PROJECT ADMINISTRATION DATA SHEET

Project No. A-3684

Project Director: Edward J. Shanahan

Sponsor: Army - HQ FORSCOM, Ft. McPherson, GA

Type Agreement: Delivery Order No. 1E03 under Contract F33657-82-G-2083*

Award Period:
- From 9/28/83 to 9/27/84 (Performance)
- 9/27/84 (Reports)

Sponsor Amount:
- Estimated: $247,600
- Funded: $247,600

Cost Sharing Amount: $ Cost Sharing No:

Title: MICROFIX Training and Application Software Development

ADMINISTRATIVE DATA

1) Sponsor Technical Contact:
   Mr. Charles R. France
   HQ FORSCOM
   Patton Hall - DCSI
   Ft. McPherson, GA 30330

2) Sponsor Admin/Contractual Matters:
   Mr. Staten Corbett
   Contracting Division
   Building 184
   Ft. McPherson, GA 30330
   752-2469

OCA Contact Brian J. Lindberg X4820

Military Security Classification: N/A
(or) Company/Industrial Proprietary: N/A

RESTRICIONS

See Attached Government Supplemental Information Sheet for Additional Requirements

Travel: Foreign travel must have prior approval — Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with Gov't except that items costing $1,000 or less vest with sponsor providing prior written approval to purchase received from Contracting Officer.

COMMENTS:

* Basic Ordering Agreement No. F33657-82-G-2083 on file with Lab Director's Office/ECSL.

Note: Funds listed for each task in Delivery Order is maximum expenditure for that task.
Date: October 2, 1985

Project No.: A-3684

Includes Subproject No.(s): N/A

Project Director(s): N.D. Ellingson

Sponsor: Army HQ, Forscom; Ft. McPherson, Ga.

Title: MICROFIX Training and Application Software Development

Effective Completion Date: 5/31/85

Grant/Contract Closeout Actions Remaining:

- [ ] None
- [X] Final Invoice or Final Fiscal Report
- [X] Closing Documents
- [X] Final Report of Inventions - Patent Questionnaire Sent to Project Director
- [X] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other

continues Project No. ____________________________ Continued by Project No. __________________________

copies to:

- Project Director
- Research Administrative Network
- Research Property Management
- Accounting
- Procurement/GTRI Supply Services
- Research Security Services
- Contracts Coordinator (OCA)
- Legal Services
- Library
- GTRC
- Research Communications (2)
- Project File
- Other: Heyser, Jones
PLAN OF INSTRUCTION FOR MICROFIX TRAINING COURSE
FOR TOPOGRAPHIC PERSONNEL
FT. BELVOIR, VIRGINIA

December 12 - 16, 1983

Prepared by

Engineering Experiment Station
Georgia Institute of Technology

for

Assistant Chief of Staff, Intelligence
U.S. Army Forces Command
Ft. McPherson, Georgia

and

Belvoir R&D Center
Ft. Belvoir, Virginia

November 30, 1983
1.0 PURPOSE

This Plan of Instruction is for a one-week short course covering the MICROFIX System I hardware and software capabilities. All aspects of the system will be presented, nevertheless, the major emphasis will be on those capabilities designed to assist the terrain analyst. The goal is twofold: first, to provide the users with the understanding and ability to utilize the system; and second, to elicit feedback from participants to aid in future system modification and development.
2.0 BACKGROUND

MICROFIX is a Quick Reaction Capability (QRC), established by the QRC #55 Management Action Plan, dated January 26, 1983, which is being developed by the Army Intelligence Community to automate tactical intelligence related functions prior to the fielding of ASAS. User validated information derived from MICROFIX will be fed to the Joint Tactical Fusion Program (JTFP) to assist in the development of ASAS.

2.1 Design Philosophy

The overall design philosophy is designed to force a closer interaction between the operational and technical aspects of system development. This approach was established after lengthy examination of the traditional top-down system development method which assumes complete understanding of all requirements at the initiation of the project. In reality, requirement statements are excellent guidelines for system development but are not in sufficient detail to support coding of software. Needed was a bottom-up evolutionary approach which employs a technique to establish a mature capability resulting from constant user/developer interaction. The MICROFIX program uses the "use-learn-develop" approach by establishing a software capability in its infant form; by obtaining feedback from analysts from operational field units who have used the system; and by rewriting the software to meet the expressed needs of the analysts. This cycle is repeated until the mature capability is established. Two groups learn by using this technique: the software developer has a more precise definition of the requirement and the analyst is educated in the capabilities and limitations of computer systems. The key is to maintain the dialogue between those two groups without the administrative delays caused by system change requests and other formal documentation employed in the traditional system development methodology. In a sense, this course is a part of this methodology, and feedback should be expected.
2.2 Hardware Configuration

MICROFIX System I consists of the following commercial data processing items and is manufactured to meet the NACSIM 5100A criteria of TEMPEST: a Central Processing Unit (CPU) with a minimum of 128K RAM, a full 128 ASCII set keyboard, joystick, dual 5-1/4" floppy disk drives, monochrome monitor, color monitor, dot matrix graphics printer, graphics entry device, laser video disk player, Winchester hard disk (20 MB), and a power conditioning system (110 and 220 models). Optional components consist of a paper tape reader/punch (PTR/P) and video cassette recorder (VCR). The CPU is the core of the system. It contains the Main Logic Board Assembly equipped with the various logic printed circuit boards and the interface cards for host and peripheral devices. The CPU performs the circuitry logic inherent to the microprocessor system. Connecting to the CPU are the plug-in diskette drives, keyboard, and monitors. The drivers are contained in the CPU assembly and can be easily removed and replaced. They accept the floppy diskettes which are used in processing data. There are two monitors in the system, one to display text and one for displaying graphics information. The keyboard plugs into the front panel of the CPU. Its purpose is to allow the operator to key in information and control joystick operations relative to the graphics and may displays. The keyboard is equipped with an accessible enhancer board that provides special operational functions such as auto repeat keying and lower case letters. The hard disk drive is for mass random access storage. When provided with a Mirror Board, it is able to transfer to and receive data from a VCR. The VCR is an add-on peripheral device that provides backup mass storage for the system. The graphics tablet is an output device to control and mark graphic representations. The power conditioner provides power overvoltage and power spike protection. There will be two models: one type for standard 60 Hz, 110 VAC, and the other for 50 Hz, 220 VAC. The video disk player stores and displays basic tactical maps. The printer provides a hard copy paper printout. The printer may, depending on operational requirements, be substituted for by a PTR/P. The PTR/P transmits and receives digital data.
2.3 Software

MICROFIX System I software is primarily a database management system for storage, manipulation, and display in text and graphic format. It is written in two languages, PASCAL and dBASE II, a database management language. The six major components include:

1. Information Retrieval System (IRS) -- enters, sorts, and maintains data in the database and generates usable battlefield reports.

2. Graphical Intelligence Analysis System (GIAS) -- allows the operator to spatially display order-of-battle and spot reports, roam across the video disk based maps, input sketches, and declutter the screen as required.

3. Collection Planning Aid (CPA) -- allows generation and maintenance of the collection plan, and formats tasking requests. Subsequent features include system selection, map display of collection assets, and coverage and collection reports.

4. Topographical Support (TOPO) -- provides for the building of a document related database and the retrieval of information by geographic area, subject, and other ancillary parameters. Additional software is currently being developed to evaluate battlefield environmental effects and for a slope analysis.

5. Commercial word processing and formatting packages.

6. System diagnostic test programs.
3.0 PLAN OF OPERATION

3.1 Instructional Personnel

The instructional team leader is Mr. Michael Rowan, a Research Scientist at Georgia Tech. Mr. Rowan has a background in geographic databases and spatial analysis. He is responsible for the design of the topographic library subsystem, and he is evaluating the feasibility of incorporating other terrain related capabilities into the MICROFIX system. Other members include: Mr. Ben Atha, a Research Scientist, who has worked on both hardware and software aspects of MICROFIX, including videodisk map graphics software; and Mr. John Peck, a Programmer, who has worked on the topographic library routines and the slope analysis software.

3.2 Location of Instruction

The course will be taught at the Defense Mapping School, Ft. Belvoir, Virginia from December 12 through December 16, 1983. The Government will furnish four MICROFIX systems and one VCR for the course.

3.3 Instructional Methodology

The course will cover both hardware and software elements of the system. The training will parallel procedures and topics developed for the New Equipment Training (NET) team courses being taught at other installations with two major exceptions. First, the NET course covers a two-week period, whereas the Ft. Belvoir course will be accomplished in one week. It will be necessary to reduce the amount of detail for some areas. Second, the NET course is primarily oriented toward the military intelligence analyst. In contract, the Ft. Belvoir course will focus more on applications related to the terrain analyst. Terrain related software, not yet implemented on the operational system, will also be demonstrated at this session.
Classroom procedures will consist of both lectures and practical exercise by the participants, with the emphasis on the latter. The small number of participants (approximately 12) is conducive to a high degree of student/instructor interaction and for a sufficient amount of hands-on experience with the computer. It is expected that these conditions will lead to feedback from the participants concerning ease of usage and overall capabilities of the topographic related software.

3.4 Typical Teaching Day

0800 - 0850  Period One
0900 - 0950  Period Two
1010 - 1100  Period Three
1110 - 1200  Period Four
1200 - 1300  Lunch
1300 - 1350  Period Five
1400 - 1450  Period Six
1510 - 1600  Period Seven
1610 - 1700  Period Eight
4.0 OUTLINE OF TOPOGRAPHIC PERSONNEL COURSE

1st Day

1 hour I. INTRODUCTION
   A. Overview of MICROFIX
      1. Background
      2. Outline of Course
      3. Introduction to MICROFIX System I Hardware
   B. Absolute Warnings
      1. Handling of Boards, Cabling, and Connectors
      2. Operations in a Hostile Environment

3 hours II. SYSTEM INITIALIZATION AND DIAGNOSTICS
   A. Power Considerations
   B. Startup Procedures
   C. Diagnostic Procedures
   D. System Check

4 hours III. CORVUS OPERATIONS AND BACKUP
   A. Overview
      1. Lights
      2. Front Panel Switches
      3. Boards/Equipment
      4. Cabling
      5. Physical Organization
      6. Volume Organization
   B. Initialization
      1. User Relationship to Volume/OS
      2. Mirror
      3. Backup and Loading Software from Backup

2nd Day
1 hour  IV.  INTRODUCTION TO COMPUTERS
   A.  Software
   B.  Operating Systems
   C.  Apple Computer
   D.  Database Management Systems

1 hour  V.  MICROFIX OVERVIEW
   A.  IRS
   B.  GIAS
   C.  CPA
   D.  TOPO

6 hours  VI.  TOPOGRAPHIC LIBRARY ROUTINE
   A.  Background
   B.  Record Structure and Indexing
   C.  Coding Standards
   D.  Data Input and Editing
   E.  Area Search

3rd Day  F.  Listing
         G.  Update

8 hours  VII.  OTHER TERRAIN RELATED ACTIVITIES
   A.  Slope Analysis
       1.  Data
       2.  Processing Techniques
       3.  Display
   B.  Battlefield Environmental Effects System
       1.  Overview
       2.  SWO Comments
       3.  Demonstration
   C.  General Discussions
4th Day

3 hours VIII. IRS (Information Retrieval System)
A. Spot Reports
B. Order of Battle
C. TOE/Attrition
D. Initialization
E. Backup
F. Command Line
G. Coordinate Conversion

3 hours IX. GIAS (Graphic Intelligence Analysis System)
A. Area Selection
B. Declutter
C. Roam
D. Reports
E. Location History
F. Create Sketch
G. Display Sketch

1 hour X. SOFTWARE CUSTOMIZATION
A. Symbols
B. Text Edit
C. Default Values

1 hour XI. APPLICABILITY OF OTHER FUNCTIONS TO TOPO PROBLEMS
5th Day

3 hours  XII. CPA (Collection Planning Aid)
A. Collection Plan
B. Tasking/Requests
C. System Selection
D. Coverge Map
E. Collection Reports

3 hours  XIII. OTHER SOFTWARE PACKAGES
A. Format Generator
B. WordStar

2 hours  XIV. REVIEW AND DISCUSSIONS
5.0 MATERIALS

The training team will provide copies of all visual aids and lesson plans to the participants during the course.
MICROFIX SYSTEM ONE
RELEASE DOCUMENTATION
(BEES)
Battlefield Environmental Effects System

Prepared by
Command and Control Division
Electronics and Computer Systems Laboratory
and the
Electro-Optics Division
Electromagnetics Laboratory
Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia 30332
September, 1984
# TABLE OF CONTENTS

Section 1 - Overview .......................................................... 1
  Filename Extensions ......................................................... 2
  MT+ System Libraries ...................................................... 2
  MICROFIX System Libraries ................................................ 2

Section 2 - Organization
  BEES System Structure ................................................... 3
  BEES System Library ..................................................... 3
  BEES System Driver ...................................................... 5
  BEES System Options .................................................... 6
    Density Altitude ....................................................... 6
    Moon Phenomena ........................................................ 8
    Sun Phenomena .......................................................... 13

Section 3 - Listings
  Data File ........................................................................... 14
  Menu Files ........................................................................ 17
  Program Listings ............................................................ 20

Section 4 - Utilities - (This section intentionally blank) .............. 57

Section 5 - Technical Items ................................................... 58
  Reference - Extract From "ALMANAC for COMPUTERS - 1984" ....... 58
  Original "BEES" BASIC Language Listing ............................... 89
Section 1

OVERVIEW
OVERVIEW

This software was adapted from programs in the Battlefield Environmental Effects System (BEES) which the Army Engineer Topographic Laboratories developed and implemented in BASIC on a Hewlett-Packard HP-85 microcomputer.

