBERS OF THE CHEOPS PYRAMID'
      if (ipla==3) then
        if (ilin>=3) write(ia,'(28x,a11,a15)')'PLANETS IN ', &
          'A LINE (SYZYGY)'
        if (ilin==1) write(ia,'(31x,a19)')'TRANSITS OF MERCURY'
        if (ilin==2) write(ia,'(32x,a17)')'TRANSITS OF VENUS'
      endif
   10 if (ipla/=3 .and.is12==0) then
        if (iaph==1 .and.ijd/=13 .and.ijd/=14) &
          write(ia,'(30x,a21)')'(Mercury at aphelion)'
        if (iaph==2 .and.ijd/=13 .and.ijd/=14) &
          write(ia,'(29x,a23)')'(Mercury at perihelion)'
        if (iaph==3 .or.(iaph==1 .and.(ijd==13 .or.ijd==14))) &
          write(ia,'(29x,a23)')'(Mercury near aphelion)'
        if (iaph==4 .or.(iaph==2 .and.(ijd==13 .or.ijd==14))) &
          write(ia,'(28x,a25)')'(Mercury near perihelion)'
        if (iaph==5) write(ia,'(24x,a34)') &
           '(time not restricted, F minimized)'
      elseif (ipla/=3 .and.is12/=0) then
        if (ipla==1) write(ia,'(17x,a48)') &
          '(more positions - coordinate system of pyramids)'
        if (ipla==2) write(ia,'(17x,a48)') &
          '(more positions - coordinate system of chambers)'
      else
        if (isep==1) then
          if (ison/=5) then
            write(ia,'(14x,a21,a33)')'(eclipt. longitudes, ', &
              'all within an angular range, JDE)'
          else
            if (ilin>=3) then
              if (nurtr==1) then
```

```
              write(ia,'(13x,a18,a37)')'(angular range of ', &
              'eclipt. longitudes dL minimized, JDE)'
            else
              write(ia,'(5x,a18,a52)')'(angular range of ', &
              'eclipt. longitudes dL minimized, only transits, JDE)'
            endif
          else
            write(ia,'(11x,a18,a41)')'(equal eclipt. lon', &
              'gitudes for Earth und transit planet, TT)'
          endif
        endif
      elseif (isep==2) then
        write(ia,'(14x,a54)') &
          '(minimum separation, without travel time of light, TT)'
      else
        if (iuniv==1) then
          write(ia,'(17x,a48)') &
            '(geocentric transit phases, terrestrial time TT)'
        else
          write(ia,'(18x,a46)') &
            '(geocentric transit phases, universal time UT)'
        endif
      endif
    endif
 20 if (isep/=4) then
      write(ia,'(34x,a8,i4,a2)')'< option',iop0,' >'
    else
      write(ia,'(11x,a8,i4,a47)')'< option',iop0, &
      ' >  (monitor line width minimal 148 characters)'
    endif
    end subroutine

    subroutine titel2(ia,imod,ivers,irb,ipla, &
      ison,ihi,iek,ijd,ika,iaph,ilin,ical,ak,zjdel,zjahr,delt, &
      dwi,dwikomb,dwi0,dwi2,dwi3,iamax,step,ikomb,zmin,zmax)
!-----Zwei weitere Titelzeilen-----------------------------------------
    implicit double precision (a-h,o-z)
    dimension :: ida(7),da(7)
    character(5)  :: ca(2),dmo
    character(7)  :: cal(2)
    character(10) :: wd
    character(15) :: text0
    character(27) :: text1
    character(19) :: text2
    character(8)  :: text3(0:6)
    character(25) :: text4
    character(22) :: text5(2)
    data ca/' (c1)',' (c2)'/,cal/'Gregor.','Julian.'/
    data text3/ '        ',' E-V-M, ',' E-M-V, ', &
     ' V-E-M, ',' V-M-E, ',' M-E-V, ',' M-V-E, '/
    data text5/',  only Greg. calendar',',  Jul./Greg. calendar'/
    if (imod==1)  text1 = '  VSOP87D short ver.(Meeus)'
    if (imod==2 .and.ivers==1) text1='  VSOP87A (2005) full ver.,'
    if (imod==2 .and.ivers==3) text1='  VSOP87C (2005) full ver.,'
    if (imod==3)  text1 = '  "Keplers equation",       '
    if (ikomb==1 .and.ivers==1)text1='  VSOP87A, comb. search,   '
    if (ikomb==1 .and.ivers==3)text1='  VSOP87C, comb. search,   '
    if (ivers==1) text2 = '  standard J2000.0,'
    if (ivers==3) text2 = '  ecliptic of date,'
```



```
    if (ipla/=3) then
     if (irb==1) then
      if (ison==1) text4 = ' "Sun" south of Myker. P.'
      if (imod==3 .and.ipla==2) text4 ='"Sun" south of sub. cham.'
      if (ison==2) text4 = '"Sun" south of Chefren P.'
      if (ison==3) text4 = '"Sun position" free, 2D  '
      if (ipla==1) then
       if (ison==4 .and.ihi==1) text4 ='"Sun" free, 3D, base, SLE'
       if (ison==4 .and.ihi==2) text4 =' "Sun" free, 3D, C-M, SLE'
       if (ison==4 .and.ihi==3) text4 =' "Sun" free, 3D, top, SLE'
       if (ison==5 .and.ihi==1) text4 ='"Sun" free 3D base, FITEX'
       if (ison==5 .and.ihi==2) text4 ='"Sun" free 3D, C-M, FITEX'
       if (ison==5 .and.ihi==3) text4 ='"Sun" free 3D, top, FITEX'
      endif
      if (ipla==2) then
       if (ison==4 .and.ihi==1) text4 ='"Sun" free, 3D, East, SLE'
       if (ison==4 .and.ihi==2) text4 ='"Sun" free, 3D, mid., SLE'
       if (ison==4 .and.ihi==3) text4 ='"Sun" free, 3D, West, SLE'
       if (ison==5 .and.ihi==1) text4 ='"Sun" free 3D East, FITEX'
       if (ison==5 .and.ihi==2) text4 ='"Sun" free 3D mid., FITEX'
       if (ison==5 .and.ihi==3) text4 ='"Sun" free 3D West, FITEX'
      endif
     endif
     if (irb==2) text4 = '   ref. Mercury orbit (A)'
     if (irb==3) text4 = '   ref. Mercury orbit (B)'
     if (irb==4) text4 = '   ref. Mercury orbit (C)'
     if (irb==5) text4 = '    reference Venus orbit'
    else
     if (ilin==1) text4 = '     all Mercury transits'
     if (ilin==2) text4 = '       all Venus transits'
     if (ilin==3) text4 = 'linear c., Merc. to Earth'
     if (ilin==4) text4 = 'linear c. Mercury to Mars'
    endif
    write(ia,'(/a27,a19,a8,a25)') text1,text2,text3(ika),text4
    if (ipla/=3) then
     if (iek==1)  text0 ='  Ecl. north p/'
     if (iek==2)  text0 ='  Ecl. south p/'
     if (ison>=3 .or.iek==3) text0 ='  Ecl. N and S,'
    else;                    text0 ='  Period   (yea'
    endif
    if (ijd==15 .and.(imod/=2 .or.(imod==2 .and. &
       (iaph==3 .or.iaph==4)))) then
     if (ipla/=3) then
      if (ison<=2) then
       if (ikomb/=1) write(ia,'(a15,'' years'',f10.2, &
        & '' to'',f10.2,a5,''    angular range:'',f8.4,'' deg'')') &
        text0,zmin,zmax,ca(ical),dwi0
       if (ikomb==1) write(ia,'(a15,'' years'',f10.2, &
        & '' to'',f10.2,a5,'', angular r.:'',f6.2,''/'',f6.2, &
        & '' deg'')') text0,zmin,zmax,ca(ical),dwi,dwikomb
      else
       if (ikomb/=1 .and.iaph/=5) then
        write(ia,'(a15,'' years'',f10.2,'' to'',f10.2,a5, &
         &'',     tolerance F <='',f8.4,'' %'')') &
         text0,zmin,zmax,ca(ical),dwi0
       else
        write(ia,'(a15,'' years'',f10.2,'' to'',f10.2,a5, &
         & '', tolerance F <='',f6.2,''/'',f5.2,'' %'')') &
         text0,zmin,zmax,ca(ical),dwi,dwikomb
```

```
          endif
         endif
        else
         if (ilin>=3) then
          if (ikomb==1) write(ia,'(a15,''rs)'',f10.2, &
           & ''  to'',f10.2,a5,'',   angular r.:'',f6.2,''/'',f6.2, &
           & '' deg'')') text0,zmin,zmax,ca(ical),dwi,dwikomb
          if (ikomb/=1) write(ia,'(a15,''rs)'',f10.2,''  to'', &
           & f10.2,a5,3x,''   angular range:'',f8.4,'' deg'')') &
           text0,zmin,zmax,ca(ical),dwi0
         else
          write(ia,'(5x,a15,''rs) from'',f10.2,''  to'',f10.2,a22)') &
           text0,zmin,zmax,text5(ical)
          return
         endif
        endif
       else
        call ephim(1,iaph,ipla,ical,ak,iak,zjde1,zjahr,delt)
        if (ijd>=1 .and.ijd<=14) then
         write(ia,'(a15,''  constellation'',i3,'',   JDE ='', &
          & f15.5,'', year ='',f9.2,a5)')text0,ijd,zjde1,zjahr,ca(ical)
        else
         write(ia,'(a15,20x,''  JDE ='',f15.5,'', year ='',f9.2,a5)') &
          text0,zjde1,zjahr,ca(ical)
        endif
        if (iaph<=2) then
          call jdedate(zjde1,ical,ida,da,dmo)
          call weekday(zjde1,wd)
          k = 1
          if (zjde1>=0.d0 .and.zjde1<2299161.d0 .and.ical==2) k = 2
          if (zjde1>=1356183.d0 .and.zjde1<=5373484.d0) then
            write(ia,'(25x,''date (''',a7,'',TT) ='', &
              & f4.0,a5,i5,'','',i3,2('':'',i2),'','',A10)') &
              cal(k),da(7),dmo,(ida(i),i=3,6),wd
            return
          else
            write(ia,'(24x,''date (''',a7,'',TT) ='', &
              & f4.0,a5,i6,'','',i3,2('':'',i2),'','',A10)') &
              cal(k),da(7),dmo,(ida(i),i=3,6),wd
            return
          endif
        endif
       endif
       if (iaph==3 .or.iaph==4) then
         write(ia,'(''  Special search (interval), step number ='',i6, &
           & '',   step width ='',f7.3,'' hour(s)'')')iamax,24.d0*step
       endif
       if ((iaph==3 .or.iaph==4).and.ijd==15) then
         write(ia,'(''  Consider without printing by tolerance ='', &
           & f8.4)') dwi2
         write(ia,'(''  Print beyond aphelion (per.) by toler. ='', &
           & f8.4)') dwi3
       endif
       end subroutine

       subroutine tabe(iaph,imod,iek,ia,io, &
         ison,ipla,ilin,itran,is12,iop0,iout)
!-----Tabellenkopf-----------------------------------------------------
!     Bei Datumsberechnungen uebernimmt das Unterprogramm
```



```
!     "zwischenzeile" die Tabellenueberschrift.
      implicit double precision (a-h,o-z)
      character(2) :: trs
      if (ilin>=3.) then
        write(ia,*)
        if (io==2 .and.imod/=3) call linie(ia,1)
      endif
      if (ipla==3) then
        trs = 'tr'
        if (itran==2 .or.ison/=5 .or.imod==3) trs = '  '
        if (ilin>=3) then
         if (ison==5) then
          write(ia,'('' co '',a2,''    k         JDE          year'', &
            & ''   dt[days]  Lm-Lv   Lm-Le  Lm-Lma   dLmin'')')trs
         else
          write(ia,'('' co '',a2,''    k         JDE          year'', &
            & ''   dt[days]  Lm-Lv   Lm-Le  Lm-Lma    dL'')')trs
         endif
        endif
      else
       if (ison<=2) then
         if (imod/=3 .and.iek/=3) then
           write(ia,'('' con   k         JDE         year  '', &
             & ''    Lm     Lm-Lv   Lm-Le    del1   del2   F[%]'')')
         else
           write(ia,'('' con   k         JDE          year'', &
             & ''       Lm     Lm-Lv   Lm-Le    del1   del2  P'')')
         endif
       else
        if (ison==3 .or.ison==4) then
         write(ia,'('' con     k      year     Lm     Lm'', &
           & ''-Lv   Lm-Le  x-Sun  y-Sun  z-Sun  dr  P   F[%]'')')
         if (iaph==3 .or.iaph==4) then
          write(ia,'(''      (  ~k       JDE          M'', &
            & ''       no.     "      "      "     "    "     "  )'')')
         endif
        endif
        if (ison==5) then
         if (iaph==3 .or.iaph==4 .or.iout/=3) then
          if (iaph/=5) then
           write(ia,'('' con   k      year    Lm-Lv   Lm'', &
             & ''-Le   e  it  x-Sun  y-Sun  z-Sun   dr  P   F[%]'')')
          else
           write(ia,'('' con   k         JDE          ye'', &
             & ''ar    e  it   x-Sun  y-Sun  z-Sun   dr  P  F[%]'')')
          endif
         else
          if (ipla==1) then
           if (iaph/=5) then
            write(ia,'('' con   k      year      X5   M/1'', &
              & ''0^7  h-Sun  x-Sun  y-Sun  z-Sun    dr  P   F[%]'')')
           else
            write(ia,'('' con   k       year   dt[days]  '', &
              & ''  X5     M/10^7   x-Sun  y-Sun  z-Sun  P   F[%]'')')
           endif
          else
           if (iaph/=5) then
            write(ia,'('' con   k      year      X5   M/1'', &
              & ''0^9   h-Sun  x-Sun  y-Sun  z-Sun   dr  P   F[%]'')')
```

```
          else
           write(ia,'('' con   k        year   dt[days]  '', &
            & ''  X5     M/10^9   x-Sun  y-Sun  z-Sun  P   F[%]'')')
          endif
         endif
        endif
        if (iaph==3 .or.iaph==4) then
         if (iout==3) then
          if (ipla==1) then
           write(ia,'(''    (  JDE       dt[h]   X5   M/'', &
            & ''10^7  h-Sun    "      "      "     "   "    "  )'')')
          else
           write(ia,'(''    (  JDE       dt[h]   X5   M/'', &
            & ''10^9  h-Sun    "      "      "     "   "    "  )'')')
          endif
         else
          write(ia,'(''      (  ~k       JDE          M'', &
           & ''      "  "     "      "      "     "   "    "  )'')')
         endif
        endif
       endif
      endif
     endif
!     (Output zum Vergleich mit den Pyramidenabstaenden)
      if (ilin>=3) then
       if (imod==3) then
        call linie(ia,1)
       else
        call linie(ia,io)
       endif
       if (io==2 .and.imod/=3 .and.is12==0) then
        write(ia,'(''   Lm       Bm       Rm       Lv       Bv     '',&
          & ''  Rv       Le       Be       Re  '')')
        if (ipla==3) write(ia,'(''   Lma      Bma      Rma'')')
        if (ipla/=3) then
        write(ia,'(''    xm       ym      zm         xv      yv     '',&
          & '' zv       xe       ye      ze   '')')
        write(ia,'(''  xv-xm    xe-xm    yv-ym    ye-ym    zv-zm   '',&
          & '' ze-zm             rel. deviation'')')
        endif
        call linie(ia,1)
       endif
      endif
      if (iop0==-803) write(ia,'(/23x,a35/31x,a19)') &
       'calculation of the file "inser-2.t"','--- please wait ---'
      end subroutine

      subroutine elements(ia,ivers,pla)
!-----Ausgabe der Bahnelemente aller Planeten-------------------------
!     im Rahmen der erweiterten Ergebnisausgabe
      use base, only : re
      implicit double precision (a-h,o-z)
      character(3) :: pla(0:9)
      write(ia,'('' pla.   mean long.   a [AU]   '', &
        & ''eccentr.  asc.node    incl.     per.    per.[AU]'')')
      call linie(ia,2)
      do i=1,8
        pd = re(26+6*i) * (1.d0-re(27+6*i))
        if (ivers==3 .and.i==3) then
```



```
         write(ia,'(1x,a3,f13.5,2f10.5,a11,f9.5,f11.5,f10.5)')pla(i),&
           (re(24+6*i+j),j=1,3),'     ---    ',(re(24+6*i+j),j=5,6),pd
        else
         write(ia,'(1x,a3,f13.5,2f10.5,f11.5,f9.5,f11.5,f10.5)') &
           pla(i),(re(24+6*i+j),j=1,6),pd
        endif
      enddo
      end subroutine

      subroutine linie(ia,ib)
!-----Linie, waagerecht-------------------------------------------------
      implicit double precision (a-h,o-z)
      if (ib==1) write(ia,'(1x,79a1)') ('=',i=1,79)
      if (ib==2) write(ia,'(1x,79a1)') ('-',i=1,79)
      if (ib==3) write(ia,'(1x,147a1)') ('=',i=1,147)
      if (ib==4) write(ia,'(1x,147a1)') ('-',i=1,147)
      end subroutine

      subroutine zwizeile(ia,io,zjde,ilin,imod,isep,ical,izp)
!-----Tabellenueberschrift und Zwischenzeile bei Datumsausgaben--------
!     Bei Transitbestimmungen werden abhaengig von der Wahl der
!     Kalender-Option Zwischenzeilen eingefuegt, die den Uebergang
!     von einem zum anderen Kalender kennzeichnen.
      implicit double precision (a-h,o-z)
      ipar = 0; if (isep==4) ipar = 2; is = isep; if (is==2) is = 1
      if (izp==1) then
        if (isep/=4) then
          write(ia,*)
        else
          write(ia,'(92x,''position angles [deg]'',13x, &
            & ''semidiameters ["]'')')
        endif
      endif
      if (izp==1) then
        if (ilin<=2 .and.io==2) call linie(ia,1+ipar)
        if (isep<=2) then
         write(ia,'('' co/p   k        date         time'', &
           & ''    dt[days]  Lm-Lv   Lm-Le  Lm-Lma  sep["]  S'')')
        elseif (isep==3) then
         write(ia,'('' co/p    date/  time:   I        I'', &
           & ''I      nearest     III       IV    sep["]a  S'')')
        else
         write(ia,'('' co/p    date/   time:   I        II     '', &
         &'' nearest     III       IV    sep["] a       P1      P2'',&
         &''      near.    P3      P4         s-Sun    s-pl.    S'')')
        endif
        if (imod/=3 .and.io/=2) then
          call linie(ia,1+ipar)
        else
          call linie(ia,io+ipar)
        endif
        if (io==2 .and.imod/=3) then
          write(ia,'(''   Lm       Bm       Rm       Lv       Bv   '',&
            & ''    Rv       Le       Be       Re  '')')
          write(ia,'(''   Lma      Bma      Rma'')')
          call linie(ia,1+ipar)
        endif
        if (ia==6) then
          izp=2; if (zjde>=0) izp=3; if (zjde>=2299161.d0) izp=4
```

```
        endif
      elseif (zjde>=0.d0 .and.izp==2 .and.ical==2) then
       select case (is)
        case(1); write(ia,'(1x,13(''-''),'' (Jul. cal.) '',53(''-''))')
        case(3); write(ia,'(1x,''----- (Jul. cal.) '',61(''-''))')
        case(4); write(ia,'(1x,''----- (Jul. cal.) '',129(''-''))')
       end select
       if (ia==6) izp = 3
      elseif (zjde>=2299161.d0 .and.izp==3 .and.ical==2) then
       select case (is)
        case(1);write(ia,'(1x,12(''-''),'' (Greg. cal.) '',53(''-''))')
        case(3);write(ia,'(1x,''---- (Greg. cal.) '',61(''-''))')
        case(4);write(ia,'(1x,''---- (Greg. cal.) '',129(''-''))')
       end select
       if (ia==6) izp = 4
      endif
      end subroutine

      subroutine comtime(za,zb,ihour,imin,sec)
!-----Bestimmung der Rechenzeit---------------------------------------
!     Stopzeit zb - Startzeit za = Rechenzeit (CPU-Zeit)
!        [s]            [s]            [hhh:mm:ss.sss]
!     (Falls der CPU-Zaehler waehrend des Programmlaufs sein Maximum
!     erreicht, wird die Zeitangabe inkorrekt. Bei einer 32-bit-Zaehl-
!     variable - gemaess Unixzeit - ist dies am 19.01.2038 der Fall.)
      implicit double precision (a-h,o-z)
      zt  = zb-za
      zih = dint(zt/3600.d0); ihour = idnint(zih)
      zm  = (zt-zih*3600.d0)/60.d0
      zim = dint(zm); imin = idnint(zim)
      sec = (zm-zim)*60.d0
      end subroutine

      subroutine endzeile(ipla,imod,ilin,iaph,isep, &
        ison,ijd,ipos,ia,inum,ihour,imin,sec,is12,iop0)
!-----Endzeilen des Outputs-------------------------------------------
!     Zusammenfassung: Anzahl gefundener Ereignisse, Erklaerung
!     von Zeichen und Ausgabe der CPU-Zeit
      implicit double precision (a-h,o-z)
      dimension :: inum(0:4)
      character(37) :: te1
      character(8)  :: te2
      character(1)  :: te3
      character(29) :: te4
      character(15) :: te5
      te1 = '                                      '
      te2 = 'CPU-time'; te3 = ':'; te5 = ' -- end of run.'
      te4 = '("<" exact deviation dr)     '
      if ((imod/=3 .and.ison>=3).or.imod==3) then
        if (ipla==1) te1 = '(P: polarity, * view from ecl. south)'
        if (ipla==2) te1 = '(P: polarity, resp. view on ecliptic)'
      endif
      if (ilin<=2 .and.isep>=3) &
        te1 = '           ("/" means ascending node)'
      if (ijd==15 .and.iop0/=-803 .and.(imod/=2 .or.(imod==2 .and. &
          (iaph==3 .or.iaph==4 .or.ilin<=2)))) then
        write(ia,500)' Computed constellations:',inum(1),te1
        if (ilin<=2) then
          write(ia,501)' Tested planet. passages:',inum(0)
```



```
          write(ia,501)' Detected transits      :',inum(2)
          write(ia,502)' Centr./grazing transits:',inum(4),' /', &
            inum(3),te2,ihour,te3,imin,te3,sec,te5
        else
          if (ipla/=3) then
            write(ia,503)' Detected constellations:',inum(2),te2, &
              ihour,te3,imin,te3,sec,te5
          else
            if (ison==5) then
              inumber = inum(2)
            else
              write(ia,501)' Detected constellations:',inum(2)
              inumber = inum(3)
            endif
            write(ia,503)' Number of syzygies     :',inumber,te2, &
              ihour,te3,imin,te3,sec,te5
          endif
        endif
      else
        if (ipos==1 .and.is12==0 .and.iop0/=-803) then
          write(ia,504)te4,te2,ihour,te3,imin,te3,sec,te5
        else
          if (iop0==-803) write(ia,'(41x,a38)') &
            'The file "inser-2.t" has been created.'
          write(ia,505)te2,ihour,te3,imin,te3,sec,te5
        endif
      endif
  500 format(1x,a25,i10,6x,a37)
  501 format(1x,a25,i10)
  502 format(1x,a25,i5,a2,i3,7x,a8,i3,a1,i2,a1,f6.3,a15/)
  503 format(1x,a25,i10,7x,a8,i3,a1,i2,a1,f6.3,a15/)
  504 format(14x,a29,a8,i3,a1,i2,a1,f6.3,a15/)
  505 format(43x,a8,i3,a1,i2,a1,f6.3,a15/)
      end subroutine

!h    subroutine histogramm(zz,ihis)  !h
!-----Einsortieren der Genauigkeiten Fpos (zz) in ein Array------------
!     fuer Pyramiden oder Kammern (ipla <= 2, imod <= 2, ison >= 3).
!     Zur Nutzung muessen alle !h-Kommentarzeilen aktiviert werden.
!h    implicit double precision (a-h,o-z)
!h    dimension :: ihis(100)
!h    i = idnint(zz*20.d0 + 0.5d0); if (i<=100) ihis(i) = ihis(i) + 1
!h    end subroutine

      subroutine save_ser
!-----Berechnung des Inhalts der Datei "inserie.t"---------------------
!     Wenn die Datei "inserie.t" mit den julianischen Tagen (JDE)
!     und den Nummern der Transit-Serien neu berechnet werden soll,
!     erfolgt dies mit der Schnellstart-Option -803. Hiermit wird
!     die neue Datei "inser-2.t" erzeugt. Falls gewuenscht kann
!     diese - durch Umbenennung in "inserie.t" - die vorherige bzw.
!     fehlende Datei "inserie.t" ersetzen. Die Verwendung dieser
!     Option ist normalerweise nicht erforderlich.
      use astro, only : ser
      implicit double precision (a-h,o-z)
      open(unit=10,file='inser-2.t')
      write(10,'(9x,a21,a42/6x,a10,a58)')'Julian Ephemeris Day ', &
       'of each first transit in a series (S-No.),','to be used', &
       ' for the years -13000 BC to 17000 AD, VSOP87C full version'
```