This system consists of a set of individual MT+ PASCAL programs connected by chaining. A driver program displays the main system menu on the user's terminal and either chains in an executable application program or exits to the calling level depending on the menu option chosen by the user. The programs associated with driver menu options contain chain statements to effect a return to the driver when they are terminated; the main menu is then redisplayed.

An individual program comprises a number of relocatable modules which are linked together to form a binary object (executable) program. Each linkable module is set up by putting the source code for a group of procedures and functions in a single disk file and compiling them as a unit. In this system procedures/functions are lumped together primarily based on logical relationship (e.g., to form a library or perform input functions for a given program). Modularization accommodates limitations on the amount of code the PASCAL compiler can compile at one time or limitations on the amount of text the SPP text editor can handle in a single source file. An overlay structure may be required if the amount of relocatable (compiled) code which must be linked together exceeds the upper limit on addressable memory space, 64K bytes. For this application the use of overlays has not so far been necessary.
FILENAME EXTENSIONS

Filenames under CP/M have the form NAME.EXT, where NAME usually serves as an application identifier and EXT describes the file's "type"; e.g., source file, relocatable, etc. The extensions used for BEES are listed below. They were chosen to conform with standard use in other MICROFIX software or enforced by system software requirements.

.BAK backup source files created by SPP text editor
.BLD PASCAL librarian command files for LIBMT
.CMD PASCAL linker command files
.COM executable object files produced by LINKMT
.ERL relocatable object files produced by PASCAL compiler
.MNU menu text files
.SRC PASCAL MT+ source files

MT+ SYSTEM LIBRARIES

The following relocatable libraries must be linked with other relocatable modules to create executable programs for the system. Some may not be needed by some programs. Necessary libraries are specified in the individual program descriptions.

FPREALS.ERL floating point numbers
PASLIB.ERL basic PASCAL library (comparisons, I/O, etc.)
RANDOMIO.ERL random disk file I/O
TRANCEND.ERL PASCAL transcendental functions

MICROFIX SYSTEM LIBRARIES

MICROFIX library CURLIB.ERL contains utility procedures for handling I/O for the terminal screen. The following procedures from CURLIB are used by BEES and declared EXTERNAL in modules which call them:

CLREDS clear to the end of the terminal screen
CLRSCR clears the terminal screen
CLRLN clears the current line on the screen
GOTOXY(X,Y: BYTE) moves cursor to position X,Y on the terminal screen; 0,0 is upper left corner.
MENU(NAME: STRING) reads menu text from disk file NAME and displays the text on the screen
Section 2

ORGANIZATION
BEES SYSTEM STRUCTURE

BEES driver
I
I
I
------
I
I
I
DENSITY  MOON  SUN
ALTITUDE  PHENOMENA  PHENOMENA

BEES SYSTEM LIBRARY

Relocatable library BEESLIB.ERL contains BEES procedures and functions used by more than one application program.

Source files:

IOMOD.SRC  I/O-related procedures or functions
MATHMOD.SRC  mathematical procedures or functions

Functions (MATHMOD.SRC)

The following REAL functions were implemented because PASCAL has no intrinsic arcsin and arccos functions and no mod function with real arguments, and the BASIC programs from which this software was adapted used degrees as trig function arguments and outputs whereas PASCAL uses radians. Source code for these functions is in file MATHMOD.SRC.

<table>
<thead>
<tr>
<th>Function</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIN(ARG: REAL)</td>
<td>ARCSIN(ARG), radian output</td>
</tr>
<tr>
<td>ACOS(ARG: REAL)</td>
<td>ARCCOS(ARG), radian output</td>
</tr>
<tr>
<td>DTR(ARG: REAL)</td>
<td>convert degrees to radians</td>
</tr>
<tr>
<td>RMOD(ARG, MODF: REAL)</td>
<td>modulus with real arguments</td>
</tr>
<tr>
<td>RTD(ARG:REAL)</td>
<td>convert radians to degrees</td>
</tr>
</tbody>
</table>

Procedures (MATHMOD.SRC)

GETLOCALTIME(LONG:REAL) Calculate time difference from LONG to Greenwich meridian and display on screen.

Procedures (IOMOD.SRC)

GET_NEXT_KEY_PRESSED(VAR C: CHAR)  
Get value for next keyboard character entry

GET_A_YESNO_ANSWER(VAR C: CHAR)  
As above but only accept and echo Y or N (lower case input is changed to upper case)

GET_MENU_CHOICE(MX:INTEGER;VAR C:INTEGER)  
Read keyboard input (up to 2 characters) followed by RETURN. Accept only integers between 1 and MX, or RETURN by itself which defaults to 0. Return result in C.
Functions (IOMOD.SRC)

CHAR_TO_INTEGER(STR:STRING): INTEGER
Convert a string to integer equivalent.

CHAR_TO_REAL(LINE:STRING): REAL
Convert a string to a real number.

INTIN(PROMPT:STRING;MIN,MAX:INTEGER): INTEGER
Use PROMPT to solicit integer input in interval [MIN,MAX].

KEYPRESSED: BOOLEAN
Detect keyboard input.

REALIN(PROMPT:STRING;MIN,MAX:REAL): REL
Use PROMPT to solicit real input in interval (MIN,MAX).

Library creation

Generation of linkable relocatable library BEESLIB.ERL requires compiling source code for individual modules to be included in the library and then processing the relocatable modules with the MT+ library processor LIBMT. LIBMT expects input in the form of a text file which contains the names of the output library file and all the relocatable modules to be entered in the library. The name of this text file must have extension BLD, as explained in the MT+ PASCAL manual. The linker LINKMT can be instructed to include only library modules containing routines which are called by the program the library is being linked with in the executable code it generates. The BEES library currently has two modules, one for mathematical routines and one for I/O related routines.

To generate library BEESLIB.ERL:

1. Compile source files BEESMOD.SRC and MATHMOD.SRC
2. Prepare (if necessary) text file BEESLIB.BLD containing BEESLIB.ERL IOMOD.ERL MATHMOD.ERL
3. Use the command: LIBMT BEESLIB
BEES SYSTEM DRIVER

Source file:

**BEES.SRC**  Driver to display main menu and chain to other system programs.

Procedures: main procedure

Data files:

**BEES.MNU**  Contains text of main menu. This is an ASCII file created with a text editor.

Relocatable modules needed to link BEES:

**BEES.ERL, BEESLIB.ERL, CURLIB.ERL, PASLIB.ERL**

Linker command file: **LKBEES.CMD**

Adding a new option requires:

1. Adding the option to the menu text in file **BEES.MNU**.
2. Adding an appropriate chain statement to the source code in **BEES.SRC**. Also the MX parameter in the call to **GETMENUCHOICE** must equal the largest menu option number on the main menu.
3. Compiling **BEES.SRC** and linking the **BEES** program.
4. Adding a 'chain back' statement to the program to be used as the new option, compiling and linking it.
BEES SYSTEM OPTIONS
---------------------

DENSITY ALTITUDE

Discussion:

This program computes density altitude from user supplied input data and can also compute loading capacity for certain helicopter models at the current density altitude. The program was adapted fairly straightforwardly from the analogous ETL BASIC program, and all possible comments and variable names have been preserved or clarified in the PASCAL version. Theoretical discussion of the equations used by the program is beyond the scope of this document. Parameters for the implemented helicopter models were obtained from the ETL BASIC program.

Source files:

- COPTER.SRC Copter menu call; calculations for some helicopters.
- COP2.SRC Calculations for remaining helicopters.
- DENSALT.SRC Input and density altitude computations.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Source file</th>
</tr>
</thead>
<tbody>
<tr>
<td>AH_IG</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>AH_1S</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>CALCULATE_DENSITY_ALTITUDE</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>CH_47A</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>CH_47B</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>CH_47C</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>CH_54A</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>CH_54B</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>COMPUTE_A_LOAD_CAPACITY</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>DO_DENSITY_ALTITUDE_CALCULATIONS</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>DO_LOAD_CAPACITY_CALCULATIONS</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>Main</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>OH_6A</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>OH_58A</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>OH_58C</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>GET_DENSITY_ALTITUDE_INPUTS</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>GET_PALTITUDE_FROM_PRESSURE</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>GET_PRESSURE_FROM_PALTITUDE</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>HEADING</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>PRINT_DENSITY_ALTITUDE</td>
<td>DENSALT.SRC</td>
</tr>
<tr>
<td>PRINT_OUTPUT</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>SETUP_LOAD_CAPACITY_CALCULATIONS</td>
<td>COPTER.SRC</td>
</tr>
<tr>
<td>UH_1CM</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>UH_IDH</td>
<td>COP2.SRC</td>
</tr>
<tr>
<td>UH_60A</td>
<td>COP2.SRC</td>
</tr>
</tbody>
</table>
Relocatable modules needed to link DENSALT:

DENSALT.ERL, COPTER.ERL, COP2.ERL, BEESLIB.ERL, CURLIB.ERL, FPREALS.ERL, TRANCEEND.ERL, PASLIB.ERL

Linker command file: LKDENS.CMD

Adding a new helicopter requires:

1. Adding the model to the helicopter menu in HELI.MNU.
2. Adding a procedure with appropriate calculations to source file COP2.SRC.
3. Adding the new case to the CASE statement, procedure COMPUTE_A_LOAD_CAPACITY in file COPTER.SRC.
4. Compiling COPTER.SRC and COP2.SRC and linking DENSALT.

Notes:

Source file COPTER.SRC got too big for the SPP editor so the rest of the helicopter calculations were placed in file COP2.SRC.

Density Altitude is close to running out of memory when it is linked. Adding more than a few lines of code may require space reduction elsewhere in the program.
MOON PHENOMENA

Discussion:

The Moon Phenomena program has two distinct functions: calculate moonrise/moonset times for a specific date and geographic location between 60 deg N and 60 deg S for the current year, or output dates for moon phases for a specific month in the current year. The operator inputs date/location information but the program must have access to additional data about the moon’s movements which must be updated annually. The source for the latter data is the Almanac for Computers available from the Naval Almanac Office of the U.S. Naval Observatory in Washington, D.C. This document contains the necessary data and information about the algorithm used to calculate moonrise/moonset times.

Monthly phase dates can be acquired by the program by straightforward retrieval. The dates are not calculated, but extracted from the almanac and put somewhere the program can find them. Since each of 12 months has 4 or 5 dates when moon phases (full, new, waxing crescent, waning crescent) occur, this means storage of about 50 data items. The most obvious way to do this in FORTRAN or BASIC would be to store them in a 1 or 2 dimensional array by using DATA statements and then get the appropriate numbers out of the array when needed. Alas, PASCAL has nothing like a DATA statement and keeping these numbers in the program itself would require 50 separate assignment statements.

The moonrise/set times are arrived at via an iterative process which employs (as of the 1984 Almanac) a 7th order series expansion. The data set for the algorithm used in the 1984 almanac consists of 768 separate floating point coefficients which may be used in the series expansion (16 each for 48 "weeks") and several constants which apply to the year itself.

The "first draft" PASCAL version of this program avoided the need for separate assignment statements by using a sequential access disk file to store all the almanac data. The use of a data file reduces the size of the program and has the additional attractive aspect of eliminating the need to edit, recompile and relink the program when the data must be modified. The data file can be updated by using a text editor. However, use of a sequential file causes the program to execute rather slowly since it has to read every number in the file until it gets to the ones it needs. The slowness problem was eliminated by using a separate program to produce a random access file which contains essentially the same data in a somewhat different format.
Program RANREAD

-------------

This program reads a sequential disk file containing a year's worth of moon data and writes a random access disk file containing the same data in a modified format. Input file name is MOONDAT, output filename is MOONRAN. MOONDAT is created by extracting appropriate data from the Almanac for Computers and putting it in the file with a text editor. Specifications for both data files are included here. RANREAD needs to be executed only to establish the MOONRAN file.

Source file: RANREAD.SRC

Procedures:

Main procedure
MOVE_DATA_TO_RANDOM_FILE

Data files:

MOONDAT- Contains moon data for current year, extracted from the Almanac for Computers. This is an ASCII text file which must be created by using a text editor. The RANREAD program reads it and writes the data to random access file MOONRAN. The Moon Phenomena program gets data from MOONRAN. The file structure for MOONDAT is as follows:

line 1: 4 digit year (e.g. 1984)
line 2: "A" value for the current year (4 in 1984)
line 3: "D5" value for the current year (347.81 in 1984)
line 4: month name (first is JANUARY)
line 5: 6 integers separated by spaces.