```
      write(10,'(34x,a9)')'(Mercury)'
      write(10,'(a14,4(12x,a3))')'S-No.      JDE',('JDE',i=1,4)
      write(10,'(79a1)')('-',i=1,79)
      do i=-150,150,5
        write(10,'(I4,5f15.5)')i,(ser(i+j,1),j=0,4)  ! Serien, Merkur
      enddo
      write(10,'(79a1)')('-',i=1,79)
      write(10,'(35x,a7)')'(Venus)'
      write(10,'(a14,4(12x,a3))')'S-No.      JDE',('JDE',i=1,4)
      write(10,'(79a1)')('-',i=1,79)
      do i=-10,10,5
        write(10,'(I4,5f15.5)')i,(ser(i+j,2),j=0,4)  ! Serien, Venus
      enddo
      ser(19,2) = 1.d12
      write(10,'(I4,4f15.5,e15.1)')i,(ser(15+j,2),j=0,4) ! "     "
      write(10,'(79a1/)')('-',i=1,79)
      close(10)
      end subroutine

      subroutine vsop1tr(ip,rk,tau,del,r3i,eps,inum,resu)
!-----Berechnung der ekliptikalen Koordinaten (Kurzversion VSOP87)-----
!     Beruecksichtigung der Laufzeit des Lichtes, die bei Berechnung
!     der Transitphasen eine Rolle spielt (siehe "vsop2tr")
!     Index ip: 1 = Merkur, 2 = Venus
      use base
      implicit double precision (a-h,o-z)
      dimension :: rk(12),rd(3),inum(0:4)
      del = del/tmil  ! Laufzeit des Lichtes: Merkur/Venus --> Erde
      ist = 3*ip-2; ii = 3*(ip-1)
      do j=ist,ist+2
        call vsop1(j,tau,resu)
        re(j) = resu
      enddo
      call kartko(0)
      do j=ist,ist+2; rk(j) = xyr(j); enddo
      do
        tau1 = tau + del; inum(1) = inum(1) + 1
        do j=7,9
          call vsop1(j,tau1,resu)
          re(j) = resu
        enddo
        call kartko(0)
        do j=7,9
          rk(j) = xyr(j)
        enddo
        do j=1,3
          rd(j) = rk(ii+j) - rk(6+j)
        enddo
        r3i = dsqrt(rd(1)**2 + rd(2)**2 + rd(3)**2)
        del = r3i*AE/(c*86400.d0*tmil); tau2 = tau + del
        if (dabs(tau2-tau1)<eps) exit
      enddo
      del = del*tmil
      end subroutine

      subroutine vsop2tr(xj2,ivers,ip,md, &
        ix,prec,lu,r,rk,ierr,del,r3i,eps,inum,rku)
!-----Aufruf der VSOP87-Subroutine (Vollversion)-----------------------
!     Beruecksichtigung der Laufzeit des Lichtes
```



```
!     Index von rku: 1 = L, 2 = B, 3 = r;  ip: 1 = Merkur, 2 = Venus
!     Input: Zeitpunkt "xj2", Output: Koordinaten der Planeten und
!     Laufzeit des Lichtes "del" vom Planet "ip" zur Erde
      use base, only : re,c,AE
      implicit double precision (a-h,o-z)
      dimension :: rk(12),rd(3),r(6),rku(3),md(0:9),inum(0:4)
      ii = 3*(ip-1)
      call vsop2(xj2,ivers,ip,md,ix,prec,lu,r,ierr,rku)
      do k=1,3
        re(ii+k) = rku(k)
        rk(ii+k) = r(k)
      enddo
      do
        xj3 = xj2 + del
        inum(1) = inum(1) + 1
        call vsop2(xj3,ivers,3,md,ix,prec,lu,r,ierr,rku)
        do k=1,3
          re(6+k) = rku(k)
          rk(6+k) = r(k)
        enddo
        do j=1,3
          rd(j) = rk(ii+j) - rk(6+j)
        enddo
        r3i = dsqrt(rd(1)**2 + rd(2)**2 + rd(3)**2)
        del = r3i*AE/(c*86400.d0)
        xj4 = xj2 + del
        if (dabs(xj4-xj3)<eps) exit
      enddo
      end subroutine

      subroutine fitmin(imod,imodus,iap,ke,x,y,ee1, &
        step,nu,iflag,ddx1,ddx2,test,itin,indx,ix)
!-----Minimum stetiger aber nicht ueberall diff.-barer Funktionen------
!     -->  Resultat = x(indx), indx = 1, 2 oder 3.
!
!     imodus = 1
!     Das Unterprogramm basiert auf einer Art ternaerem Suchen. Es
!     verwendet 3 Stuetzpunkte, um einen neuen Punkt zu finden und
!     einen alten durch diesen zu ersetzen. Dabei ruecken die Punkte
!     immer naeher zusammen, bis die Suchgenauigkeit (ee1) unter-
!     schritten wird. Das Minimum wird durch wiederholten Aufruf
!     von fitmin gefunden. Dieser Such-Algorithmus ist nicht beson-
!     ders schnell, konvergiert aber zuverlaessig und wird u.a. zur
!     Minimierung von "dL" bei Syzygien verwendet.
!
!     imodus = 2 (Spezialsuche)
!     Das Unterprogramm findet den Scheitelpunkt (Minimum) hyper-
!     bolischer Funktionen der Form: y = a * sqrt((x-b)**2 + c**2).
!     Dieser Algorithmus konvergiert deutlich schneller, findet
!     jedoch im konkreten Fall der Planetenbewegung die Loesung nur
!     dann, wenn sie zeitlich nicht zu weit entfernt liegt. Er dient
!     zur schnellen Berechnung der minimalen Separation des Transits.

      implicit double precision (a-h,o-z)
      dimension :: rx(3,4),x(5),y(5),test(10),d(3)
      ie  = 0
      ze  = 0.d0
      ee2 = 1.d-30
      zpa = 5.d0  ! zpa >= 2.d0
```

```
   10 iconv = 0
!c    do iu=ix,6,5; write(iu,'(''  nu,imod,imodus,indx,ddx1,ddx2 ='',&
!c       & i4,3i3,2f13.8)')nu,imod,imodus,indx,ddx1,ddx2
!c      write(iu,'(a12,3f18.8)') '  x(1..3) = ',(x(i),i=1,3)
!c      write(iu,'(a12,3f18.12/)')'  y(1..3) = ',(y(i),i=1,3); enddo
      nulim = 1
!.....Bestimmung der ersten drei x- und y-Werte
      if (iap==5 .and.imod==2) then
        nulim = 2
        if (nu==0) then
          indx = 1; go to 99
        endif
      endif
      if (nu<=nulim) then
        do i=1,2
          x(4-i) = x(3-i)
          y(4-i) = y(3-i)
        enddo
        x(1) = x(1) + step
        indx = 1; go to 99
      endif
      dy1 = y(2)-y(1); dy2 = y(3)-y(2)

! . . Pruefen auf numerisches Rauschen (im Minimum) und Konvergenz-
!     problem. Letzteres Problem entsteht eventuell beim Umschalten
!     von der VSOP87-Kurzversion zur -Vollversion.
      if (dy1>=ze.and.dy2<ze) then
        i1 = 0; if (ddx1+ddx2>1.d-3) i1 = 1
        i2 = 0; if (dabs(dy1)+dabs(dy2)>1.d-3) i2 = 1
!c      if (i1==0.and.i2==0) write(6,*)' -->  num. noise, nu =',nu
!c      if (i2==1) write(6,'(a23,i3)') ' -->  switch-pr.(dy), ',nu
!c      if (i1==1) write(6,'(a23,i3)') ' -->  switch-pr.(dx), ',nu
        if (i1==1 .or.i2==1) then
          iconv = 1; go to 20
        endif
        if (imodus==1) then; ke = 0; return; endif
      endif
   20 if (imodus==1) then
!.......Quasiternaeres Suchen (imodus = 1)
        if (dy1>=ze.and.dy2>=ze.and.iflag==0) then
          do i=1,2
            x(4-i) = x(3-i)                         ! way 1
            y(4-i) = y(3-i)
          enddo
          x(1) = x(1)+x(2)-x(3)
          if (dabs(x(1)-x(4))<1.d-8) then
            y(1) = y(4); go to 10
          endif
          indx = 1
        elseif ((dy1<ze.and.dy2<ze.and.iflag==0).or.iconv==1) then
          do i=1,2
            x(i) = x(1+i)                           ! way 2
            y(i) = y(1+i)
          enddo
          x(3) = x(3)+x(2)-x(1)
          if (dabs(x(3)-x(5))<1.d-8) then
            y(3) = y(5); go to 10
          endif
          indx = 3
```



```
        elseif ((dy1<ze.and.dy2>=ze).or.iflag==1) then
          select case (iflag)
          case(0)                                    ! way 3
            do i=1,2
              x(3+i) = x(2*i-1); y(3+i) = y(2*i-1)
            enddo
            x(3) = (x(3)+(zpa-1.d0)*x(2))/zpa
            indx = 3; iflag = 1
          case(1)
            x(1) = (x(1)+(zpa-1.d0)*x(2))/zpa
            indx = 1; iflag = 0
          end select
        endif
      else
!.......Suche mit hyperbolischem Fit (imodus = 2)
        a1 = x(1)-x(2); a3 = x(3)-x(2)
        b1 = (y(2)**2-y(1)**2)*a3
        b2 = (y(3)**2-y(2)**2)*a1
        if (dabs(b1+b2)<ee2) then; ke = 0; return; endif
        b = 0.5d0*(b1*a3+b2*a1)/(b1+b2) + x(2)
        d(1) = dabs(x(1)-b)
        d(2) = dabs(x(2)-b)
        d(3) = dabs(x(3)-b);         indx = 1
        if (d(2)>d(1).and.d(2)>d(3)) indx = 2
        if (d(3)>d(1).and.d(3)>d(2)) indx = 3
        x(indx) = b
        if (x(1)>x(2)) call pchange(2,1,2,rx,x,y,indx)
        if (x(2)>x(3)) call pchange(2,2,3,rx,x,y,indx)
        if (x(1)>x(2)) call pchange(2,1,2,rx,x,y,indx)
      endif
      ddx1 = dabs(x(2)-x(1))
      ddx2 = dabs(x(3)-x(2))
      ddx3 = dabs(x(3)-x(1))
      if (imodus==2) then
        do i=1,10
          if (dabs(ddx3-test(i))<1.d-7) ie = 1
        enddo
      endif
!.....Hauptbedingung pruefen und Check auf Endlosschleife (ie=1)
      if (ddx1<=ee1.or.ddx2<=ee1.or.ie==1) then
!c      do iu=ix,6,5; write(iu,'(''  nu,imod,imods,indx,dx1,dx2,ie'',&
!c       & '' ='',i4,3i3,2f13.8,i3)') nu,imod,imodus,indx,ddx1,ddx2,ie
!c       write(iu,'(a12,3f18.8/)') '  x(1..3) = ',(x(i),i=1,3); enddo
        ke = 0; return
      endif
      if (imodus==2) then
        itin = itin + 1; if (itin>10) itin = 1
        test(itin) = ddx3
      endif

   99 nu = nu + 1
!c    write(6,'(a11,2i2,3f18.7)')' m,n,x1-3 =',imodus,nu,(x(i),i=1,3)
      if (nu<=100) return
      ke = 2
      do iu=ix,6,5
        write(iu,'(/''  ---->  error in "fitmin", ke ='',I2/)') ke
      enddo
      end subroutine
```

```
      subroutine ringfit(x1,x2,x3,y1,y2,y3,ep,step,nu,itmax,ix,ke)
!-----Nullstellenbestimmung---------------------------------------------
!     Die Routine liefert fuer die Kreisfunktion, die durch (x1,y1),
!     (x2,y2) und (x3,y3) verlaeuft, die naechstgelegene Nullstelle
!     (neuer x2-Wert). Wie bei "sekante" ergibt wiederholtes Aufrufen
!     von "ringfit" die Nullstelle einer stetig differenzierbaren Funk-
!     tion. Abhaengig von den Optionen (ilin<=2, isep=3, 4 bzw. 1) ver-
!     kuerzt sich die Rechenzeit um bis zu 3%, was wenig ist. Da die
!     Grundidee und die Gleichungen jedoch auch eine gewisse Aesthetik
!     besitzen, wurde diese Routine beibehalten. (Der Einsatz von
!     "ringfit" ist nur sinnvoll, wenn die Berechnung der Ausgangs-
!     funktion deutlich mehr Zeit erfordert als "ringfit" selbst.)
      implicit double precision (a-h,o-z)
      if (ke/=5) ke = 1; ep0 = 1.d-20
      if (nu<=1 .or.ke==5) then
        call sekante(x1,x2,y1,y2,ep,step,nu,itmax,ix,ke); return
      endif
      if (nu==2) then  ! Erzeugung des 3. Startpunktes
        x31 = x1; y31 = y1; x32 = x2; y32 = y2
        call sekante(x1,x2,y1,y2,ep,step,nu,itmax,ix,ke)
        if (x1==x31) then; x3 = x32; y3 = y32
                     else; x3 = x31; y3 = y31
        endif; return
      endif
      sh = x2  ! Verschiebung (x2) zum Ursprung
      x1 = x1-sh; x2 = 0.d0; x3 = x3-sh
!c    do iu=ix,6,5; write(iu,'(a16,i3,6f10.6)') &
!c      'nu, x123, y123 =',nu,x1,x2,x3,y1,y2,y3; enddo
      z1 = x1*x1 + y1*y1;  ya = y2-y1;  xa = -x1
      z2 =         y2*y2;  yb = y3-y2;  xb = x3
      z3 = x3*x3 + y3*y3;  yc = y1-y3;  xc = x1-x3
      xy = 0.5d0/(x1*yb + x3*ya)
      if (dabs(xy)<ep0) then
        x1 = x1+sh; x2 = sh; ke = 5; return
      endif
      x0 =  (z1*yb + z2*yc + z3*ya)*xy
      y0 = -(z1*xb + z2*xc + z3*xa)*xy
      wu = x0*x0 + (y2-y0)**2 - y0*y0
      if (wu<0.d0) then; ke = 4; go to 10; endif
      wu = dsqrt(wu)
      xx = x0 + wu; xx2 = x0 - wu  ! (2 Loesungen)
      if (dabs(xx)>dabs(xx2)) xx = xx2
      d1 = dabs(x1-xx); d2 = dabs(xx); d3 = dabs(x3-xx)
      if     (d3>d1.and.d3>d2) then; x3 = 0.d0; y3 = y2
      elseif (d1>d2.and.d1>d3) then; x1 = 0.d0; y1 = y2
      endif
      x1 = x1+sh; x2 = xx+sh; x3 = x3+sh; nu = nu+1
      if (dabs(x2-x1)<ep.or.dabs(x3-x2)<ep) then
!c      do iu=ix,6,5; write(iu,'(a8,7x,a1,i3,3f14.10)') &
!c        'nu, x123','=',nu,x1-sh,x2-sh,x3-sh; enddo
        ke = 0; return
      endif
      if (nu<=itmax) return
      ke = 2
   10 do iu=ix,6,5
        write(iu,'(/''  ---->  error in "ringfit", ke ='',I2/)') ke
      enddo
      end subroutine
```



```
      subroutine sekante(x1,x2,y1,y2,ep,step,nu,itmax,ix,ke)
!-----Nullstellenbestimmung der Sekante---------------------------------
!     Das Programm liefert die Nullstelle der linearen Funktion, die
!     durch (x1,y1) und (x2,y2) verlaeuft. Das Ergebnis wird als
!     neuer x2-Wert ausgegeben. Wiederholtes Aufrufen dieser Routine
!     liefert die Nullstelle (erster Ordnung) einer stetig differen-
!     zierbaren, nicht notwendigerweise linearen Funktion.
      implicit double precision (a-h,o-z)
      if (ke/=5) ke = 1
!c    do iu=ix,6,5; write(iu,'(a16,i3,2f16.6,2f12.6)') &
!c      'nu,x1,x2,y1,y2 =',nu,x1,x2,y1,y2; enddo
      nu = nu + 1
      if (nu<=1) then
        x1 = x2                       !-------------------------------
        y1 = y2                       !  In "sekante", "ringfit",
        x2 = x1 + step                !  "fitmin", and "vsop3" the
        return                        !  error code "ke" means:
      endif                           !
      if (y1==y2) then                !  ke=0:  no error
        ke = 3; go to 10              !  ke=1:  routine runs (internal)
      endif                           !  ke=2:  too many iterations
      x0 = x2-y2*(x2-x1)/(y2-y1)      !  ke=3:  division by zero
      if (dabs(y2)<dabs(y1)) then     !  ke=4:  root of negative number
        x1 = x2                       !  ke=5:  switchover to "sekante"
        y1 = y2                       !-------------------------------
      endif
      x2 = x0
      if (dabs(x2-x1)<ep.and.nu>2) then
!c      do iu=ix,6,5; write(iu,'(a16,i3,2f16.6)') &
!c        'nu,x1,x2       =',nu,x1,x2; enddo
        ke = 0; return
      endif
      if (nu<=itmax) return; ke = 2
   10 do iu=ix,6,5
        write(iu,'(/''  ---->  error in "sekante", ke ='',I2/)') ke
      enddo
      end subroutine

! >>    Update: The 4 subroutines of FITEX have been updated       <<
! >>            for Fortran 95 standard, double precision,         <<
! >>            and free source form.                              <<

!-----------------------------------------------------------------------
!     FITEX                                              M O D I N A 8 7
!-----------------------------------------------------------------------
!
!  PROGRAMM BESCHREIBUNG NR. 320 VON G. W. SCHWEIMER    (VERSION 1985)
!
!  CHISQUARE MINIMISING SUBROUTINE
!  SOLVES THE NONLINEAR LEAST SQUARES PROBLEM
!  USING A LEAST SQUARES INTERPOLATION BETWEEN VARIABLES AND FUNCTIONS
!  OR THE EXACT GRADIENT OF THE FUNCTIONS
!  CALLED SUBROUTINES: LILESQ(LINEAR LEAST SQUARES PROBLEM)
!                      INVATA(INVERSION OF A(TRANSPOSED)*A)
!                      FIT1(ONE DIMENSIONAL MINIMUM SEARCH)
!  CALLING SEQUENCE
!     KE=0
!     M=NUMBER OF FUNCTIONS, M GE N
!     N=NUMBER OF VARIABLES, N GE 1
```

```
!     DO 1 I=1,N
!     X(I)=STARTING VALUES OF THE VARIABLES
!   1 E(I)=ABSOLUTE SEARCH ACCURACIES FOR THE VARIABLES, E(I) NE 0
!     W(1)=FIRST STEP SIZE IN UNITS OF E(I), IF LE 1 W(1) = 100 BY
!          FITEX THE MAXIMUM ALLOWED STEP SIZE IS 2*W(1)
!     W(2)=METHOD OF APPROXIMATION, 0 FOR LEAST SQUARES INTERPOLATION
!                                 1 FOR EXACT GRADIENT OF THE FUNCTIONS
!     IW(1)=NUMBER OF POINTS TO BE REMEMBERED, IF LE N IW(1) = N+1
!     IW(2)=MAXIMUM NUMBER OF FUNCT. EVALUATIONS, IF EQ 0 IW(2)=2IW(1)
!           IF IW(2) LT 0 NO ACTION EXCEPT KE = 0
!     JA=4+MAX0(14,(N*(N+5))/2)+(M+N+1)*(IW(1)+1)
!   2 W(4)=0.
!     DO 3 I=1,M
!     F(I)=FUNCTION VALUES AT THE POINT X
!     IF(W(2)==0.) GO TO 3
!     W(JA+I+M*(J-1))= DF(I)/DX(J) FOR J=1,N
!   3 W(4)=W(4)+F(I)*F(I)
!     OPTIONAL WRITE(*,*) IW(3),IW(4),W(3),W(4),X,F
!     CALL FITEX(KE,M,N,F4,X4,E4,W4,IW)
!     IF(KE==1) GO TO 2
!     W(3)=ERROR RENORMALISATION FACTOR
!     W(4)=MINIMUM QUADRATIC SUM OF THE F(I)
!     X=MINIMUM POINT
!     F=FUNCTIONS AT THE MINIMUM POINT
!     KE=ERROR CODE   KE=0: WITHOUT ERRORS
!                     KE=2: USER INTERRUPT; RETURNS MINIMUM VALUES
!                           WITHOUT ERRORS. THE CURRENT POINT IS
!                           IGNORED. FOR NORMAL USER INTERRUPT SET
!                           IW(2)=IW(3).
!                     KE=3: MAXIMUM NUMBER OF FUNCTION EVALUATIONS
!                     KE=4: ROUNDING ERRORS
!                     KE=5: THE FUNCTIONS DO NOT DEPEND ON X(IW(4))
!                     KE=6: USELESS VARIABLES IN THE PREPARATORY CALLS,
!                           THE LABELS OF THE VARIABLES ARE IW(3),IW(4)
!                     KE=7: M LT N OR N LT 0 OR W(2)*(W(2)-1.) NE 0
!     W(4+I)=STANDARD ERRORS OF THE VARIABLES
!            THE ERROR CALCULATION ASSUMES LINEAR FUNCTIONS.
!            THE PROGRAM SHOWS THE LINEARITY BY THE KIND OF
!            PREDICTION IW(3)
!            IW(3)=0: LINEAR PREDICTION
!                 =1: STEP SIZE LIMITATION
!                 =2: ONE DIMENSIONAL SEARCH
!                 =3: RANDOM SEARCH
!            THE ERRORS ARE CORRECTLY CALCULATED IF THE LAST
!            N ITERATIONS WERE LINEAR, I.E. IW(3)=0.
!     W(4+N+I)=ERROR ENHANCEMENTS
!     W(4+N+N+I+(J*(J-1))/2)=ERROR CORRELATION BETW. X(I) AND X(J) I<J
!     IW(3): NUMBER OF FUNCTION EVALUATIONS
!     IW(4): NUMBER OF DEGREES OF FREEDOM
!  WORKING FIELD:  IW: LENGTH 4+K WITH K = IW(1)
!     W: LENGTH 4+MAX(14,(N*(N+5))/2)+(M+N+1)*(K+1)+M*N
!                  ADRESSES IN IW
!                      4+L: LABELS OF THE QUADRATIC SUMS
!                  ADRESSES IN W
!                      4+I: STANDARD ERROR OF X(I)
!                      4+N+I: ERROR ENHANCEMENT FOR X(I)
!                      FROM 4+N+N+1: MATRIX D AND ERROR CORRELATIONS
!                      FROM JS+1 MATRIX S; JS = 4+MAX0(14,(N*(N+5))/2)
!                      FROM JA+1: MATRIX A WITH JA = JS+(M+N+1)*(K+1)
```



```
!  THE WORKING FIELDS CONTAIN ALL INFORMATION FOR THE CONTINUATION OF
!  THE SEARCH. THIS ALLOWS A SEARCH WITHIN ANOTHER SEARCH JUST CHANGING
!  THE WORKING FIELDS
!
!-------------------------------------------------------------------
      SUBROUTINE FITEX(KE,M,N,F,X,E,W,IW)
      IMPLICIT NONE
      INTEGER(4) :: KE,M,N,I,I1,I2,J,J1,J2,J3,JA,JD,JM,JS,K,KV
! >>  Sizes of IW and W are increased because of index overflow,
! >>  although FITEX ran correctly before. (The numbers 100 and 1000
! >>  are appropriate, if n = 7 and m = 9.)
      INTEGER(4) :: IW(100),L,LM,MF
      REAL(8) :: E(N),F(M),W(1000),X(N),EPS,S,T,U,V,BIG
      REAL(4) :: A
      INTEGER(2) :: IR
! >>  A and IR in the equivalence statement have still the original
! >>  single precision, since they are used to generate random numbers
! >>  and so the calculation is not changed.
      EQUIVALENCE (A,IR)
      DATA EPS/1.D-8/,BIG/7.D+75/
      DATA MF/0/,J/0/,LM/0/,JS/0/,JM/0/,JD/0/,JA/0/,J3/0/ ! pre-init.
      IF (IW(2)<0) GO TO 50
      JD = 4 + N + N
      JS = 4 + MAX0(14,(N*(N+5))/2)
      LM = M + N + 1
      IF (KE/=0) GO TO 2
      IF (IW(1)<=N) IW(1) = N + 1
      IF (IW(2)==0) IW(2) = 2*IW(1)
      IF (W(1)<=1.D0) W(1) = 100.D0
      IW(3) = 1
      K = IW(1)
      DO L = 1,K
        IW(L+4) = 1 + K - L
        W(JS+LM*L) = 7.D75
      ENDDO
      KE = 1
    2 K = IW(1)
      KV = K
      JA = JS + LM* (K+1)
      JM = JS + LM*IW(5) - LM
      J3 = JA - LM
      IF (KE==2) GO TO 52
      IF (M<N.OR.N<1 .OR.W(2)*(W(2)-1.D0)/=0.D0) GO TO 57
      IF (W(4)<=0.D0) GO TO 50
      L = IW(K+4)
      IF (W(JS+LM*L)==BIG) KV = L - 1
      DO I = 1,K
        J1 = JS + LM*IW(I+4)
        IF (W(4)<W(J1)) GO TO 4
      ENDDO
      GO TO 37
    4 IF ((W(2)==0.D0 .AND.I>MAX0(N+1,KV)).OR. &
         (W(2)==1.D0 .AND.I>1)) GO TO 37
      IF (KV<K) KV = KV + 1
      I1 = K + 4
      I2 = K - I
      IF (I2==0) GO TO 6
      DO J = 1,I2
        I1 = I1 - 1
```