Integer 1 is a code number the program will use to decide what column headings to output for phase date info for this month. Some months have 4 phase dates, others have 5. The dates are most reasonably output in increasing order. The codes are as follows:

1   WANING, NEW, WAXING, FULL
2   NEW, WAXING, FULL, WANING
3   WAXING, FULL, WANING, NEW
4   FULL, WANING, NEW, WAXING
10  WANING, NEW, WAXING, FULL, WANING
20  NEW, WAXING, FULL, WANING, NEW
30  WAXING, FULL, WANING, NEW, WAXING
40  FULL, WANING, NEW, WAXING, FULL

Integers 2 - 6 are days of the current month when the phases represented by the codes above occur. Integer 6 should be zero for months with only 4 phase dates.
8 coefficients for moon GHA power series for week 1 of this month, in order c7....c0 and separated by spaces.

8 coefficients for moon declination power series for week 1 of this month, in order c7...c0 and separated by spaces.

same as line 6, for week 2.
same as line 7, for week 2.
same as line 6, for week 3.
same as line 7, for week 3.
same as line 6, for week 4.
same as line 7, for week 4.

Repeat lines 4 - 13, 11 more times for the 11 remaining months.

IMPORTANT NOTES

1. For formatting the MOONDAT file, numbers on the same line must be separated by SPACES. Commas are NOT valid separators.

2. All decimal points must be embedded between 2 visible digits. Leading and trailing decimal points are unacceptable to the PASCAL compiler. (This applies to numbers within programs or numbers to be read using standard PASCAL I/O.)

Examples: ACCEPTABLE NOT ACCEPTABLE

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>.5</td>
</tr>
<tr>
<td>332.0</td>
<td>332.</td>
</tr>
<tr>
<td>-0.7</td>
<td>-.7</td>
</tr>
<tr>
<td>0.13E-5</td>
<td>13E-5</td>
</tr>
</tbody>
</table>

A copy of the 1984 MOONDAT data file appears at the end of this section.

MOONRAN - random access moon data file produced by RANREAD.

This file's structure is defined by its record definition in programs RANREAD and MOON. All records in the file have the same structure and exactly the same length, and are stored contiguously on the disk. Instead of reading sequentially through the file to locate data, the random I/O handler calculates the
physical location of a record by its record number and starts reading data at that location. I/O is fast and any record can be found in the same length of time. Random files and random file I/O are discussed in the MT+ PASCAL manual on pp. 184 - 188. The record structure for the 12-record MOONRAN file is as follows:

```pascal
MONTHNAME: STRING[9];
YEAR: INTEGER;
RA: INTEGER; ("A" constant for current year)
RDS: REAL; ("D5" constant for current year)
WD: INTEGER; (1st integer from 'line 5' of MOONDAT)
PHASEDATE: ARRAY[1..5] OF INTEGER; (rest of 'line 5')
MONTHDATA: ARRAY[1..4,1..16] OF INTEGER; (coefficients)
```

Refer to the MT+ PASCAL manual and program listings of RANREAD.SRC and MOON.SRC to see how this file is used.

Relocatable modules needed to link RANREAD:

```
RANREAD.ERL,FPREALS.ERL,RANDOMIO.ERL,PASLIB.ERL
```

Linker command file: none
Program MOON

Source files:

  INMOON.SRC  Performs date/location input.
  MOON.SRC    Main/computation module for moon phenomena.

Procedure Source file

  ASSIGN_MONTHS    INMOON.SRC
  CHOOSE_PROGRAM_OPTION MOON.SRC
  FIND_DATA        MOON.SRC
  GET_DATE_AND_LOCATION INMOON.SRC
  GET_MONTH_AND_PRINT_PHASES MOON.SRC
  GHA_AND_DEC      MOON.SRC
  ITERATE          MOON.SRC
  Main procedure   MOON.SRC
  PHENOMENA_TIME_CALCULATIONS MOON.SRC
  PRINT_MOON_HEADING MOON.SRC
  PRINT_OUTPUT     MOON.SRC
  SETUP            MOON.SRC

Data files:

  MOON.MNU        Contains text of moon phenomena option menu.
  MOONRAN.SRC     Moon data input for current year. This file was
discussed along with the RANREAD program.

Relocatable modules needed to link MOON:

  MOON.ERL, INMOON.ERL, BEESLIB.ERL, CURLIB.ERL, FPREALS.ERL,
  TRANCEND.ERL, RANDOMIO.ERL, PASLIB.ERL

Linker command file: LKMOON.CMD
SUN PHENOMENA

Discussion:

This program computes sun phenomena (sunrise/sunset, civil twilight, nautical twilight, and astronomical twilight) for user specified date and geographic location between 60 deg N and 60 deg S. It requires no additional data or annual update. The algorithms used are described in the Almanac for Computers.

Source files:

INSUN.SRC     Input module for sun phenomena.
SUNRISE.SRC   Main processing module for sun phenomena.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Source file</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN_MONTHS</td>
<td>INSUN.SRC</td>
</tr>
<tr>
<td>COMPUTE_PHENOMENA_TIMES</td>
<td>SUNRISE.SRC</td>
</tr>
<tr>
<td>FIND_RIGHT_ASCENSION_AND_DECLINATION</td>
<td>SUNRISE.SRC</td>
</tr>
<tr>
<td>GET_DATE_AND_LOCATION</td>
<td>SUNRISE.SRC</td>
</tr>
<tr>
<td>GET_INPUTS</td>
<td>INSUN.SRC</td>
</tr>
<tr>
<td>Main procedure</td>
<td>SUNRISE.SRC</td>
</tr>
<tr>
<td>PRINT-SUN-HEADING</td>
<td>SUNRISE.SRC</td>
</tr>
</tbody>
</table>

Relocatable modules needed to link SUNRISE:

SUNRISE.ERL, INSUN.ERL, BEESLIB.ERL, CURLIB.ERL, FPREALS.ERL, TRANCEND.ERL, PASLIB.ERL

Linker command file: LKSUN.CMD
Section 3

LISTINGS
1: 1984
2: 4
3: 347.81
4: JANUARY
5: 2 3 11 18 25 0
6: 1956.697 1392.2494 4.7149 1.0619 -1.3222 -0.041 0.1996 -0.036 
7: 23.0755 8.7563 4.9178 -0.0434 0.1222 0.0513
8: 1401.7515 1386.0607 7.0169 -0.8042 -2.1805 0.2125 -0.213
9: 14.7271 -21.1826 -6.7235 4.9178 -0.0434 -0.7043 0.1862 0.0636
10: 1664.3651 1389.577 -1.1272 2.0618 0.8783 -0.4557 -0.1409 0.0379
11: -24.0859 -6.7991 9.8414 0.6801 -0.7801 -0.1219 0.0444 0.0213
12: FEBRUARY
13: 2 1 10 17 23 0
14: 1583.2818 1400.0894 2.3508 -1.9301 -0.3858 0.102 -0.0344 0.0177
15: -9.3321 18.9236 2.7345 -1.7525 0.142 -0.0355 -0.0326 0.0029
16: 1494.3531 1382.9185 -6.9839 3.3789 2.479 -0.4741 -0.5385 -0.0037
17: 24.7242 5.5618 -11.9805 -3.141 -1.1162 0.0678 0.0223 -0.0171
18: 1743.5885 1391.4612 0.61 -2.2693 0.5491 0.3466 -0.08 -0.0014
19: -7.1489 -23.6776 4.4485 3.1948 -0.7645 0.0781 0.0332 -0.0383
20: 1644.8979 1393.9355 4.4829 0.3412 -0.9961 0.059 0.1217 -0.034
21: MARCH
22: 2 2 10 17 24 0
23: 1593.0264 1400.9025 -0.428 -2.1974 -0.1539 0.0678 0.0226 0.0113
24: -0.5209 20.5402 0.4576 -1.7649 -0.0877 -0.0124 0.0072
25: 1497.7162 1381.5885 -1.2897 4.6442 -0.689 -1.3327 0.0084 0.234
26: 25.0777 -23.6776 4.4485 3.1948 -0.4741 -0.5385 -0.0037
27: -0.5209 20.5402 0.4576 -1.7649 -0.0877 -0.0124 0.0072
28: 1474.0227 1388.4301 3.1077 -1.529 0.5295 0.0683 -0.0578
29: 10.9676 -23.7277 -5.6194 4.2293 0.471 -0.2849 0.0133 0.0041
30: 1730.4134 1387.148 3.0262 2.8268 -0.6343 -0.5651 0.1302 0.0737
31: -25.8676 -0.8498 11.0932 -1.1508 -0.9222 -0.0616 0.0332 -0.0383
32: 1652.7643 1393.9724 3.8831 -1.364 -0.5177 0.2235 -0.0189 -0.0329
33: -16.0829 16.3826 4.9265 -0.8491 0.0788 -0.019 -0.0393 0.0227
34: APRIL
35: 2 1 9 15 23 0
36: 1580.8114 1394.0368 -6.1669 -1.2152 0.9618 -0.4734 -0.0222 -0.0827
37: 17.8407 13.6132 -5.7978 -2.7625 -0.2204 0.1394 0.0891 0.0101
38: 1474.0227 1388.4301 3.1077 -1.529 -1.0297 0.5295 0.0683 -0.0578
39: 10.9676 -23.7277 -5.6194 4.2293 0.471 -0.2849 0.0133 0.0041
40: 1730.4134 1387.148 3.0262 2.8268 -0.6343 -0.5651 0.1302 0.0737
41: -25.8676 -0.8498 11.0932 -1.1508 -0.9222 -0.0616 0.0332 -0.0383
42: 1642.1874 1400.4176 -1.6840 -2.2857 0.0064 0.0481 0.047 0.0397
43: 1.9147 21.0725 0.0251 -1.903 -0.2588 -0.1089 0.0219 0.02
44: MAY
45: 2 1 8 15 22 0
46: 1572.6984 1385.2009 -3.963 3.4604 1.2123 -0.7677 -0.27 0.0921
47: -25.536 4.325 -11.5658 1.2123 -0.7677 -0.27 0.0921
48: 1468.1854 1391.1379 -1.8198 -2.2545 0.6303 0.4641 -0.0107 -0.0245
49: -5.6221 -24.9319 2.575 4.014 -0.0204 -0.1469 -0.0457 -0.0235
50: 1724.1429 1392.9074 6.284 -0.3137 -1.3945 0.3956 0.1302 0.0737
51: -22.0919 11.7944 7.4617 -2.4286 0.1157 0.2405 -0.1031 -0.0026
52: 1641.1095 1394.8413 -6.4668 -1.712 0.957 0.5667 0.0309 -0.0839
53: 15.4139 18.3217 -4.6046 -2.9777 -0.4553 0.1163 0.1227 0.0297
54: JUNE
55: 3 6 13 21 29 0
56: 1547.4293 1387.2098 5.3086 0.0616 -2.0493 0.4293 0.2691 -0.1264
57: 18.9816 -17.4897 8.8187 3.3523 0.4594 -0.4374 0.0864 0.0448
58: 1448.3706 1386.1382 -2.6657 2.7725 1.5622 -0.5559 -0.2666 0.043
59: -23.266 -11.135 11.0061 1.7642 -1.2924 -0.323 0.12 0.0635
60: 1712.2374 1400.7844 1.7076 -2.5578 -0.1633 0.0884 -0.045 0.0233
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23.1024</td>
<td>-13.276</td>
<td>11.5975</td>
<td>2.8067</td>
<td>-1.412</td>
<td>-0.6415</td>
<td>0.1165</td>
</tr>
<tr>
<td></td>
<td>1551.2149</td>
<td>1398.8825</td>
<td>4.7572</td>
<td>-2.5992</td>
<td>0.3171</td>
<td>0.29</td>
<td>-0.1149</td>
</tr>
<tr>
<td></td>
<td>-11.63</td>
<td>19.7582</td>
<td>2.8291</td>
<td>-2.1343</td>
<td>0.504</td>
<td>-0.1033</td>
<td>-0.0728</td>
</tr>
</tbody>
</table>

FILE MOONDAT.
1. DENSITY ALTITUDE
2. MOON PHENOMENA
3. SUNRISE/SUNSET/TWILIGHT TIMES

FILE BEES.MNU
MICROFIX BEES: DENSITY ALTITUDE

HELI OPTER SELECTION

1. AH-1G 8. EH-1H
2. AH-1S 9. OH-6A
3. CH-47A 10. OH-58A
4. CH-47B 11. OH-58C
5. CH-47C 12. UH-1C/M
6. CH-54A 13. UH-1D/H
7. CH-54B 14. UH-60A

FILE HELI.MNU
1. Moonrise/Moonset times (1984)

2. Moon phase dates for months in 1984

FILE MOON.MNU
Program Listings
(* **********************************************************************
 * PROGRAM BEES
 * Programmer:  Susan R. Wheeler, Ga Tech/GTRI
 * Source file:  BEES.SRC
 * Last modified:  7/23/84
 *
 * This is the driver module for the BEES system. It displays the main menu
 * and chains to an application or terminates BEES as the user chooses.
 *
 * ********************************************************************** *)