```
          IW(I1+1) = IW(I1)
        ENDDO
        IW(I1) = L
        JM = JS + LM*IW(5) - LM
  !  NEW ROW
     6 J1 = JS + LM* (L-1)
        DO I = 1,N
          J1 = J1 + 1
          W(J1) = X(I)
        ENDDO
        DO I = 1,M
          J1 = J1 + 1
          W(J1) = F(I)
        ENDDO
        W(J1+1) = W(4)
  !  TEST MAXIMUM NUMBER OF FUNCTION EVALUATIONS
        IF (IW(3)>=IW(2)) GO TO 53
        IF (N==1) GO TO 42
  !  EXACT GRADIENTS OR END OF PREPARATORY FUNCTION EVALUATIONS
        IF (W(2)==1.D0 .OR.IW(3)>N+1) GO TO 15
  !  PREPARATORY FUNCTION EVALUATIONS
        MF = IW(3)
        IF (MF==1) GO TO 12
        X(MF-1) = W(3)
        J2 = JS + N
        S = 0.D0
        DO I = 1,M
          T = F(I) - W(J2+I)
          S = S + T*T
        ENDDO
        J = 2
        IF (S<EPS*EPS*W(JS+LM)) GO TO 55
        W(3) = S
        J1 = 2 + N + MF
        W(J1) = DSQRT(W(3))
        IF (MF<=2) GO TO 12
        I1 = N + 1
        DO J = 3,MF
          I2 = J2 + LM* (J-2)
          S = 0.D0
          DO I = 1,M
            S = S + (W(I2+I)-W(J2+I))* (F(I)-W(J2+I))
          ENDDO
          IF (DABS(W(J1)*W(I1+J)-DABS(S))<EPS*DABS(S)) GO TO 56
        ENDDO
     12 IF (MF==N+1) GO TO 15
        W(3) = X(MF)
        X(MF) = X(MF) + W(1)*E(MF)
        GO TO 100
  !  END OF PREPARATORY FUNCTION EVALUATIONS
  !  SUM OF INVERSES OF THE QUADRATIC SUMS
     15 S = 0.D0
        DO L = 1,KV
          T = W(JS+LM*L)
          S = S + 1.D0/ (T*T)
        ENDDO
        W(JA) = 1.D0/S
  !  CENTRE OF THE VARIABLES AND FUNCTIONS
        I1 = M + N
```



```
        DO I = 1,I1
          J1 = JS
          S = 0.D0
          DO L = 1,KV
            T = W(J1+LM)
            S = S + W(J1+I)/ (T*T)
            J1 = J1 + LM
          ENDDO
          W(J3+I) = S*W(JA)
        ENDDO
        IF (KE/=1) GO TO 60
        IF (W(2)==0.D0) GO TO 20
        J1 = JA - M - 1
        DO I = 1,M; W(J1+I) = F(I); ENDDO
        GO TO 23
  !  MATRIX A
     20 J1 = JA
        DO I = 1,N
          U = W(J3+I)
          DO J = 1,M
            J1 = J1 + 1
            J2 = JS
            S = 0.D0
            T = W(J3+N+J)
            DO L = 1,KV
              V = W(J2+LM)
              S = S + (W(J2+N+J)-T)* (W(J2+I)-U)/ (V*V)
              J2 = J2 + LM
            ENDDO
            W(J1) = S*W(JA)
          ENDDO
        ENDDO
        IF (KE/=1) GO TO 62
  !  LINEAR LEAST SQUARES PROBLEM
     23 CALL LILESQ(M,N,IR,W(JA+1),W(JA-M),W(5),W(N+5))
        IF (IR<0) GO TO 54
        IF (IR==0) GO TO 24; GO TO 35
  !  MATRIX D
     24 J1 = JD
        DO I = 1,N
          T = W(J3+I)
          DO J = 1,I
            J1 = J1 + 1
            J2 = JS
            S = 0.D0
            U = W(J3+J)
            DO L = 1,KV
              V = W(J2+LM)
              S = S + (W(J2+I)-T)* (W(J2+J)-U)/ (V*V)
              J2 = J2 + LM
            ENDDO
            W(J1) = S*W(JA)
          ENDDO
        ENDDO
  !  NEW VARIABLES
        IF (W(2)==0.D0) GO TO 28
        DO I = 1,N; X(I) = W(JM+I) - W(I+4); ENDDO
        GO TO 31
     28 DO I = 1,N
```

```
            I2 = 1
            J1 = JD + (I*I-I)/2
            S = 0.D0
            DO J = 1,N
              J1 = J1 + I2
              IF (J>=I) I2 = J
              S = S + W(J1)*W(J+4)
            ENDDO
            X(I) = W(J3+I) - S
          ENDDO
!  TEST OF CONVERGENCE
   31 A = 0.E0
          DO I = 1,N
            W(I+4) = X(I) - W(JM+I)
            A = AMAX1(A,SNGL(DABS(W(I+4)/E(I))))
          ENDDO
          IF (A<1.E0) GO TO 50
          IW(4) = 0
          W(3) = 1.D0
          IF (A<2.E0*W(1)) GO TO 33
!  STEP SIZE LIMITATION
          IW(4) = 1
          W(3) = 2.D0*W(1)/A
   33 DO I = 1,N; X(I) = W(JM+I) + W(3)*W(I+4); ENDDO
          GO TO 100
!  RANDOM PREDICTION
   35 DO I = 1,N
            A = SNGL(W(J3+I))
            X(I) = W(JM+I) + W(1)*E(I)* &
                   (MOD(IABS(INT(IR,KIND=4)),200)-100)/100.D0
          ENDDO
          IW(4) = 3
          GO TO 100
!  ONE DIMENSIONAL SEARCH
   37 IF (N==1) GO TO 43
          IF (IW(3)>=IW(2)) GO TO 53
          IF (IW(4)==2) GO TO 39
          IW(4) = 2
          DO I = 1,N; W(J3+I) = X(I) - W(JM+I); ENDDO
          IR = 3
          W(5) = IR
          IR = 20
          W(6) = IR
          W(8) = 0.5D0
          W(11) = 0.D0
          W(12) = 0.D0
          W(13) = 0.D0
          W(14) = 1.D0
          W(16) = W(JM+LM)
          W(17) = W(4)
          GO TO 40
   39 W(9) = W(4)
          CALL FIT1(KE,W(5),W(8))
   40 DO I = 1,N; X(I) = W(JM+I) + W(8)*W(J3+I); ENDDO
          IF (KE==3) KE = 2
          IF (KE==2) GO TO 53
          KE = 1
          W(3) = W(8)
          GO TO 100
```



```
!  ONLY ONE VARIABLE X
   42 IF (IW(3)>1) GO TO 43
          KE = 0
          W(10) = W(1)*E(1)
          W(11) = E(1)
          W(12) = 0.D0
   43 IR = INT(IW(2),KIND=2)
          W(6) = A
          W(8) = X(1)
          W(9) = W(4)
          CALL FIT1(KE,W(5),W(8))
          IW(4) = 2
          X(1) = W(8)
          IF (KE==1) GO TO 100
          IF (KE>0) KE = KE + 1
          W(3) = 0.D0
          W(5) = 0.D0
          IF (W(6)/=0.D0) GO TO 74
          W(5) = DSQRT(DABS((W(13)-W(15))/ ((W(16)-W(17))/ (W(13)-W(14))- &
          (W(17)-W(18))/ (W(14)-W(15)))))
          W(6) = 1.D0
          W(7) = 1.D0
          GO TO 71
!  END OF SEARCH
   50 KE = 0
          IF (W(4)==0.D0 .OR. IW(2)<0) GO TO 100
          GO TO 52
!  ERROR CODE DEFINITION
   57 KE = KE + 1
   56 KE = KE + 1
   55 KE = KE + 1
   54 KE = KE + 1
   53 KE = KE + 2
   52 DO I = 1,N; W(I+4) = 0.D0; ENDDO
          W(3) = 0.D0
          IF (KE*(KE-3)/=0 .OR.(KE==3 .AND.(W(2)==1.D0 .OR. &
             (W(3)==0.D0 .AND.IW(3)<=N)))) GO TO 74
!  COMPUTATION OF THE ERRORS OF THE VARIABLES
!  RESTORE MATRIX G
          IF (W(2)==0.D0) GO TO 15
          J1 = JA
          I1 = N + 1
          DO 45 I = 2,I1
            IF (I>M) GO TO 45
            DO J = I,M; W(J1+J) = 0.D0
            ENDDO
            J1 = J1 + M
   45 ENDDO
          DO 49 I = 1,N
            DO I1 = I,N
              A = SNGL(W(4+N+I1))
              IF (IR==I) EXIT
            ENDDO
            IF (I1==I) GO TO 49
            J1 = JA + M* (I-1)
            J2 = JA + M* (I1-1)
            W(4+N+I1) = W(4+N+I)
            DO J = 1,N
              A = SNGL(W(J1+J))
```

```
          W(J1+J) = W(J2+J)
          W(J2+J) = A
        ENDDO
     49 ENDDO
        GO TO 66
   !  INVERSE OF MATRIX D
     60 T = DSQRT(W(JA))
        J1 = JA
        DO I = 1,N
          S = W(J3+I)
          J2 = JS + I - LM
          DO L = 1,KV
            J1 = J1 + 1
            W(J1) = T*(W(J2+L*LM)-S)/W(JS+L*LM)
          ENDDO
        ENDDO
        CALL INVATA(KV,N,IR,W(JA+1),W(JD+1),X)
        IF (IR==0) GO TO 20
        GO TO 74
   !  MATRIX G = A*INVERSE OF D
     62 DO L = 1,M
          J1 = L + JA - M
          DO I = 1,N
            I1 = JD + (I*I-I)/2
            I2 = 1
            S = 0.D0
            DO J = 1,N
              I1 = I1 + I2
              IF (J>=I) I2 = J
              S = S + W(I1)*W(J1+J*M)
            ENDDO
            X(I) = S
          ENDDO
          DO J = 1,N; W(J1+J*M) = X(J); ENDDO
        ENDDO
   !  DIAGONAL ELEMENTS OF G(T)*G
     66 J1 = JA
        DO I = 1,N
          S = 0.D0
          DO L = 1,M
            J1 = J1 + 1
            S = S + W(J1)*W(J1)
          ENDDO
          W(4+N+I) = DSQRT(S)
        ENDDO
   !  STANDARD ERRORS AND ERROR CORRELATIONS
        CALL INVATA(M,N,IR,W(JA+1),W(JD+1),X)
        IF (IR/=0) GO TO 74
        DO I = 1,N
          W(I+4) = DSQRT(W(JD+ (I*I+I)/2))
          W(4+N+I) = W(I+4)*W(4+N+I)
        ENDDO
        J1 = JD
        DO I = 1,N
          DO J = 1,I
            J1 = J1 + 1
            W(J1) = W(J1)/ (W(I+4)*W(J+4))
          ENDDO
        ENDDO
```



```
   !  ERROR RENORMALISATION FACTOR
     71 S = 0.D0
        DO I = 1,M; S = S + W(JM+N+I); ENDDO
        W(3) = DSQRT(DABS(W(JM+LM)-S*S/M)/MAX0(M-N-1,1))
        DO I = 1,N; W(I+4) = W(I+4)*W(3); ENDDO
   !  RESTORE OPTIMUM VALUES TO X AND F
     74 IW(4) = M - N - 1
        IF ((KE-5)* (KE-6)/=0) GO TO 75
        IW(3) = J - 2
        IW(4) = MF - 1
     75 DO I = 1,N; X(I) = W(JM+I); ENDDO
        DO I = 1,M; F(I) = W(JM+N+I); ENDDO
        W(4) = W(JM+LM)
    100 IF (KE==1) IW(3) = IW(3) + 1
        END SUBROUTINE

   !-----------------------------------------------------------------------
   !    FIT1                                                 M O D I N A 8 7
   !-----------------------------------------------------------------------
   !
   !  PROGRAMM BESCHREIBUNG NR. 309 VON G. W. SCHWEIMER (VERSION 1985)
   !
   !  MINIMISATION OF A FUNCTION F(X) OF ONE VARIABLE X
   !  CALLING SEQUENCE
   !     KE=0
   !     I(2)=MAXIMUM NUMBER OF FUNCTION EVALUATIONS
   !     W(1)=START VALUE OF X
   !     W(3)=FIRST STEP SIZE
   !     W(4)=ABSOLUTE SEARCH ACCURACY
   !     W(5)=RELATIVE SEARCH ACCURACY
   !   1 W(2)=FUNCTION VALUE F(X) AT X=W(1)
   !     OPTIONAL WRITE VI(1),X,F
   !     CALL FIT1(KE,VI,W)
   !     IF(KE==1) GO TO 1
   !     XMIN=W(1)
   !     FMIN=W(2)
   !     NF=VI(1)
   !  KE = ERROR CODE: KE=0 NO ERRORS, KE=
   !   2 MAXIMUM NUMBER OF FUNCTION EVALUATIONS
   !   3 ROUNDING ERRORS, PROB. BECAUSE BOTH W(4) AND W(5) ARE TOO SMALL
   !  THE WORKING FIELDS I AND W HAVE THE LENGTH 3 AND 11 RESPECTIVELY
   !  THEY CONTAIN ALL INFORMATION FOR THE CONTINUATION OF THE SEARCH
   !  THEREFORE A SEARCH WITHIN ANOTHER SEARCH CAN BE DONE JUST CHANGING
   !  THE WORKING FIELDS
   !  IF 2 FUNCTION VALUES F1 AND F2 ARE KNOWN FOR X = X1 AND X2 RESPEC-
   !  TIVELY WITH X1 NE X2 ENTER THE CALLING SEQUENCE AFTER DEFINING :
   !  KE = 1; I(1) = 3; W(6) = X1; W(7) = X2; W(9) = F1; W(10) = F2 AND
   !  W(1) = USERS CHOICE
   !  WORKING FIELD VARIABLES:
   !  I(1): CURRENT NUMBER OF FUNCTION EVALUATIONS
   !  I(2): MAXIMUM NUMBER OF FUNCTION EVALUATIONS
   !  I(3): MINIMUM POINTER, THE MINIMUM FUNCTION VALUE IS AT W(7+I(3))
   !  W(1): CURRENT VALUE OF X
   !  W(2): USER SUPPLIED FUNCTION VALUE
   !  W(3): FIRST STEP SIZE
   !  W(4 AND 5): SEARCH ACCURACIES
   !  W(6, 7 AND 8): X1, X2 AND X3 WITH X1 < X2 < X3
   !  W(9, 10 AND 11): FUNCTION VALUES AT X1, X2 AND X3 RESPECTIVELY
   !
```

```
!-----------------------------------------------------------------------
      SUBROUTINE FIT1(KE,V,W)
      IMPLICIT NONE
      INTEGER(4) :: KE,IV,J,K
      REAL(8) :: V(3),W(11)
      IF (KE==1) GO TO 2
      KE = 1
      V(1) = 1
      V(3) = -1
      W(6) = W(1)
      W(9) = W(2)
    1 W(1) = W(1) + W(3)
      GO TO 12
    2 IF (V(1)>2.D0) GO TO 3
      V(3) = 0.D0
      W(7) = W(1)
      W(10) = W(2)
      IF (W(2)<=W(9)) GO TO 1
      V(3) = -1.D0
      W(1) = W(6) - W(3)
      GO TO 12
    3 IF (V(1)>3.D0) GO TO 5
      W(8) = W(1)
      W(11) = W(2)
      DO 4 J = 1,3
        K = 7 - MOD(J,2)
        IF (W(K)<=W(K+1)) GO TO 4
        W(1) = W(K)
        W(K) = W(K+1)
        W(K+1) = W(1)
        K = K + 3
        W(1) = W(K)
        W(K) = W(K+1)
        W(K+1) = W(1)
    4 ENDDO
      V(3) = 0.D0
      IF (W(9)<W(10).AND.W(9)<W(11)) V(3) = -1.D0
      IF (W(11)<W(10).AND.W(11)<W(9)) V(3) = 1.D0
      GO TO 9
!  SORT IN THE NEW VALUES OF X AND F
    5 IF (V(3)==0.D0) GO TO 6
      J = IDINT(V(3))
      W(7-J) = W(7)
      W(10-J) = W(10)
      IF ((W(7+J)-W(1))*(W(1)-W(7))>0.D0) GO TO 7
      W(7) = W(7+J)
      W(10) = W(10+J)
      W(7+J) = W(1)
      W(10+J) = W(2)
      IF (W(2)>=W(10)) V(3) = 0.D0
      GO TO 9
    6 J = -1
      IF (W(1)<W(7)) J = 1
      IF (W(2)>W(10)) GO TO 8
      W(7+J) = W(7)
      W(10+J) = W(10)
    7 W(7) = W(1)
      W(10) = W(2)
      IV = IDINT(V(3))
```



```
      IF (W(2)<=W(10+IV)) V(3) = 0.D0
      GO TO 9
    8 W(7-J) = W(1)
      W(10-J) = W(2)
    9 IV = IDINT(V(3))
      J = 7 + IV
!  ERROR TESTS
      IF (W(6)==W(7) .OR. W(7)==W(8) .OR. &
         (W(9)==W(10).AND.W(10)==W(11))) GO TO 15
      IF (V(1)>=V(2)) GO TO 16
      IF (V(3)==0.D0) GO TO 10
!  STEP SIZE LIMITATION
      W(1) = W(J) + 2.D0*V(3)* (W(8)-W(6))
      GO TO 12
   10 W(1) = DMIN1(W(8)-W(7),W(7)-W(6))/(W(8)-W(6))
      IF (W(1)>0.1D0) GO TO 11
      W(1) = .5D0* (W(6)+W(8))
      GO TO 12
!  PREDICTION OF THE POSITION OF THE MINIMUM
   11 W(1) = ((W(9)-W(10))/ (W(6)-W(7))- (W(10)-W(11))/ (W(7)-W(8)))/ &
        (W(6)-W(8))
      W(1) = .5D0* (W(6)+W(8)+ (W(11)-W(9))/ (W(1)* (W(6)-W(8))))
!  TEST OF CONVERGENCE
      W(2) = DABS(W(1)-W(J))
      IF (W(2)<DABS(W(4)) .OR. W(2)<DABS(W(5)*W(J))) GO TO 13
   12 V(1) = V(1) + 1.D0
      RETURN
   13 KE = 0
   14 IV = IDINT(V(3))
      W(1) = W(7+IV)
      W(2) = W(10+IV)
      RETURN
   15 KE = KE + 1
   16 KE = KE + 1
      GO TO 14
      END SUBROUTINE

!-----------------------------------------------------------------------
!     INVATA                                               M O D I N A 8 7
!-----------------------------------------------------------------------
!
!  PROGRAMM BESCHREIBUNG NR. 320 VON G. W. SCHWEIMER   (VERSION 1985)
!
!  INVERSION OF THE PRODUCT MATRIX A(TRANSPOSED)*A
!  THE MATRIX A IS REDUCED TO AN UPPER TRIANGULAR MATRIX R BY
!  HOUSEHOLDER TRANSFORMATIONS. THE REMAINING COMPUTATION IS STRAIGHT
!  FORWARD.
!  INPUT VARIABLES: N: NUMBER OF COLUMNS OF MATRIX A
!                   M: NUMBER OF ROWS OF MATRIX A, M >= N > 0
!                   A: INPUT MATRIX (DESTROYED)
!  OUTPUT VARIABLES:   IR: ERROR CODE
!                         IR=-2: M LT N OR N LT 1
!                         IR=-1 RANK OF MATRIX A IS ZERO
!                         IR=0 NO ERROR, RANK OF MATRIX A IS N
!                         IR>0 RANK OF MATRIX A IS IR, THE INVERSE
!                               OF A(T)*A IS COMPUTED CONSIDERING THE
!                               IR COLUMNS OF A INDICATED BY THE FIRST
!                               IR COMPONENTS OF IP
!                  A: TRIANGULAR MATRIX R, R=A(I,J) I<=J=1,N
```

```
!                  D: VECTOR OF LENGTH (N*(N+1))/2, IT CONTAINS THE
!                        UPPER TRIANGULAR PART OF THE INVERSE OF A(T)*A
!                 IP: PERMUTATION VECTOR OF LENGTH N, ITS FIRST IR
!                        COMPONENTS CONTAIN THE LABELS OF THE USEFULL
!                        COLUMNS OF A, THE LAST COMPONENTS CONTAIN
!                        THE LABELS OF THE COLUMNS WHICH ARE LINEAR
!                        COMBINATIONS OF THE FIRST.
!     THE RANK OF THE MATRIX A IS DETECTED COMPARING THE RESULT
!     OF A SUM WITH THE SUM OF ABSOLUTE VALUES.
!     IF SUM OVER I OF T(I) <= EPS * (SUM OF ABS(T(I))) THEN
!     SUM IS SET TO EXACTR ZERO.
!----------------------------------------------------------------------
      SUBROUTINE INVATA(M,N,IR,A,D,VP)
      IMPLICIT NONE
      INTEGER(2) :: IR
      INTEGER(4) :: M,N,I,I1,IJ,J,K,L
!     Size of D changed (see above, FITEX)
      REAL(8) :: A(M,N),D(15*N),VP(N)
      REAL(8) :: EPS,P,Q,R,S,SIG,T,U,V,C
      DATA EPS/1.D-8/
      DATA I1/0/ ! pre-init.
      IR = INT(N,KIND=2)
      IF (M<N.OR.N<1) GO TO 19
      DO I = 1,IR; VP(I) = I; ENDDO
!  HOUSEHOLDER LOOP
      K = 0
    2 K = K + 1
!  PIVOT ELEMENT
    3 C = 0.D0
      DO 4 I = K,M
        IF (DABS(A(I,K))<=C) GO TO 4
        C = DABS(A(I,K))
        I1 = I
    4 ENDDO
      IF (C>0.D0) GO TO 8
      IR = IR - INT(1,KIND=2)
      IF (K>IR) GO TO 13
!  SET UP THE PERMUTATION VECTOR IP AND PERMUTE THE COLUMNS OF MATRIX A
      L = IDINT(VP(K))
      DO J = K,IR; VP(J) = VP(J+1); ENDDO
      VP(IR+1) = L
      DO I = 1,M
        C = A(I,K)
        DO J = K,IR; A(I,J) = A(I,J+1); ENDDO
        A(I,IR+1) = C
      ENDDO
      GO TO 3
!  ROTATION OF THE LOWER COLUMN FRAGMENTS OF A(K)
    8 DO J = K,IR
        C = A(K,J)
        A(K,J) = A(I1,J)
        A(I1,J) = C
      ENDDO
      S = A(K,K)
      V = 0.D0
      DO I = K,M
        U = A(I,K)/S
        V = V + U*U
      ENDDO
```



```
      V = 1.D0/DSQRT(V)
      SIG = S/V
      U = S + SIG
      A(K,K) = -SIG
      IF (K>=IR) GO TO 13
      L = K + 1
      DO J = L,IR
        S = V*A(K,J)
        P = DABS(S)
        DO I = L,M
          R = (A(I,K)/SIG)*A(I,J)
          S = S + R
          P = P + DABS(R)
        ENDDO
        IF (DABS(S)<=EPS*P) S = 0.D0
        T = (A(K,J)+S)/U
        IF (DABS(T)<=EPS*DABS(S/U)) T = 0.D0
        A(K,J) = -S
        DO I = L,M
          Q = A(I,J)
          P = T*A(I,K)
          R = Q - P
          IF (DABS(R)<=EPS*DABS(P)) R = 0.D0
          A(I,J) = R
        ENDDO
      ENDDO
      GO TO 2
!  END OF HOUSEHOLDER LOOP
   13 IF (IR==0) GO TO 20
!  INVERSE OF THE TRIANGULAR MATRIX R STORED IN D
      IJ = 0
      DO 16 K = 1,IR
        D(IJ+K) = 1.D0/A(K,K)
        IF (K==1) GO TO 16
        I = K
        DO L = 2,K
          I1 = I
          I = I - 1
          S = 0.D0
          DO J = I1,K; S = S + A(I,J)*D(IJ+J); ENDDO
          D(IJ+I) = -S/A(I,I)
        ENDDO
        IJ = IJ + K
   16 ENDDO
!  INVERSE OF THE PRODUCT MATRIX
      IJ = 0
      DO J = 1,IR
        DO I = 1,J
          IJ = IJ + 1
          I1 = IJ
          L = J - I
          S = 0.D0
          DO K = J,IR
            S = S + D(I1)*D(I1+L)
            I1 = I1 + K
          ENDDO
          D(IJ) = S
        ENDDO
      ENDDO
```