PROGRAM BEES;

VAR
  OPTION, QUIT: INTEGER;
  CHAINFIL: FILE;

EXTERNAL FUNCTION CHARTOINTEGER(LINE: STRING): INTEGER;  (* BEESLIB *)
EXTERNAL PROCEDURE CLRSCR;  (* CURLIB *)
EXTERNAL PROCEDURE GETMENUCHOICE(MX: INTEGER; VAR C: INTEGER);  (* BEESLIB *)
EXTERNAL PROCEDURE GOTOXY(X, Y: BYTE);  (* CURLIB *)
EXTERNAL PROCEDURE MENU(NAME: STRING);  (* CURLIB *)

BEGIN
  QUIT := 0;
  CLRSCR;
  MENU('BEES.MNU');
  GETMENUCHOICE(3, OPTION);
  IF OPTION <> QUIT THEN BEGIN
    CASE OPTION OF
      1: ASSIGN(CHAINFIL, 'DENSALT.COM');
      2: ASSIGN(CHAINFIL, 'MOON.COM');
      3: ASSIGN(CHAINFIL, 'SUNRISE.COM');
    END;  (* CASE *)
    CLRSCR;
    GOTOXY(1, 10);
    WRITE('Working......');
    RESET(CHAINFIL);
    CHAIN(CHAINFIL)
    END  (* IF *)
  END.  (* BEES *)

END.
MODULE IOMOD;

TYPE
  STR6 = STRING[6];
  STR8 = STRING[8];
  STR15 = STRING[15];
  STR30 = STRING[30];
  STR35 = STRING[35];

EXTERNAL FUNCTION OBDO(FUNC: INTEGER; PARM: INTEGER): INTEGER;
EXTERNAL PROCEDURE CLRNL;
EXTERNAL PROCEDURE CLREOS;
EXTERNAL PROCEDURE GOTOXY(X, Y: BYTE);

FUNCTION CHARTOINTEGER(LINE: STR6): INTEGER;
(* convert string of digits in line to an integer *)
VAR
  IDX, NUMBER: INTEGER;
BEGIN
  NUMBER := 0;
  FOR IDX := 1 TO LENGTH(LINE) DO
    NUMBER := NUMBER * 10 + ORD(LINE[IDX]) - ORD('0');
  CHARTOINTEGER := NUMBER
END;

FUNCTION CHARTOREAL(LINE: STR8): REAL;
(* convert a string (up to 8 chars to a real number *)
(* does minimal error checking *)
VAR
  IDX, DIVISOR, START: INTEGER;
  NUMBER: REAL;
  FOUNDPOINT, FOUNDMINUS: BOOLEAN;
BEGIN
  NUMBER := 0;
  DIVISOR := 10;
  FOUNDPOINT := FALSE;
  FOUNDMINUS := FALSE;
  START := 0;

61: REPEAT START:= START + 1 UNTIL LINE[START] <> ' '; (* check leading blanks *)
62: IF LINE[START] = '-' THEN BEGIN
63:  FOUNDMINUS:= TRUE;
64:  START:= START + 1
65:  END;
66: FOR IDX:= START TO LENGTH(LINE) DO
67: IF LINE[IDX] = '.' THEN FOUNDPOINT:= TRUE
68: ELSE
69: IF FOUNDPOINT THEN BEGIN
70:  NUMBER:= NUMBER + ((ORD(LINE[IDX]) - (ORD('0'))) / DIVISOR)
71:  DIVISOR:= DIVISOR * 10
72: END
73: ELSE NUMBER:= (NUMBER * 10) + (ORD(LINE[IDX]) - (ORD('0'))) * (ORD(LINE[IDX]) - (ORD('0')));
74: IF FOUNDMINUS THEN NUMBER:= -1 * NUMBER;
75: IF ABS(NUMBER) < 1.0E-10 THEN NUMBER:= 0.0;
76: CHARTOREAL:= NUMBER
77: END;
78:
79:
80: FUNCTION INTIN(PROMPT:STR30;MIN,MAX: INTEGER): INTEGER;
81:
82: (* get an integer input in the interval [MIN,MAX], Repeat PROMPT till *)
83: (* they enter a valid input. *)
84: (*)
85: VAR
86: LINE: STRING[5];
87: DUMMY: INTEGER;
88: BEGIN
89: REPEAT
90:  WRITE(PROMPT);
91:  READD(LINE);
92:  DUMMY:= CHARTOINTEGER(LINE)
93:  UNTIL (DUMMY = MIN AND DUMMY = MAX);
94: INTIN:= DUMMY
95: END;
96:
97:
98:
99: FUNCTION KEYRESSED : BOOLEAN; (* detect keyboard input *)
100:
101: BEGIN
102: KEYRESSED := (BBDOS(11,0) <> 0)
103: END;
104:
105:
106: FUNCTION REALIN(PROMPT:STR35;MIN,MAX:REAL): REAL;
107:
108: (* get a real number in interval [MIN,MAX]. Repeat PROMPT till *)
109: (* CHARTOREAL thinks it has a valid input. *)
110:
111: VAR
112: LINE: STRING[9];
113: NUMOUT: REAL;
114: BEGIN
115: REPEAT
116:  WRITE(PROMPT);
117:  READD(LINE);
118:  NUMOUT:=CHARTOREAL(LINE);
119:  UNTIL (NUMOUT = MAX AND (NUMOUT = MIN);
REALIN := NUMOUT
END;

PROCEDURE Get_next_key_pressed(VAR C : CHAR);
(* most of procedure from larry becker 12/12/84 *)
(* empty keyboard buffer and wait for keypress, return with key value *)
VAR
K, I : INTEGER;
BEGIN
I := 0;
REPEAT
(* empty keyboard buffer of type-aheads *)
K := @BDOS(6, $FF); (* DIRECT CONSOLE I/O *)
I := I + 1;
UNTIL I = 128; (* empty 128 characters ahead *)
REPEAT
I := @BDOS(6, $FF);
UNTIL I = 0;
C := CHR(I);
(* xfer to upper case if necessary *)
IF (C >= 'a') AND (C <= 'z') THEN
BEGIN
I := I - (ORD('a') - ORD('A'));
C := CHR(I);
END;
K := @BDOS(6, $FF); (* DIRECT CONSOLE I/O *)
END; (* Get_key *)

PROCEDURE GET_A_YESNO_ANSWER(VAR C : CHAR);
(* get keyboard input limited to (i t y l N,n) *)
(* followed by RETURN *)
BEGIN
REPEAT
GET_NEXT_KEY_PRESSED(C);
IF (C = 'Y') OR (C = 'N') THEN
BEGIN
WRITE(C);
READLN;
EXIT
END;
ELSE
WRITE(CHR(7));
UNTIL FALSE (* FOREVER *)
END;

PROCEDURE GETMENUCHOICE(MAX:INTEGER; VAR OPTION:INTEGER);
(* returns an integer in interval [0,MAX] in the OPTION argument *)
VAR CHOICE: STRING[3];
BEGIN
GOTOXY(1,21);
CLREDS;
WRITELN('-----------------------------------------------');
WRITE('Enter a number from 1 to ',MAX:2,' and RETURN: RETURN to exit. ');
GOTOXY(39,22);
READLN(CHOICE);
IF CHOICE = ' ' THEN OPTION:= 0
ELSE OPTION:= CHARTOINTEGER(CHOICE);
IF (OPTION ( 0) OR (OPTION ) MAX) THEN REPEAT
GOTOXY(0,22);
CLRLN;
WRITE('Sorry, ',CHOICE, ' is not a valid option. RETURN to continue:');
READLN;
GETMENUCHOICE(MAX,OPTION)
UNTIL (OPTION = 0) AND (OPTION (= MAX)
END;

PROCEDURE MENU(NAME:STR15);

CONST
END_FILE = $1A;
(* displays text file NAME on the *)
(* terminal screen, *)

VAR
SRC : TEXT;
CH : CHAR;
ERR : INTEGER;

BEGIN Of menu if)
OF'EN(SRC,NAME,ERR);
IF ERR=255 THEN
BEGIN
WRITELN('NO FILE ',NAME);
EXIT
END;
CH:= GNB(SRC);
REPEAT
ERR:=@BDOS(6,ORD(CH));
CH:= GNB(SRC);
UNTIL CH=CHR(END_FILE)
END; (if MENU JO

PROCEDURE PAUSE;
(* get keyboard input before continuing *)
BEGIN
GOTOXY(0,23);
WRITE('RETURN to continue: ');
READLN
END;

MODEND.
(* ***********************************************************************
 * MODULE MATHMOD
 * Programmer: Susan R. Wheeler, Ga Tech/GTRI
 * Source file: MATHMOD.SRC
 * Last revised: 8/22/84
 * This module contains mathematical procedures and functions to be part of
 * the BEESLIB library.
 * *********************************************************************** *)

MODULE MATHMOD;   (* MATHMOD.SRC revised 2/28/84 *)

EXTERNAL FUNCTION BDOS(FUNC: INTEGER; PARM: INTEGER): INTEGER;

FUNCTION ASIN(ARG: REAL) : REAL; (* arccosin, output is radians *)
BEGIN
ASIN:= ARCTAN(ARG / SQRT(1 - SQR(ARG)));
END;

FUNCTION ACOS(ARG: REAL) : REAL; (* arcsin, output is radians *)
BEGIN
ACOS:= 1.5708 - ARCTAN(ARG / SORT(.1 - SQR(ARG)));
END;

FUNCTION DTR(ARG: REAL) : REAL; (* convert degrees to radians *)
BEGIN
DTR:= 6.283185 * ARG / 360
END;

FUNCTION RMOD(ARG, MODF: REAL) REAL; (* modulus with real arguments *)
VAR TEMP: REAL;
BEGIN
TEMP:= ARG - MODF * TRUNC (ARG / MODF);
IF TEMP < 0 THEN TEMP:= MODF - TEMP;
RMOD:= TEMP
END;

FUNCTION RTD(ARG: REAL) : REAL; (* radians to degrees *)
BEGIN
RTD:= 360 * ARG / 6.283185
END;

PROCEDURE GETLOCALTIME(LONG: REAL); (* get absolute time difference *)
(* from the greenwich meridian *)
VAR
HCHANGE,MCHANGE: INTEGER;
MINUTES: REAL;
CHANGE: STRING;
BEGIN
IF LONG < 0.0 THEN CHANGE:= 'earlier than'
ELSE IF LONG > 0.0 THEN CHANGE := 'later than'
ELSE CHANGE := 'the same as';
HCHANGE := TRUNC(ABS(LONG/15));
MINUTES := ABS(LONG) - 15 * HCHANGE;
MCHANGE := TRUNC(4 * MINUTES);
WRITELN;
WRITE('Time at this longitude is ',CHANGE,' ZULU time');
IF LONG <> 0 THEN WRITELN(' by ',HCHANGE:2,' hours ',
MCHANGE:2,' minutes')
END;
This is the main module of the program which computes density altitude from temperature, elevation, and pressure inputs and optionally outputs load capacities for certain helicopter models at the density altitude just computed. 14 helicopter models have been implemented. Model choice is made from a menu. All possible comments have been preserved from the original BASIC source as well as most variable names. Floating point precision of the MT+ PASCAL compiler is approximately 6.5 digits whereas the precision of BASIC on the HP85 was 12 digits. All digits appearing in the BASIC program were transferred to the PASCAL version, however. Comparison of load capacities output by the PASCAL and BASIC versions shows a typical agreement within 3-5 pounds. No test case so far has varied more than 8 pounds.

DENSITY ALTITUDE is chained to by, and chains back to the main BEES menu.

************************************************************************

PROGRAM DENSALT;

VAR

ANSWER: CHAR;

BP,C,CDEWPT,D,D2,E,ELEV,F,FDEWPT,G,H,I: REAL;

ICEPT,PA,R,S,STDATM,STEAMPT,SVP,TEMP,Z: REAL;

CHAINFIL: FILE;

EXTERNAL FUNCTION REALIN(PROMPT:STR30;MAX,MIN:REAL):REAL; (* BEESLIB *)

EXTERNAL PROCEDURE CLRSCR; (* CURLIB *)

EXTERNAL PROCEDURE DOLOADCAPACITYCALCULATIONS; (* COPTER *)

EXTERNAL PROCEDURE GETYESNOANSWER(VAR C: CHAR); (* BEESLIB *)

EXTERNAL PROCEDURE GETMENUCHOICE(MX:INTEGER;VAR C:INTEGER); (* BEESLIB *)

EXTERNAL PROCEDURE MENU(NAME:STRING); (* CURLIB *)

EXTERNAL PROCEDURE PAUSE; (* BEESLIB *)

PROCEDURE CALCULATEDENSITYALTITUDE;

BEGIN

IF BP <> 0 THEN (* CONVERT PRESSURE TO PRESSURE ALTITUDE *)

GETPALTITUDEFROMPRESSURE

ELSE (* CONVERT PRESSURE ALTITUDE TO PRESSURE *)

GETPRESSUREFROMPALTITUDE;

ICEPT:= 273.16;

STDATM:= 1013.246;

STEAMPT:= 373.16;

IF CDEWPT > 0 THEN

BEGIN

...
CDEWPT := CDEWPT + ICEPT;
D := -7.90298 * (STEAMPT / CDEWPT - 1);
E := 5.02808 * LN(STEAMPT / CDEWPT) / LN(10);
R := 1 - CDEWPT / STEAMPT;
F := 1.3816E-7 * (EXP(11.344 * R * LN(10)) - 1);
S := STEAMPT / CDEWPT - 1;
G := 8.1328E-3 * (EXP(-3.49149 * S * LN(10)) - 1);
H := LN(STDATM) / LN(10);
I := D + E - F + G + H;
Z := EXP(I * LN(10));
SVP := Z * 0.02953
END

ELSE
BEGIN
D2 := CDEWPT + ICEPT;
D := -9.09718 * (ICEPT / D2 - 1);
E := -3.56654 * LN(ICEPT / D2) / LN(10);
F := 0.876793 * (1 - D2) / ICEPT;
G := LN(6.1071) / LN(10);
I := D + E + F + G;
Z := EXP(I * LN(10));
SVP := Z * 0.02953
END;

F := 0.622 * SVP / (BP - SVP); (* MXING RATIO CALCULATIONS *)
I := (TEMP + 459.69) * (1 + 1.61 * F) / (1 + F); (* VIRTUAL TEMP CALC *)
DA := (1 - EXP(0.235 * LN(17.326 * BP / I))) * 145366.0; (* DENSITY ALT *)
END;

PROCEDURE DODENSITYALTITUDECALCULATIONS;
BEGIN
GETDENSITYALTITUDEINPUTS;
CALCULATEDENSITYALTITUDE;
PRINTDENSITYALTITUDE;
WRITELN(WRITELN('Do you want load capacities for selected helicopters? ');
WRITE(' at this density altitude? (Y or N) ');
GETAYESNOANSWER(ANSWER);
IF ANSWER = 'Y' THEN DLLOADCAPACITYCALCULATIONS
END;

PROCEDURE HEADING;
BEGIN
CLRSCR;
WRITELN('  	 MICROFIX BEES:  	 DENSITY ALTITUDE');
WRITELN('------------------------------',
'------');
WRITELN('------------------------------',
'------')
END;

PROCEDURE GETDENSITYALTITUDEINPUTS;
BEGIN
HEADING;
END;
121:  WRITELN;
122:  ELEV:= REALIN('Enter STATION ELEVATION (feet): ', -500, 30000.0);
123:  WRITELN;
124:  TEMP:=REALIN('Enter TEMPERATURE (degrees F): ', -150, 150);
125:  WRITELN;
126:  FDEWPT:= REALIN('Enter DEW POINT (degrees F): ', -150, 150);
127:  CDEWPT:=5 * (FDEWPT-32) / 9;
128:  WRITELN;
129:  WRITELN('Enter UNCORRECTED STATION PRESSURE (inches Hg) or a "0" if only the PRESSURE ALTITUDE is known: ');
130:  WRITELN;
131:  BP:= REALIN('Enter 0 or barometric pressure: ', 0, 33);
132:  WRITELN;
133:  END;
134:  
135:  PROCEDURE GETPALTITUDEFROMPRESSURE;
136:  {
137:      (* CONVERT PRESSURE TO PRESSURE ALTITUDE *)
138:  BEGIN
139:      PA:= BP * 33.8639;  {(* CONVERT INCHES HG TO MILLIBARS *)
140:      IF PA > 1013.25 THEN PA:= 26859.7345224 - 26.5185 * PA
141:      ELSE
142:         IF PA < 705.09 THEN PA:= 50125.7923315 - 79.0246642817 * PA
143:            + 0.030808325969 * PA * PA
144:         ELSE PA:= 36483.525204 - 57.4057011468 * PA
145:            + 1.51260276686E-2 * PA * PA;
146:      IF PA < 0 THEN C:= 7.13966666655 + 0.8264698 * PA + 1.35663E-5*PA*PA
147:      ELSE
148:         IF PA < 5000 THEN C:= -11.4763604151 + 1.40661336336E-2 * PA
149:            - 2.36456659555E-6 * PA * PA
150:         ELSE
151:            IF PA < 9000 THEN C:= -1175.00071 + 0.73075 * PA
152:                - 1.65866E-4 * PA * PA + 1.6224E-8 * PA * PA * PA
153:                -5.78465E-13 * PA * PA * PA;
154:         ELSE
155:            IF (PA > 9350) AND (PA < 9700) THEN
156:                C:= 1.18740544813E+5 - 38.10748 * PA + 4.0738E-3 * PA * PA
157:                - 1.45063E-7 * PA * PA * PA;
158:            ELSE
159:                C:= -1.27373891E+3 + 0.2686 * PA - 1.83266E-5 * PA * PA
160:                   + 4.05324E-10 * PA * PA * PA;
161:            PA:= PA+C
162:      END;  (* BP (>) 0 *)
163:  END;
164:  PROCEDURE GETPRESSUREFROMPALTITUDE;
165:  {
166:      (* CONVERT PRESSURE ALTITUDE TO PRESSURE *)
167:  BEGIN
168:      PA:= REALIN('Pressure Altitude (feet)? ', -1000.0, 10000.0);
169:  WRITELN;
170:      BP:= 1013.08968413 - 3.60310856232E-2 * PA + 4.38653185363E-7*PA*PA;
171:      C:= -0.1403422 + 5.58757E-4 * PA - 9.1921E-8 * PA * PA
172:                + 3.46176E-12 * PA * PA * PA;
173:  WRITELN;
174:      BP:= BP-C;
BP := BP / 33.8639 /* CONVERT MILLIBARS TO INCHES HG */
END; /* BF = 0 */

PROCEDURE PRINTDENSITYALTITUDE;

BEGIN
  HEADING;
  WRITELN;WRITELN;
  WRITE(' Station Elevation = 'pELEV:6:0); WRITELN (' feet');
  WRITE(' Temperature = ',TEMP:6:0); WRITELN (' degrees Farenheit');
  WRITE(' Pressure = ',BP:6:2); WRITELN (' inches of mercury');
  WRITE(' Pressure Altitude = ',PA:6:2); WRITELN (' feet');
  WRITELN;
  WRITE ('Your Density Altitude is ',DA:6:2); WRITELN (' feet');
END;

(* MAIN PROCEDURE *)

BEGIN
  DODENSITYALTITUDECALCULATIONS;
  WRITELN;WRITE('Want to calculate another density altitude? (Y or N) ');
  GETAYESNOANSWER(ANSWER);
  UNTIL ANSWER = 'N';
  ASSIGN(CHAINFIL,'BEES.COM'); /* chain back to main BEES menu */
  RESET(CHAINFIL);
  CHAIN(CHAINFIL)
END.
(* ************************************************************************* *
 * MODULE COPTER 
 * 
 * Programmer:  Susan R. Wheeler, Ga Tech/GTRI 
 * Source file:  COPTER.SRC 
 * Last modified:  7/24/84 
 * 
 * This module contains procedures to handle the helicopter model menu used 
 * by the Density Altitude program, and procedures to perform load capacity 
 * calculations for some of the helicopters on the menu. The rest of the 
 * helicopters had to be put into another file (NEWCOP.SRC) because this file 
 * exceeded the size limit of the SPP editor. 
 * 
 * The helicopter load capacity equations and parameters in this source file 
 * and in NEWCOP.SRC were extracted from the corresponding ETL BEES density 
 * altitude program. 
 * ************************************************************************* *)
 
MODULE COPTER;  (* revised 2/28/84 *)
VAR
   TEMP,PA,DA: EXTERNAL REAL;
   WT,MAXWT,GROSSWT,A,B,TOROUE,MAXTOROUE: REAL;
   Y,YSCALE,TEMP2,TEMP3,TEMP4,DA2,DA3,DA4: REAL;
   CRAFT,OIL,CREW: REAL;
   NCREW,HELNUM: INTEGER;
   HELTYPE: STRING;

EXTERNAL PROCEDURE GETMENUCHOICE(MX:INTEGER;VAR C:INTEGER); (* BEESLIB *)
EXTERNAL PROCEDURE MENU(NAME:STRING);    (* CURLIB *)
EXTERNAL PROCEDURE PAUSE;                 (* BEESLIB *)
EXTERNAL PROCEDURE CLRSCR;                (* CURLIB *)
EXTERNAL PROCEDURE HEADING;               (* DENSALT *)
EXTERNAL PROCEDURE CH54A;                 (* NEWCOP *)
EXTERNAL PROCEDURE CH54B;                 (* " *)
EXTERNAL PROCEDURE OH6A;                  (* " *)
EXTERNAL PROCEDURE OH58A;                 (* " *)
EXTERNAL PROCEDURE OH58C;                 (* " *)
EXTERNAL PROCEDURE UH1CM;                 (* " *)
EXTERNAL PROCEDURE UH1DH;                 (* " *)
EXTERNAL PROCEDURE UH60A;                 (* " *)

PROCEDURE SETUP_LOADCAPACITYCALCULATIONS;
BEGIN
   TEMP::: (TEMP - 32) * 5 / 9;  (* convert to centigrade *)
   TEMP2::: SQR (TEMP);
   TEMP3::: TEMP * TEMP2;
   TEMP4::: SQR (TEMP2);
   DA2::: SQR(DA);
   DA3::: DA * SQR(DA);
   DA4::: SQR(DA2);
   END;

PROCEDURE COMPUTE_ALOADCAPACITY;
BEGIN
   CLRSCR;
   MENU('HELI.MNU');  (* File HELI.MNU contains text of helicopter menu *)
61:  GETMENUCHOICE(14,HELM); (* 14 is max valid menu option number *)
62:  CASE HELNUM OF
63:    0: EXIT; (* 0 is 'quit the menu' *)
64:    1: AH_1G;
65:    2: AH_1S;
66:    3: CH_47A; (* Procedures containing calculations for the *)
67:    4: CH_47B; (* helicopters indicated *)
68:    5: CH_47C;
69:    6: CH54A;
70:    7: CH54B;
71:    8: UH1DH;
72:    9: OH6A;
73:   10: OH58A;
74:   11: OH58C;
75:   12: UH1CM;
76:   13: UH1DH;
77:   14: UH60A;
78:   END (* CASE *)
79:  END; (* load capacity *)
80:
81:  PROCEDURE OLOADCAPACITYCALCULATIONS;
82:  BEGIN
83:    SETUPLOADCAPACITYCALCULATIONS;
84:    REPEAT COMPUTEALOADCAPACITY UNTIL HELNUM=0
85:  END;
86:
87:  PROCEDURE PRINT_OUTPUT;
88:  BEGIN
89:    TEMP:= TRUNC (TEMP * 100) / 100;
90:    HEADING;
91:    WRITELN;WRITELN('HELICOPTER LOADING CAPABILITY');
92:    WRITELN ('HELICOPTER LOADING CAPABILITY');
93:    WRITELN ('------------------------------');WRITELN;
94:    WRITELN('Helicopter type = ',HELTYPE);
95:    WRITELN('OAT = ',TEMP:8:2, ' degrees Centigrade');
96:    WRITELN('Pressure Altitude = ',PA:8:2, ' feet');
97:    WRITELN('Density Altitude = ',DA:8:2, ' feet');
98:    WRITELN('Maximum allowable load (cargo, fuel, passenger mix) = ',
99:      TRUNC(WT):6:0, ' lbs (approx)');
100:   WRITELN('Allowable load assumes:');WRITELN;
101:   WRITELN('Aircraft basic weight = ',TRUNC(CRAFT):10:0, ' lbs');
102:   WRITELN('Oil = ',TRUNC(OIL):10:0, ' lbs');
103:   NCREW:=TRUNC(CREW);
104:    CREW:=CREW * 200;
105:    CREW:=CREW / 2;
106:    CREW:=CREW * 200;
107:   WRITELN('Crew(',NCREW:1,') = ',
108:     TRUNC(CREW):10:0, ' lbs');
109:   PAUSE
110:  END;
111:
112:  PROCEDURE AH_1G;
113:  BEGIN
114:    HELTYPE:= 'AH-1G (T53-L-13B)';
115:    CRAFT:=6067;
OIL := 24;
CREW := 1;
MAXWT := 9500;
MAXTORQUE := 50; (* PSI *)
A := 98872.516782 - 177.53792 * TEMP + 3.058016 * TEMP2
+ 0.1009999 * TEMP3 - 4.65033E-3 * TEMP4;
B := -23549.2176 - 5.53587 * TEMP - 1.18097 * TEMP2
- 0.0315638 * TEMP3 + 1.35519E-3 * TEMP4;
TORQUE := EXP((PA - A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE := MAXTORQUE;
YSCALE := 0.2 * TORQUE - 5;
GROSSWT := A + B * YSCALE;
IF GROSSWT > MAXWT THEN GROSSWT := MAXWT;
WT := GROSSWT - (CRAFT + OIL + CREW * 200);
PRINT_OUTPUT