```
      GO TO 20
   19 IR = -2
   20 IF (IR==0) IR = -1
      IF (IR==N) IR = 0
      END SUBROUTINE

!---------------------------------------------------------------------
!     LILESQ                                         M O D I N A 8 7
!---------------------------------------------------------------------
!
!  PROGRAMM BESCHREIBUNG NR. 320 VON G. W. SCHWEIMER   (VERSION 1985)
!
!  LINEAR LEAST SQUARES PROBLEM !!B-A*X!!=MIN(X)
!  SOLVED BY HOUSEHOLDER TRANSFORMATIONS
!  REDUNDANT VARIABLES ARE DETECTED BY THE METHOD OF G.GOLUB,
!  NUMERISCHE MATHEMATIK, VOL. 7, PAGE 206-216, (1965)
!  INPUT VARIABLES:M: NUMBER OF ROWS OF A AND B
!                  N: NUMBER OF COLUMNS OF A AND ROWS OF X
!                  A: M*N MATRIX (DESTROYED)
!                  B: VECTOR OF M COMPONENTS (DESTROYED)
!  OUTPUT VARIABLES: X: VECTOR OF VARIABLES, THE REDUNDANT VARIABLES
!                       ARE SET TO ZERO. THE !!X!!=MIN IS NOT USED
!                       BECAUSE THE COMPONENTS OF X ARE ASSUMED TO BE
!                       NOT COMMENSURABLE
!                  IP: PERMUTATION VECTOR OF N COMPONENTS, IT CONTAINS
!                       THE COLUMN LABLES OF MATRIX A ORDERED ACCORDING
!                       THEIR IMPORTANCE IN REDUCING THE EUCLIDEAN NORM
!                  A: THE UPPER PART CONTAINS THE TRANSFORMED INPUT A
!                       A(2,1) CONTAINS THE SQUARE OF THE EUCLIDEAN
!                       NORM
!                  B: TRANSFORMED INPUT B
!                  IER: ERROR CODE
!                       IER=0 NO ERROR
!                       IER=-1 ALL COMPONENTS OF X ARE ZERO AND MAY BE
!                       REDUNDANT
!                       IER=-2 NO ACTION BECAUSE M < N OR N < 1
!                       IER>0 THE FIRST IER COMPONENTS OF IP CONTAIN
!                       THE LABELS OF THE NONZERO COMPONENTS OF X, THE
!                       REMAINING COMPONENTS OF X ARE ZERO AND MAY BE
!                       REDUNDANT
!  NOTE: ALL ARITHMETIC OPERATIONS ARE PERFORMED IN DOUBLE PRECISION,
!  AN ITERATIVE IMPROVEMENT IS IMPOSSIBLE WITHOUT SAVING A AND B.
!  THE ROUND OFF ERROR OF !!B-A*X!!**2 IS APPROXIMATLY GIVEN BY
!  !!B(INITIAL)!!**2 - !!B(TRANFORMED)!!**2
!---------------------------------------------------------------------
      SUBROUTINE LILESQ(M,N,IER,A,B,X,VP)
      IMPLICIT NONE
      INTEGER(2) :: IER
      INTEGER(4) :: M,N,I,IP,J,K,L,L1,L2
      REAL(8) :: C,DELTA,EPS,P,Q,R,S,SIG,T,U,V,W
      REAL(8) :: A(M,N),B(M),VP(N),X(N)
      DATA EPS/1.D-8/
      DATA W/0.d0/,SIG/0.d0/,L2/0/,L1/0/,L/0/ ! pre-init.
      IER = 0
      IF (M<N.OR.N<1) GO TO 19
      DO J = 1,N; VP(J) = J
      ENDDO
!  ROTATION LOOP
      DO 10 K = 1,N
```



```
!  PIVOT ELEMENT
        U = 0.D0
        DO 4 J = K,N
          C = 0.D0
          DO 2 I = K,M
            IF (DABS(A(I,J))<=DABS(C)) GO TO 2
            L2 = I
            C = A(I,J)
    2     ENDDO
          IF (C==0.D0) GO TO 4
          S = 0.D0
          T = 0.D0
          DO I = K,M
            V = A(I,J)/C
            S = S + V*V
            T = T + V*B(I)
          ENDDO
          IF (U>=T* (T/S)) GO TO 4
          U = T* (T/S)
          SIG = C*DSQRT(S)
          W = T
          L = J
          L1 = L2
    4   ENDDO
        IF (U==0.D0) GO TO 11
!  PERMUTE A(K) AND B(K)
        I = IDINT(VP(L))
        VP(L) = VP(K)
        VP(K) = I
        DO I = 1,M
          C = A(I,L)
          A(I,L) = A(I,K)
          A(I,K) = C
        ENDDO
        C = B(K)
        B(K) = B(L1)
        B(L1) = C
        DO J = K,N
          C = A(K,J)
          A(K,J) = A(L1,J)
          A(L1,J) = C
        ENDDO
!  ROTATION OF THE LOWER COLUMN FRAGMENT OF A(K) AND B(K)
        U = SIG + A(K,K)
        V = A(K,K)/SIG
        DELTA = (B(K)+V*W)/U
        A(K,K) = -SIG
        B(K) = -V*W
        L = K + 1
        IF (L>M) GO TO 10
        IF (K>=N) GO TO 8
        DO J = L,N
          S = V*A(K,J)
          P = DABS(S)
          DO I = L,M
            R = A(I,K)/SIG*A(I,J)
            S = S + R
            P = P + DABS(R)
          ENDDO
```