PROCEDURE AH_1S:
BEGIN
HELTYPE := 'AH-1S (T53-L-703), TWO ENGINE OPERATION';
CRAFT := 6910;
OIL := 29;
CREW := 1;
MAXWT := 10000;
MAXTORQUE := 56; (* PSI *)
A := 85128.094 - 391.18 * TEMP + 6.933 * TEMP2 - 0.0974 * TEMP3
+ 9.4623E-4 * TEMP4;
B := -26178.3089 - 9.0711 * TEMP + 1.25465 * TEMP2 - 0.04209 * TEMP3
+ 3.8448E-4 * TEMP4;
TORQUE := EXP((PA - A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE := MAXTORQUE;
YSCALE := 0.2 * TORQUE - 5;
A := 5539.57727 - 4.67087E-3 * DA - 8.47021E-7 * SQR(DA);
B := 812.884 - 0.014 * DA;
GROSSWT := A + B * YSCALE;
IF GROSSWT > MAXWT THEN GROSSWT := MAXWT;
WT := GROSSWT - (CRAFT + OIL + CREW * 200);
PRINT_OUTPUT

END;

PROCEDURE CH_47A;
BEGIN
HELTYPE := 'CH-47A (T55-L-7/7B), TWO ENGINE OPERATION';
CRAFT := 17105;
OIL := 28;
CREW := 2;
MAXWT := 33000.0;
MAXTORQUE := 860; (* LB FT *)
A := 180909.633 - 100.4284 * TEMP - 9.0486 * TEMP2
+ 0.2776 * TEMP3 - 2.6125E-3 * TEMP4;
B := -26178.3089 - 9.0711 * TEMP + 1.25465 * TEMP2 - 0.04209 * TEMP3
+ 3.8448E-4 * TEMP4;
TORQUE := EXP((PA - A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE := MAXTORQUE;
YSCALE := 0.02 * TORQUE - 6;
181: \[ A := 17865.134 - 0.03225 \times DA + 6.6797E-6 \times DA^2 - 8.496E-10 \times DA^3; \]
182: \[ B := 2029.0 - 0.0399 \times DA - 2.127E-6 \times DA^2 + 2.382E-10 \times DA^3; \]
183: \[ \text{GROSSWT} := A + B \times YSCALE; \]
184: \[ \text{IF GROSSWT} > \text{MAXWT} \text{THEN GROSSWT} := \text{MAXWT}; \]
185: \[ \text{WT} := \text{GROSSWT} - (\text{CRAFT} + \text{OIL} + \text{CREW} \times 200); \]
186: \[ \text{PRINT_OUTPUT} \]
187: \[ \text{END}; \]
188: \[ \]
189: \[ \text{PROCEDURE CH\_47B;} \]
190: \[ \]
191: \[ \text{BEGIN} \]
192: \[ \text{HELTYP} := \text{\textasciitilde CH47B (T55-L-7B), TWO ENGINE OPERATION}; \]
193: \[ \text{CRAFT} := 19410; \]
194: \[ \text{OIL} := 28; \]
195: \[ \text{CREW} := 2; \]
196: \[ \text{MAXWT} := 40000.0; \]
197: \[ \text{MAXTORQUE} := 860; \text{(* LB FT *)}; \]
198: \[ \text{A} := 180909.633 - 100.4284 \times \text{TEMP} - 9.0486 \times \text{TEMP}^2 + 0.2776 \times \text{TEMP}^3 \]
199: \[ - 2.6125E-3 \times \text{TEMP}^4; \]
200: \[ \text{B} := - 26178.3089 - 9.0711 \times \text{TEMP} + 1.25465 \times \text{TEMP}^2 - 0.04209 \times \text{TEMP}^3 \]
201: \[ + 3.8448E-4 \times \text{TEMP}^4; \]
202: \[ \text{TORQUE} := \exp ((\text{PA} - \text{A}) / \text{B}); \]
203: \[ \text{IF TORQUE} > \text{MAXTORQUE} \text{THEN TORQUE} := \text{MAXTORQUE}; \]
204: \[ \text{YSCALE} := 0.01 \times \text{TORQUE} - 3; \]
205: \[ \text{IF TEMP} < 0 \text{THEN TEMP} := \text{ABS}(); \]
206: \[ \text{IF TEMP} > 10 \text{THEN BEGIN} \]
207: \[ \text{A} := 0; \]
208: \[ \text{B} := 1 \]
209: \[ \text{END} \]
210: \[ \text{ELSE BEGIN} \]
211: \[ \text{A} := -0.016543 + 6.9857E-4 \times \text{TEMP} + 9.64286E-5 \times \text{TEMP}^2; \]
212: \[ \text{B} := 0.98997 + 6.02381E-4 \times \text{TEMP} + 1.42857E-5 \times \text{TEMP}^2 + 2.8333E-6 \times \text{TEMP}^3 \]
213: \[ \text{END}; \]
214: \[ \text{Y} := \text{A} + \text{B} \times \text{YSCALE}; \]
215: \[ \text{A} := 18600.2289 - 4.61E-3 \times \text{DA} - 4.895E-6 \times \text{DA}^2; \]
216: \[ \text{B} := 3938.1889 - 0.07542786 \times \text{DA} + 1.1847E-6 \times \text{DA}^2; \]
217: \[ \text{GROSSWT} := \text{A} + \text{B} \times \text{Y}; \]
218: \[ \text{IF GROSSWT} > \text{MAXWT} \text{THEN GROSSWT} := \text{MAXWT}; \]
219: \[ \text{WT} := \text{GROSSWT} - (\text{CRAFT} + \text{OIL} + \text{CREW} \times 200); \]
220: \[ \text{PRINT_OUTPUT} \]
221: \[ \text{END}; \]
222: \[ \text{PROCEDURE CH\_47C;} \]
223: \[ \]
224: \[ \text{BEGIN} \]
225: \[ \text{HELTYP} := \text{\textasciitilde CH-47C (T55-L-7C), TWO ENGINE OPERATION}; \]
226: \[ \text{CRAFT} := 21791; \]
227: \[ \text{OIL} := 28; \]
228: \[ \text{CREW} := 2; \]
229: \[ \text{MAXWT} := 40000.0; \]
230: \[ \text{MAXTORQUE} := 990; \text{(* LB FT *)}; \]
231: \[ \text{A} := 1012.8261 - 0.039624 \times \text{PA} + 6.5447E-7 \times \text{SQR}(); \]
232: \[ \text{B} := - 7.22367 + 3.429E-4 \times \text{PA} + 7.6667E-9 \times \text{SQR}(); \]
233: \[ \text{TORQUE} := \text{A} + \text{B} \times \text{TEMP}; \]
234: \[ \text{IF TORQUE} > \text{MAXTORQUE} \text{THEN TORQUE} := \text{MAXTORQUE}; \]
235: \[ \text{YSCALE} := 0.0078 \times \text{TORQUE} - 3.12; \]
236: \[ \text{A} := -0.042277 - 4.1146E-4 \times \text{TEMP} + 9.07749E-5 \times \text{TEMP}^2 - 1.3535E-6 \times \text{TEMP}^3; \]
237: \[ \text{B} := 1.06055 - 3.278E-3 \times \text{TEMP} + 4.3958E-5 \times \text{TEMP}^2; \]
238: \[ \text{Y} := (\text{YSCALE} \times \text{A}) / \text{B}; \]
239: \[ \text{A} := 22070.8728 - 0.0336509 \times \text{DA} - 1.70867E-6 \times \text{DA}^2 + 3.01605E-10 \times \text{DA}^3; \]
241: \[ B = 4997.193 - 0.114 \times DA; \]
242: \[ \text{GROSSWT} = A + B \times Y; \]
243: \[ \text{IF GROSSWT} > \text{MAXWT} \text{ THEN GROSSWT} = \text{MAXWT}; \]
244: \[ \text{WT} = \text{GROSSWT} - (\text{CRAFT} + \text{OIL} + \text{CREW} \times 200); \]
245: \[ \text{PRINT_OUTPUT} \]
246: \[ \text{END;} \]
247: \[ \text{MODEND.} \]
MODULE COP2;

VAR

C: REAL;
TEMP, PA, DA: EXTERNAL REAL;
UT, MAXWT, GROSSWT, A, B, TORQUE, MAXTORQUE: EXTERNAL REAL;
Y, YSCALE, TEMP2, TEMP3, TEMP4, DA2, DA3, DA4: EXTERNAL REAL;
CRAFT, OIL, CREW: EXTERNAL REAL;
HELNUM: EXTERNAL INTEGER;
HELTYPE: EXTERNAL STRING;

EXTERNAL PROCEDURE PRINT_OUTPUT;

PROCEDURE CH_54A;
BEGIN
HELTYPE:='CH-54A (T73-P-1), TWO ENGINE OPERATION';
CRAFT:= 21895;
OIL:=15;
CREW:=2;
MAXWT:=42000.0;
MAXTORQUE:= 81.5; (* PERCENT *)
IF TEMP <= -11 THEN BEGIN
  A:=35613.1689 - 1.22378 * TEMP+ 0.56865 * TEMP2;
  B:= -296.125 - 0.166 * TEMP - 7.698E-3 * TEMP2
END
ELSE BEGIN
  A:=34787.769488 - 70.79067 * TEMP + 0.7604 * TEMP2;
  B:= -296.62895 - 0.2883 * TEMP - 0.03579 * TEMP2
END;
TORQUE:=(PA-A)/B;
IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;
YSCALE:= 0.05 * TORQUE - 1;
A:= 0.98398 + 4.568E-4 * TEMP - 2.9856E-6 * TEMP2;
B:= 0.935 + 1.2618E-3 * TEMP;
Y:= A * EXP (B * LN(YSCALE));
A:= 25925.7756 - 0.167786 * DA - 6.1351E-6 * DA2;
B:= 0.4199 - 7.528E-7 * DA - 5.4282E-11 * DA2;
GROSSWT:= A * EXP(B * LN(Y));
IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;
WT:= GROSSWT - (CRAFT + OIL + CREW * 200);
PRINT_OUTPUT
END;

PROCEDURE CH_54B;
BEGIN
CRAFT:= 22625;
OIL:= 15;
CREW:= 2;

HELTYPE:= 'CH-54B (T73-P-700), TWO ENGINE OPERATION';

MAXWT:= 47000.0;
MAXTORQUE:= 100.0; (* PERCENT *)

IF TEMP >= -14 THEN BEGIN
A:= 121998.47092 - 95.3369 * TEMP - 7.31065 * TEMP2;
B:= -26009.6487 - 30.9293 * TEMP + 0.84051 * TEMP2
+ 0.102388 * TEMP3 - 1.72256E-3 * TEMP4;
END;

ELSE BEGIN
A:= 94594.5894 - 3884.3305 * TEMP - 184.65234 * TEMP2
- 3.58638 * TEMP3 - 0.02472 * TEMP4;
B:= -21752.0191 + 473.1486 * TEMP + 16.41736 * TEMP2 + 0.17233 * TEMP3;
END;

TORQUE:= EXP((PA - A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;

YSCALE:= 0.05 * TORQUE - 1;

A:= 0.9892857 + 4.166667E-4 * TEMP - 1.387357E-5 * TEMP2 + 1.9733E-7 * TEMP3;
B:= 0.9479 + 1.71E-3 * TEMP - 1.3371E-5 * TEMP2;

C:= -1253.3072 + 0.03978 * DA + 1.0865E-6 * DA2 - 2.325E-10 * DA3;
GROSSWT:= A + B * Y + C * SQR(Y);

IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;
WT:= GROSSWT - (CRAFT + OIL + CREW * 200);

PRINT_OUTPUT
END;

PROCEDURE OH_6A;
BEGIN
END;

PROCEDURE OH_58A;
BEGIN
END;

PROCEDURE OH_6A;
BEGIN
CRAFT:= 1050;
OIL:= 6;
CREW:= 1;
HELTYPE:= 'OH-6A (T63-A-5A/-700)';
MAXWT:= 2550;
MAXTORQUE:= 80.3; (* PSIG *);

A:= 102996.723 - 304.236 * TEMP;
B:= -21965.0879998 + 34.059849213 * TEMP - 0.19067595237 * TEMP2
- 1.88977778529E-3 * TEMP3;

TORQUE:= EXP((PA - A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;

YSCALE:= 0.2 * TORQUE - 8;

A:= 1587.6971 - 0.016613 * DA + 4.7096E-7 * DA2;
B:= 134.8292 - 6.686E-5 * DA - 7.266E-8 * DA2;

GROSSWT:= A + B * YSCALE;

IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;

WT:= GROSSWT - (CRAFT + OIL + CREW * 200);

PRINT_OUTPUT
END;

PROCEDURE OH_58A;
BEGIN
CRAFT:= 1725;
OIL:= 13;
CREW:= 1;
121: HELTYPE:= 'OH-58A (T63-A-700)';
122: MAXWT:= 3000;
123: MAXTORQUE:= 78; (* PSI *)
124: A:= 100574.314 - 335.296 * TEMP;
125: B:= - 21976.59 + 33.793 * TEMP;
126: TORQUE:= EXP((PA-A) / B);
127: IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;
128: YSCALE:= 0.2 * TORQUE - 7;
129: A:= 1556.74286 - 4.4852E-4 * DA - 2.14579E-7 * DA2;
130: B:= 179.227 - 0.003 * DA;
131: GROSSWT:= A + B * YSCALE;
132: IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;
133: WT:= GROSSWT - (CRAFT + OIL + CREW * 200);
134: PRINT_OUTPUT
135: END;
136:
137: PROCEDURE OH_58C;
138:
139: BEGIN
140: CRAFT:= 1898;
141: OIL:= 11;
142: CREW:= 1;
143: HELTYPE:= 'OH-58C (T63 A-720)';
144: MAXWT:= 3200;
145: MAXTORQUE:= 85; (* PERCENT *)
146: A:= 111904.6343 - 333.287 * TEMP - 0.9386 * TEMP2;
147: B:= -22630.94 + 50.807 * TEMP;
148: TORQUE:= EXP((PA-A) / B);
149: IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;
150: YSCALE:= 0.1 * TORQUE - 5;
151: A:= 2012.5 - 5.05E-3 * DA - 1.05322E-6 * DA2 + 2.2958E-10 * DA3
152: - 8.3415E-15 * DA4;
153: B:= 367.6616 - 5.8745E-3 * DA - 4.7198E-7 * DA2;
154: GROSSWT:= A + B * YSCALE;
155: IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;
156: WT:= GROSSWT - (CRAFT + OIL + CREW * 200);
157: PRINT_OUTPUT
158: END;
159:
160: PROCEDURE UH_1CM;
161:
162: BEGIN
163: CRAFT:= 5156;
164: OIL:= 28;
165: CREW:= 1;
166: HELTYPE:= 'UH-1C/M (T53-L-13)';
167: MAXWT:= 9500;
168: MAXTORQUE:= 50; (* PSI *)
169: A:= 104798.2095 - 230.8065 * TEMP - 1.3629 * TEMP2;
170: B:= -24682.61 - 2.0857 * TEMP - 0.2726 * TEMP2 + 0.0108 * TEMP3;
171: TORQUE:= EXP((PA-A) / B);
172: IF TORQUE > MAXTORQUE THEN TORQUE:= MAXTORQUE;
173: YSCALE:= 0.4 * TORQUE - 12;
174: A:= 6084.1843 + 0.02774 * DA - 2.473E-6 * DA2;
175: B:= 437.239 - 0.018 * DA;
176: GROSSWT:= A + B * YSCALE;
177: IF GROSSWT > MAXWT THEN GROSSWT:= MAXWT;
178: WT:= GROSSWT - (CRAFT + OIL + CREW * 200);
179: PRINT_OUTPUT
180: END;
PROCEDURE UH_1DH;
BEGIN
CRAFT := 5320;
OIL := 27;
CREW := 1;
MAXWT := 9500;
MAXTORQUE := 50; (* PSI *)
IF HELNUM = 8 THEN HELTYPE := 'EH-1H (T53-L-13)';
ELSE HELTYPE := 'UH-1H (T53-L-13)';
A := 101054.187157 - 18.262435 * TEMP - 6.0301 * TEMP2 + 0.14874 * TEMP3
- 2.0619E-3 * TEMP4;
B := -24007.94261 - 48.213595 * TEMP + 1.35253 * TEMP2 - 0.0387325 * TEMP3
+ 4.4448E-4 * TEMP4;
TORQUE := EXP((DA-A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE := MAXTORQUE;
YSCALE := 0.2 * TORQUE - 4;
A := 4876.42675 + 9.1772E-3 * DA + 4.2057E-7 * DA2 - 3.2717E-12 * DA3
- 3.484E-15 * DA4;
B := -24007.94261 - 48.213595 * TEMP + 1.35253 * TEMP2 - 0.0387325 * TEMP3
+ 4.4448E-4 * TEMP4;
TORQUE := EXP((DA-A) / B);
IF TORQUE > MAXTORQUE THEN TORQUE := MAXTORQUE;
YSCALE := 0.2 * TORQUE - 4;
A := 133092.5625 + 646.745925 * TEMP + 8.2227 * TEMP2;
B := -27493.0 - 116.449945 * TEMP - 1.5191325 * TEMP2;
END;
PROCEDURE UH_60A;
BEGIN
CRAFT := 10984;
OIL := 15;
CREW := 2;
HELTYPE := 'UH-60A (T700), TWO ENGINE OPERATION';
MAXWT := 20250;
MAXTORQUE := 100; (* PERCENT *)
IF (TEMP <= -9) AND (PA < 15000) OR (TEMP <= -14) AND (PA < 15000)
THEN BEGIN
A := 133092.5625 + 646.745925 * TEMP + 8.2227 * TEMP2;
B := -27493.0 - 116.449945 * TEMP - 1.5191325 * TEMP2;
END
ELSE BEGIN
A := 114366.99834 - 166.88466 * TEMP - 28.12099 * TEMP2
+ 1.13862 * TEMP3 - 0.011 * TEMP4;
B := -23803.01534 + 11.427755 * TEMP + 6.567375 * TEMP2
- 0.27295 * TEMP3 + 2.70916E-3 * TEMP4;
END;
END;
BEGIN
.yy.
END;
BEGIN
.yy.
END;
BEGIN
.yy.
END;
PROGRAM RANREAD;

TYPE
  MOONDATA = RECORD (* record structure *)
    MONTHNAME: STRING[9]; (* for MOONRAN *)
    YEAR: INTEGER;
    RA: INTEGER; (* 'A' constant *)
    RD5: REAL; (* 'D5' constant *)
    WD: INTEGER;
    PHASEDATE: ARRAY[1..5] OF INTEGER;
    MONTHDATA: ARRAY[1..4,1..163] OF REAL;
  END;

VAR
  RF: FILE OF MOONDATA;
  ERR,II: INTEGER;

PROCEDURE MOVEDATATORANDOMFILE(AFILE: STRING;RFILE: STRING);

CONST
  END_FILE = $1A;

VAR
  SRC : TEXT;
  MONTH,WK,II: INTEGER;

BEGIN
  OPEN(SRC,AFILE,ERR); (* open sequential file *)

  IF ERR=255 THEN
  BEGIN
    WRITELN('ASCII input file ',AFILE,' does not exist.');
    EXIT
  END;

  WRITELN('Data will be moved from ASCII file ',AFILE,' to random file ',RFILE);

  OPEN(RF,RFILE,ERR); (* open random file *)

  (* Read data from the sequential file into the MOONDATA record. *)
  (* The record with its current contents is written to specified *)
  (* record locations in the random access output file. (*)
  (* The RF prefix indicates an element of the record. *)

  READLN(SRC,RF^.YEAR); (* These values appears just once in the *)
  READLN(SRC,RF^.RA); (* sequential file but will be in every *)
61: READLN(SRC,RF^.RD5); (* record of the random file. *)
62: (* Read data for 12 months from the sequential file and write a *)
63: (* random record after reading each monthly data set. The month *)
64: (* data part of the MOONDAT record is replaced by new data from *)
65: (* the sequential file before each write, but YEAR remains the *)
66: (* same. *)
67: (* same. *)
68: 69: FOR MONTH:= 1 TO 12 DO
70: BEGIN
71: 72: READLN(SRC,RF^.MONTHNAME);
73: WRITELN(' Moving data for ',RF^.MONTHNAME,',RF^.YEAR);
74: READLN(SRC,RF^.WD,RF^.PHASEDATE[1],RF^.PHASEDATE[2],RF^.PHASEDATE[3],
75: RF^.PHASEDATE[4],RF^.PHASEDATE[5]);
76: (* For reading in 64 array elements, a READ statement is more *)
77: (* convenient than READLN since you can avoid having to list *)
78: (* the input target variables explicitly. But you do have to *)
79: (* read a carriage return to go to the next line when it's *)
80: (* time to start reading data for a new month. *)
81: 82: FOR WEEK:= 1 TO 4 DO
83: BEGIN
84: FOR K:= 1 TO 16 DO
85: BEGIN
86: READ(SRC,RF^.MONTHDATA[WEAK,K]);
87: END (* FOR K *)
88: END;(* FOR WEEK *)
89: READLN(SRC); (* go to the next line *)
90: SEEKWRITE(RF,MONTH) (* random write record number 'MONTH' *)
91: END; (* FOR MONTH *)
92: CLOSE(RF,ERR) (* close random file *)
93: END; (* movedata *)
94:
95: BEGIN
96: MOVEDATATORANDOMFILE('MOONDAT','MOONRAN');
97: OPEN(RF,'MOONRAN',ERR); (* open random file *)
98: SEEKREAD(RF,11);
99: WRITELN(RF^.MONTHNAME);
100: FOR II:=1 TO 5 DO WRITELN(RF^.PHASEDATE[II]);
101: FOR II:= 1 TO 16 DO WRITELN(RF^.MONTHDATA[1,II])
102: END.
103:
(* ************************************************************************** *
* MODULE INMOON *
* Programmer:  Susan R. Wheeler, Ga Tech/GTRI *
* Source file:  INMOON.SRC *
* Last revised:  8/27/84 *
* *
* This the input module for moonrise/moonset times. *
* *)

MODULE INMOON;

VAR
  ANNUM, DAY, JDATE, MONTH: EXTERNAL INTEGER;
  LATM, LONGM: EXTERNAL INTEGER;
  LAT, MINUTES, LONG: EXTERNAL REAL;
  LATD, LONGD: EXTERNAL CHAR;
  MDAY: ARRAY [1..12] OF INTEGER;
  DEG: INTEGER;

EXTERNAL FUNCTION INTIN(PROMPT:STRING; MIN, MAX: INTEGER): INTEGER; (* BEESLIB *)
EXTERNAL PROCEDURE GET_NEXT_KEY_PRESSED(VAR C: CHAR); (* BEESLIB *)
EXTERNAL PROCEDURE PRINTMOONHEADING; (* MOON *)

PROCEDURE ASSIGN_MONTHS;

BEGIN
  MDAY[1] := 0; (* store Julian date for start *)
  MDAY[2] := 31; (* of the 12 months in array *)
  MDAY[5] := 120;
  MDAY[8] := 212;
  MDAY[10] := 273;
  MDAY[12] := 334;
END;

PROCEDURE GET_DATE_AND_LOCATION;

BEGIN
  ASSIGN_MONTHS;
  PRINTMOONHEADING;
  WRITELN; WRITELN('Enter DATE for moonrise/moonset times:');
  WRITELN;
  MONTH := INTIN(' MONTH (1-12): ',1,12);
  DAY := INTIN(' DAY (1-31): ',1,31);
  IF (ANNUM/4 = ANNUM DIV 4) AND (MONTH <= 2) THEN
      JDATE := DAY + MDAY[MONTH] + 1 (* Julian date *)
  ELSE
      JDATE := DAY + MDAY[MONTH];
  END;
  WRITELN; WRITELN('Enter LATITUDE of moonrise/moonset location:');
  DEG := INTIN(' Degrees (0-60): ',0,60);
  LATM := INTIN(' Minutes (0-60): ',0,60);
61: MINUTES := LATM;
62: LAT := DEC + MINUTES/60;
63: WRITE (' Direction from equator (N/S): ');
64: REPEAT GET_NEXT_KEY_PRESSED(LATD);
65: UNTIL LATD IN ['S','N'];
66: WRITE(LATD);
67: READLN;
68: IF LATD = 'S' THEN LAT := -1*LAT;
69: WRITELN;
70: WRITELN('Enter LONGITUDE for moonrise/moonset location:');
71: WRITELN;
72: DEG := INTIN(' Degrees (0-180): ',0,180);
73: LONGM := INTIN(' Minutes (0-60): ',0,60);
74: WRITE (' Direction from Greenwich (E/W): ');
75: REPEAT GET_NEXT_KEY_PRESSED(LONGD);
76: UNTIL LONGD IN ['E','W'];
77: WRITE(LONGD);
78: READLN;
79: MINUTES := LONGM;
80: LONG := DEC + MINUTES/60;
81: IF LONGD = 'W' THEN LONG := -1 *LONG;
82: END;
83: MODEND.
(* **************************************************************************
***** MODULE MOON

Programmer: Susan R. Wheeler, Georgia Tech/GTRI
Source file: MOON.SRC
Last modified: 8/31/84

This is the main module for the Moon Phenomena program which computes moonrise/moonset times for geographic locations between 60 deg N and 60 deg S.

Accuracy inside the allowable location range should be within 5 minutes.

This program requires an auxiliary data file to execute (see the programmer reference documentation for more information) which must be updated annually. Moonrise/moonset times are only calculated for the 'current year' as determined by the contents of the data file.

The Moon Phenomena program can also output dates for moon phases in months of the current year.