```
            IF (DABS(S)<=EPS*P) S = 0.D0
            T = (A(K,J)+S)/U
            IF (DABS(T)<=EPS*DABS(S/U)) T = 0.D0
            A(K,J) = -S
            DO I = L,M
              Q = A(I,J)
              P = T*A(I,K)
              R = Q - P
              IF (DABS(R)<=EPS*DABS(P)) R = 0.D0
              A(I,J) = R
            ENDDO
          ENDDO
     8    DO I = L,M; B(I) = B(I) - DELTA*A(I,K); ENDDO
    10 ENDDO
!   END OF ROTATION LOOP
       K = N
       GO TO 12
    11 K = K - 1
       IER = int(K,KIND=2)
!   SQUARE OF THE EUCLIDEAN NORM
    12 S = 0.D0
       L = K + 1
       IF (K==M) GO TO 14
       DO I = L,M; S = S + B(I)*B(I); ENDDO
    14 A(2,1) = S
       IF (K==N) GO TO 16
!   COMPONENTS OF X WHICH DO NOT REDUCE THE EUCLIDEAN NORM
       DO I = L,N
         DO J = L,N
           IP = IDINT(VP(J))
           X(IP) = 0.D0
         ENDDO
       ENDDO
       IF (K==0) GO TO 20
!   COMPUTATION OF X
    16 IP = IDINT(VP(K))
       X(IP) = B(K)/A(K,K)
       IF (K==1) GO TO 21
       DO J = 2,K
         L = K + 2 - J
         S = B(L-1)
         DO I = L,K
           IP = IDINT(VP(I))
           S = S - A(L-1,I)*X(IP)
         ENDDO
         IP = IDINT(VP(L-1))
         X(IP) = S/A(L-1,L-1)
       ENDDO
       GO TO 21
!   ERROR CODE
    19 IER = IER - INT(1,KIND=2)
    20 IER = IER - INT(1,KIND=2)
    21 RETURN
       END SUBROUTINE
!   Number of lines: 6191
```

Use of P4 program

Concerning the copyrights of H. Jelitto, the executable P4 program with all of its supplemental program, text, and data files, listed in Table 1 – except this manual “p4-manual-06-2015.pdf” (licenced under “CC” BY-NC-SA 4.0; see beginning of this manual) – can be used freely for private, scientific, and educational purposes, but may not be used for any commercial purpose. In case of use for any publication including any sort of presentation, appropriate quotation of the author(s) must be given. For the other program parts (see below), it has to be checked whether permission from the copyright owners is necessary. For any kind of commercial use, a written permission from the author is required.

This program is distributed in the hope that it will be useful, but without any kind of warranty!

Further Copyrights

The following statements concerning other authors
– without guaranty of completeness and correctness –
apply to any use of the P4 computer program, of the
previous P3 version, and of all associated files in Table 1.

Subroutine VSOP87 and associated data files (based on the theory “Variations Séculaires des Orbites Planétaires,” VSOP87): P. Bretagnon and G. Francou, Institut de mécanique céleste et de calcul des éphémérides (IMCCE), 77 Avenue Denfert-Rochereau, F-75014 Paris, France.

Program package FITEX (consisting of four subroutines at the end of the source code of P4): KIT, Karlsruhe Institute of Technology (before: FZK, Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft), Institut für Kernphysik, Postfach 3640, D-76021 Karlsruhe. FITEX was developed by G. W. Schweimer around 1972 and published by H. J. Gils: The Karlsruhe Code MODINA for Model Independent Analysis of Elastic Scattering of Spinless Particles; KfK 3063 (1980) and KfK 3063, 1. Supplement (1983) Kernforschungszentrum Karlsruhe (KfK), Zyklotron Laboratorium. (See also Wissenschaftliche Berichte of the KIT-Campus Nord.)

Subroutine DELTA_T and numbered equations in “Universal Time” section (conversion of terrestrial time TT to universal time UT): The subroutine is based on polynomials up to 7^{th} degree, created by Fred Espenak and Jean Meeus, and published on the “NASA Eclipse Web Site”, Polynomial Expressions for Delta-T.

P4, P3 Programs and all remaining program parts, data files, text, and figures (according to Table 1, including the changes in the VSOP87-subroutine → VSOP87X, VSOP87Y): Hans Jelitto, Ewaldsweg 12, D-20537 Hamburg, Germany.

Comment: *Concerning the further copyrights, it seems that there is no problem for the use if it is a nonprofit use and if appropriate quotation is given to the authors and copyright owners, respectively. Nevertheless, the correct use and acceptance of copyrights is solely the responsibility of the user.*

Acknowledgement

The subroutine **“JDEDATUM”** (to transform the Julian Ephemeris Day into a calendar date) was created on the basis of an algorithm from the very useful book of Jean Meeus: “Astronomical Algorithms,” p. 63 ff. (1991) Willmann-Bell, Inc., P. O. Box 35025, Richmond, Virginia 23235, USA. Additionally, the book “Transits” from J. Meeus (same publisher) was rather valuable for developing and testing the transit computations. Special thanks go to Dipl. Ing. Manfred Geerken from TUHH for valuable help concerning the used computer hardware and software.

References

[1] Bretagnon, P.: Théorie du mouvement de l'ensemble des planétes – Solutions VSOP82. Astronomy and Astrophysics **114** (1982) 278 – 288, ADS (→ pdf)

[2] Bretagnon, P., Francou, G.: Planetary theories in rectangular and spherical variables – VSOP87 solutions. Astronomy and Astrophysics **202** (1988) 309 – 315, ADS (→ pdf)

[3] Jelitto, H.: Geometrie und Anordnung der drei großen Pyramiden von Giza – Teil I: Die Cheops-Pyramide. Grenzgebiete der Wissenschaft, Resch Verlag, Innsbruck, **GW 44/1** (1995) 3 – 28, pdf

[4] Jelitto, H.: Geometrie und Anordnung der drei großen Pyramiden von Giza – Teil II: Chefren- und Mykerinos-Pyramide sowie Gesamtbild, Grenzgebiete der Wissenschaft, Resch Verlag, Innsbruck, **GW 44/2** (1995) 99 – 120, pdf

[5] Jelitto, H.: Pyramiden und Planeten – Ein vermeintlicher Meßfehler und ein neues Gesamtbild der Pyramiden von Giza. Wissenschaft & Technik Verlag, Berlin (1999) more info

[6] Petrie, William Matthew Flinders: The Pyramids and Temples of Gizeh. Field & Tuer, Simpkin, Marshall & Co., Hamilton, Adams & Co., London; Scribner & Welford, New York, first edition (1883) Cornell University (→ pdf)

[6a] Petrie, William Matthew Flinders: The Pyramids and Temples of Gizeh. (as above) New and revised edition (1883) (This edition is partly abridged.) pdf

[7] République Arabe d'Egypte (maps of Egypt), échelle/ map scale 1:5000, Ministère de l'Habitat et de la Re-construction: (for example) Le Caire **F–17**, **F–18** (Pyramids of Giza), or Le Caire **H–23** (Pyramid at Saqqara). Dresse par CONSORTIUM S.F.S./I.G.N. – FRANCE en 1978, Photographies aériennes 1:5000 de Avril 1977, Stéréopréparation appuyée sur le canevas géodésique du SURVEY OF EGYPT (available at Cairo, Zamalek Business Center, home)

[8] Egypt (maps based on the Cadastral Survey), scale 1:25 000: (for example) Portions of Sheets **79/630 and 79/645** (Helwan, pyramids at Abusir) or **78/630 and 78/645** (pyramids at Dashur), (available at Cairo, Zamalek Business Center)

[9] Maragioglio, Vito and Rinaldi, Celeste: L'Architettura delle Piramidi Menfite. Tipografia Canessa, Rapallo (1965), used maps: **Part IV**, **map 1**: Necropolis of Giza (overview). It follows the interior of Cheops Pyramid: **map 3**: overview of inner construction, **map 4**: subterranean chamber, **map 5**: ascending corridor, well shaft, **map 6**: grand gallery, Queen's chamber, **map 7**: King's chamber, openlib

[10] Helck, Wolfgang and Otto, Eberhard: Lexikon der Ägyptologie. Editor: Helck, W., Westendorf, W., publisher: Otto Harrassowitz, Wiesbaden, Germany (1982)

[11] Wölfli, Willy: Archäologie mit einem Schwerionenbeschleuniger. Physik in unserer Zeit, 25. Jahrgang (1994) Nr. 2, 58

[12] Haas, H., Devine, J., Wenke, R., Lehner, M., Woelfli, W., Bonani, G.: Radiocarbon Chronology and the Historical Calendar in Egypt. Chronologies in the Near East, BAR International Series **379 ii** (1987) 585

[13] Jelitto, H.: Book 2 (in preparation)

[14] Stadelmann, R.: Die ägyptischen Pyramiden – Vom Ziegelbau zum Weltwunder. Verlag Philipp von Zabern, Mainz/Rhein, in Zusammenarbeit mit der Wissenschaftlichen Buchgesellschaft, Darmstadt, 1. Auflage (1985)

[15] Schweimer, G. W.: Internal report of the Kernforschungszentrums Karlsruhe, Zyklotron-Laboratorium (1973, 1985) (unpublished)

[16] Gils, H. J.: The Karlsruhe Code MODINA for Model Independent Analysis of Elastic Scattering of Spinless Particles., Kernforschungszentrum Karlsruhe (KfK, heute KIT), Zyklotron Laboratorium, **KfK 3063** (1980) pdf, **KfK 3063 – 1. Supplement** (1983) (includes FITEX) pdf

[17] Meeus, J.: Astronomical Algorithms. First Engl. Edition, Willmann-Bell, Inc., Richmond, Virginia (1991) W-Bell

[18] Haney, M. A.: Skyglobe 4.0 (computer program), "planetarium" simulation of the sky. KlassM Software, P. O. Box 1067, Ann Arbor, MI 48106, USA, © Mark A. Haney, program

[19] Bauval, R., Gilbert, A.: Das Geheimnis des Orion. List Verlag, München, Leipzig (1994)

[20] Bastian, U., Röser, S.: PPM Star Catalogue 1– 4 (Positions and Proper Motions of 181 731 stars north of -2,5 degrees declination for equinox and epoch J2000.0). Spektrum, Akademischer Verlag, Heidelberg, Berlin, New York (1991, 1993)

[21] Röser, S., Bastian, U.: The final PPM Star Catalogue for both Hemispheres. Bull. Inform. CDS **42** (1993) 11, complete catalogue: NASA – (For star coordinates, see also: SIMBAD.)

[22] Meeus, J.: Transits. Willmann-Bell, Inc., Richmond, Virginia (1989) W-Bell

[23] Espenak, F., Meeus, J.: Polynomial Expressions for Delta-T (adapted from "Five Millennium Canon of Solar Eclipses"). NASA Eclipse Web Site, GSFC, Solar System Exploration Division (2004) NASA

[24] Espenak, F., Meeus, J.: Uncertainty in Delta-T (adapted from "Five Millennium Canon of Solar Eclipses"). NASA Eclipse Web Site, GSFC, Solar System Exploration Division (2004) NASA

[25] Smyth, Charles Piazzi: Life and Work at the Great Pyramid. Edmonston and Douglas, Edinburgh, **Vol. I – III** (1867) e-books: vol. I, vol. II, vol. III

[26] Smyth, Charles Piazzi: The Great Pyramid in Egypt. Aus: Astronomical Observations Made at the Royal Observatory, **Vol. XIII**, for the years 1860 – 1869, Neill & Co., Edinburgh (1871)

[27] Edgar, John and Morton: The Great Pyramid – Passages and Chambers. Bone and Hulley, Glasgow, **Vol. I, II** (1910, 1913) archives (pdf): vol. I, vol. II

[28] Vyse, Colonel Howard, and Perring, John Shae: Operations carried on at the Pyramids of Gizeh in 1837. James Fraser, London, **Vol. I – III** (1840 – 1842) e-books: vol. I, vol. II, vol. III

[29] Perring, John Shae: The Pyramids of Gizeh. **Vol. I – III**, Ed. Andrews, E. J., London (1939 – 1942) Universität Heidelberg: vol. I, vol. II, vol. III (→ pdf). Perrings drawings are published also in the appendix of the 3rd volume of "Operations carried on at the Pyramids of Gizeh in 1837" from H. Vyse.

[30] Fienga, A., Simon, J.-L.: Analytical and numerical studies of asteroid perturbations on solar system planet dynamics. Astronomy and Astrophysics **429** (2005) 361 – 367, pdf

[31] Standish, E. M.: Orientation of the JPL Ephemerides, DE200/LE200, to the Dynamical Equinox of J2000. Astronomy and Astrophysics **114** (1982) 297 – 302, ADS (→ pdf)

[32] Standish, E. M.: The Observational Basis for JPL's DE200, the planetary ephemeris of the Astronomical Almanac. Astronomy and Astrophysics **233** (1990) 252 – 271

[33] Standish, E. M., et al.: JPL Planetary and Lunar Ephemerides on CD-ROM, available via Internet from the Willmann-Bell, Inc.-Website, W-Bell

[34] Vogt, H.-H.: Abriß der Astronomie. BI Wissenschaftsverlag, Mannheim, Wien, Zürich (1988)

[35] Brown, T. M., Christensen-Alsgaard, J.: Accurate Determination of the Solar Photospheric Radius, The Astrophysical Journal **500**: L195 – L198 (1998)

[36] Dünkel, A.: Java-Aplett zur Visualisierung der Erdorientierung. Studienarbeit (project work), Universität Stuttgart, Geodätisches Institut (Juni 2009) pdf

[37] Coppola, V., Seago, J. H., Vallado, D. A.: The IAU 2000A and IAU 2006 Precession-Nutation Theories and their Implementation. Spacefligth Mechanics 2009, Volume 134, Advances in the Astronautical Science, Proceedings of the 19th AAS/AIAA Space Flight Mechanics Meeting, Febr. 8-12, 2009, Savannah, Georgia. **AAS 09-159** (2009) 919 ff. pdf

[38] Wittmann, A. D.: On the variation of the obliquity of the ecliptic. Universitäts-Sternwarte Göttingen, **MitAG 62** (1984) 201 – 204, home (→ "Downloads" → pdf)

[39] Danjon, A.: Astronomie Général – Astronomie Sphérique et Éléments de Mécanique Céleste. Seconde Edition, Éditeurs: J. & R. Sennac, Paris (1959)

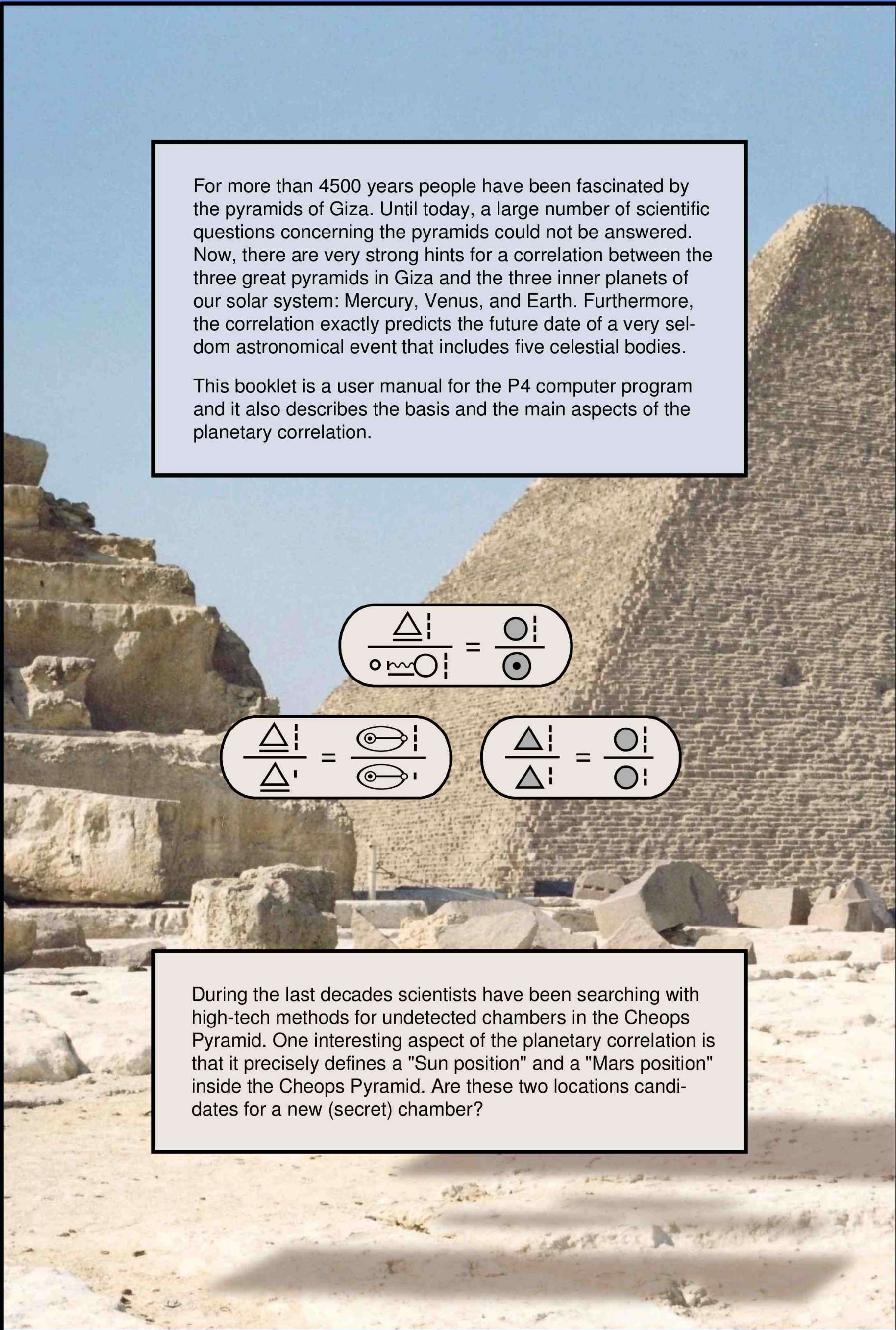
[40] Morrison, L. and Stephenson, F. R., "Historical Values of the Earth's Clock Error DT and the Calculation of Eclipses", J. Hist. Astron. **35**, **Part 3**, August 2004, **No. 120** (2004) 327 – 336

[41] Stephenson F. R. and Houlden M. A.: *Atlas of Historical Eclipse Maps.* Cambridge Univ. Press., Cambridge (1986)

[42] Jelitto, H.: Gespiegelte Planeten – Die Anordnung der Pyramiden von Gizeh. Argo-Verlag, Marktoberdorf, Magazin 2000plus **Spezial 6/156** (2000) 12 – 22, TUHH (→ pdf)

[43] Jelitto, H.: Große Pyramide – Der asymmetrische Kristall. Argo-Verlag, Marktoberdorf, Magazin 2000plus **Spezial 27/221** (2006) 6 – 16, TUHH (→ pdf)

[44] Using GNU Fortran – For gcc version 4.7.2 (GCC), published by the Free Software Foundation, Boston, MA, USA (2012) pdf



For more than 4500 years people have been fascinated by the pyramids of Giza. Until today, a large number of scientific questions concerning the pyramids could not be answered. Now, there are very strong hints for a correlation between the three great pyramids in Giza and the three inner planets of our solar system: Mercury, Venus, and Earth. Furthermore, the correlation exactly predicts the future date of a very seldom astronomical event that includes five celestial bodies.

This booklet is a user manual for the P4 computer program and it also describes the basis and the main aspects of the planetary correlation.

During the last decades scientists have been searching with high-tech methods for undetected chambers in the Cheops Pyramid. One interesting aspect of the planetary correlation is that it precisely defines a "Sun position" and a "Mars position" inside the Cheops Pyramid. Are these two locations candidates for a new (secret) chamber?