*************************************************************************)

PROGRAM TMOON;

TYPE
MOONDATA = RECORD
  MONTHNAME: STRING[9]; (* access file containing moon data *)
  YEAR: INTEGER; (* for the current year *)
  RA: INTEGER;
  RD5: REAL;
  WD: INTEGER;
  PHASEDATE: ARRAY[1..5] OF INTEGER;
  MONTHDATA: ARRAY[1..4,1..16] OF REAL;
END;

VAR
CHAINFIL: FILE;
FULL,NEW,OPTION,PHENOM,WANE,WAX: STRING[147];
DAYS,HOURS,LAT,LONG,MINUTES: REAL;
B1,B2,B3,B4,B5,B6,B7,D5,H0,H1,T0,T1,T2,W,X,X2: REAL;
GHA,DEC: REAL;
DELI,GHA0,TAU,TI: REAL;
TERM: ARRAY[1..16] OF REAL;
A,ANNUM,CHOICE,DAY,ERR,JDATE,LAST,LATM,LONM: INTEGER;
MONTH,ICOUNT,RISING,SETTING,TIME,WEEK: INTEGER;
LATD,LONGD: CHAR;
RF: FILE OF MOONDATA;
FILENOT_THERE: BOOLEAN;
ILINE: STRING[3];

EXTERNAL FUNCTION ACOS(ARG: REAL) : REAL; (* BCESLIB *)
EXTERNAL FUNCTION ASIN(ARG: REAL) : REAL; (* *)
EXTERNAL FUNCTION CHARTOINTEGER(LIN: STRING): INTEGER; (* *)
EXTERNAL FUNCTION DTR(ARG: REAL) : REAL; (* *)
EXTERNAL FUNCTION RMOD(ARG, MODF: REAL) : REAL; (* *)

(* Define record structure for random *)
(* access file containing moon data *)
(* for the current year *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
(* *)
EXTERNAL FUNCTION RTD(ARG: REAL) : REAL; (* " *)
EXTERNAL PROCEDURE GETMENUCHOICE(MAX:INTEGER;VAR C:INTEGER);(* " *)
EXTERNAL PROCEDURE CLEARSCR; (* CURLIB *)
EXTERNAL PROCEDURE GET_DATE_AND_LOCATION; (* INMOON *)
EXTERNAL PROCEDURE GET_NEXT_KEY_PRESSED(VAR C: CHAR); (* BEESLIB *)
EXTERNAL PROCEDURE GET_A_YEgNO(iiNSWER(VAR C: CHAR); (* BEESLIB *)
EXTERNAL PROCEDURE GETLOCALTIME(LONG: REAL); (* BEESLIB *)
EXTERNAL PROCEDURE GOTOXY(X,Y: BYTE); (* CURLIB *)
EXTERNAL PROCEDURE MENU(NAME:STRING); (* CURLIB *)
EXTERNAL PROCEDURE PAUSE; (* BEESLIB *)

PROCEDURE CHOOSE_PROGRAM_OPTION;
BEGIN
PRINT_MOON_HEADING;
MENU('MOON.MNU');
GETMENUCHOICE(2,CHOICE);
CASE CHOICE OF
  0: OPTION: 'QUIT';
  1: OPTION: 'RISE&SET';
  2: OPTION: 'PHASES';
END; (* case *)
END;

PROCEDURE FIND_DATA;
VAR I: INTEGER;
BEGIN
WEEK:= (DAY - 1) DIV 8; (* integer divide *)
WEEK:: WEEK + 1;
SEEKREAD(RF,MONTH); (* random read record for current month *)
FOR I:= 1 TO 16 DO TERM[I]:= RF^.MONTHDATEDAY[WEK, I];
END;

PROCEDURE GET_MONTH_AND_PRINT_PHASES;
VAR I: INTEGER;
BEGIN
PRINT_MOON_HEADING;
WRITELN;WRITELN;
REPEAT
  WRITE('Which month? (1-12) '); 
  READLN(ILINE);
  MONTH:: CHARTOINTEGER(ILINE)
UNTIL (MONTH )= 1) AND (MONTH (= 12);
SEEKREAD(RF,MONTH); (* random read, record # = MONTH *)
PRINT_MOON_HEADING;
GOTOXY(10,6);
WRITELN('Approximate moon phase dates for ',RF^.MONTHNAME, ',RF',YEAR);
WRITELN;WRITELN;
CASE RFA.WD OF
  1: WRITELN(WANE,NEW,WAX,FULL);
  2: WRITELN(NEW,WAX,FULL,WANE);
  3: WRITELN(WAX,FULL,WANE,NEW);
END;
PROCEDURE GHA_AND_DEC; (* compute moon GHA and dec for *)
(* current iteration *)
BEGIN
  W:= TRUNC((DAY - 1) / 8);
  W:= W * 8 + 1;
  (* series expansion *)
  X:= (DAY + T1 - W) / A - 1;
  B7:= X * TERM8;
  B6:= X * (TERM7 + B7);
  B5:= X * (TERM6 + B6);
  B4:= X * (TERM5 + B5);
  B3:= X * (TERM4 + B4);
  B2:= X * (TERM3 + B3);
  B1:= X * (TERM2 + B2);
  GHA:= RMOD((TERM1 + B1),360); (* moon Greenwich Hour Angle at time TO *)
  B7:= X * TERME16;
  B6:= X * (TERME15 + B7);
  B5:= X * (TERME14 + B6);
  B4:= X * (TERME13 + B5);
  B3:= X * (TERME12 + B4);
  B2:= X * (TERME11 + B3);
  B1:= X * (TERM10 + B2);
  DEC:= RMOD((TERM9 + B1),360); (* moon declination at time TO *)
END;

PROCEDURE ITERATE;
BEGIN
  X2:= (0.00233 - SIN(DTR(LAT)) * SIN(DTR(DEC)) / (COS(DTR(LAT)) * COS(DTR(DEC)));
  H1:= RTD (ACOS(X2)); (* H1 = approx to moon local hour angle *)
  IF TIME = RISING THEN H1:= 360 - H1;
  TAU:= (H1 - HO) / DELH; (* TAU = correction to 1st guess for phenom time *)
  IF TAU > 0.5 THEN TAU:= TAU - 360 / DELH
  ELSE
    IF TAU < 0.5 THEN TAU:= TAU + 360 / DELH;
  END;
  T2:= T0 + TAU;
  IF ABS(T2-T1) < 0.01 THEN PRINTOUTPUT (* computed time is good enough *)
  ELSE BEGIN
    ICOUNT:= ICOUNT + 1;
    IF ICOUNT > 12 THEN PRINT ..OUTPUT (* do no more than 12 iterations *)
  END;
  T1:=T2;
  GHAANDDEC;
  DELH:= (GHA - GHA0) / TAU; (* compute delh for iterations past the 1st *)
  IF DELH < 0 THEN DELH:= DELH + 360 / ABS(TAU);
ITERATE
PROCEDURE PHENOMENA_TIME_CALCULATIONS;

BEGIN
(* D5 = approximation to moon's daily rate of change in *)
(* GHA. The value to use on iteration 0 is a constant *)
(* which is updated annually. *)

SEEKREAD(RF,1);
D5:= RF"RD5; (* Get D5 off of record 1 in the moon data file. *)
ICOUNT:= 0;
TO:= (12 - LONG / 15) / 24; (* initial approximation to phenomenon time *)
T1:= TO;
GHAANDDEC;
DELH:= D5; (* initial approx daily rate of change in GHA *)
GHA0:= GHA; (* save GHA from iteration 0 *)
HO:= GHA0 + LONG; (* HO = moon local hour angle on iteration 0 *)
ITERATE;
END;

PROCEDURE PRINTINPUTS;
BEGIN
PRINTMOONHEADING;
WRITELN;
WRITE('Date: ',MONTH:2, '/', TRUNC(DAY):2:0, '/', RF"YEAR:4);
WRITELN;
WRITE(' ',ABS(TRUNC(LAT)):4:0, ' Degrees', LATM:3, ' Minutes ', LATD);
WRITE(' ', ABS(TRUNC(LONG)):4:0, ' Degrees', LONGM:3, ' Minutes ', LONGD);
GETLOCALTIME(LONG);
WRITELN;
END;

PROCEDURE PRINT_MOON_HEADING; (* clear screen and print a heading *)
(* for this program *)
BEGIN
CLRSCR;
WRITELN('MICROFIX BEES: MOON PHENOMENA');
WRITELN('------------------------------', '
' '--------');
WRITELN('------------------------------', '
' '--------');
END;

PROCEDURE PRINT_OUTPUT;
BEGIN
DAYS:= T2 + JDATE;
241:  HOURS:= (DAYS - TRUNC(DAYS)) * 24;
242:  MINUTES:= (HOURS - TRUNC(HOURS)) * 60;
243:  IF TIME = RISING THEN BEGIN
244:    PHENOM:= 'MOONRISE';
245:    WRITELN(' MOON RISE AND SET TIMES');
246:    WRITELN(' --------------------------');
247:    WRITELN;
248:    WRITELN(' JULIAN DATE ZULU TIME');
249:    WRITELN;
250:  END
251:  ELSE PHENOM:= 'MOONSET';
252:  IF ICOUNT < 13 THEN WRITELN(' 	 ' tPHENOM,TRUNC(DAYS):15:0,
253:                              TRUNC(HOURS):15:0,'C I TRUNC(MINUTES):3:0)
254:    ELSE WRITELN(' 	 ' I PHENOM, None expected. Check preceding ','
255:                              'and following dates.);
256:  IF TIME = SETTING THEN BEGIN
257:    WRITELN;
258:    WRITELN(' Values are accurate to + or - 5 minutes.');
259:    PAUSE
260:  END;
261:  END;
262:  END;
263:  END;
264:  PROCEDURE SETUP;
265:  BEGIN
266:    OPEN(RF,'MOONRAN',ERR); (* Open random moon data file *)
267:    IF ERR = 255 THEN FILE_NOT_THERE:= TRUE
268:    ELSE BEGIN
269:      FILE NOT THERE:: FALSE;
270:      SEEKREAD(RF,1);
271:      A:=RF".YEAR (* get year from the 1st record *)
272:      END;
273:    SEEKREAD(RF,1);
274:    A:=RF".RA; (* annual updated value from file *)
275:    WAX:= 'CRESCENT/WAX ';
276:    WANE:= 'CRESCENT/WANE ';
277:    FULL:= 'FULL MOON ';
278:    NEW:= 'NEW MOON '
279:    RISING:= 1;
280:    SETTING:= 2
281:    END;
282:  END;
283:  END;
284:  BEGIN
285:    SETUP;
286:    IF FILE_NOT_THERE THEN BEGIN
287:      WRITELN('Data file is missing.');
288:      WRITELN('Sorry, program must be terminated.');
289:    END
290:    ELSE REPEAT
291:      CHOOSE_PROGRAM_OPTION;
292:      IF OPTION='PHASES' THEN GET_MONTH_AND_PRINT_PHASE_DATES
293:      ELSE IF OPTION = 'RISE&SET' THEN BEGIN
294:        GET_DATE_AND_LOCATION;
295:        PRINTINPUTS;
296:        FIND_DATA;
297:        FOR TIME:= RISING TO SETTING DO PHEMOMENA_TIME_CALCULATIONS;
298:      END
301: RESET(CHAINFIL);
302: CHAIN(CHAINFIL) (* chain back to driver *)
303:
304: END.
305:
(* ***************************************************************************
 * MODULE INSUN
 * Programmer: Susan R. Wheeler, Ga Tech/EES
 * Source file: INSUN.SRC
 * Last revised: 8/27/84
 * This is the input module for the sun phenomena computation program. It solicits date and latitude/longitude/direction information from the user. The input information is available to the main module, as all variables indicated EXTERNAL to this module are declared in the main module SUNRISE.
 *********************************************************************** *)

MODULE INSUN;

VAR
LATM,LONGM,MONTH,DAY,YEAR,JDATE: EXTERNAL INTEGER;
LAT,MINUTES,LONG: EXTERNAL REAL;
LATD,LONGD: EXTERNAL CHAR;
MDAY: ARRAY [1..12] OF INTEGER;
DEG: INTEGER;

EXTERNAL FUNCTION INTIN(FROMSTRING:STRING;MAX,MIN:INTEGER):INTEGER; (* BEESLIB *)
EXTERNAL PROCEDURE GET_NEXT_KEY_PRESSED(VAR C: CHAR); (* BEESLIB *)

PROCEDURE ASSIGN_MONTHS;
BEGIN
MDAY[1]:: 0; (* Store Julian date for the beginning *)
MDAY[2]:: 31;
MDAY[3]:: 59;
MDAY[4]:: 90;
MDAY[5]:: 120;
MDAY[6]:: 151;
MDAY[7]:: 181;
MDAY[8]:: 212;
MDAY[9]:: 243;
MDAY[10]:: 273;
MDAY[11]:: 304;
MDAY[12]:: 334;
END;

PROCEDURE GET_INPUTS;
BEGIN
ASSIGN_MONTHS;
WRITELN;WRITELN('Enter DATE for sunrise/sunset times:');
WRITELN;
MONTH:: INTIN(' MONTH (1-12): ',1,12);
DAY:: INTIN(' DAY (1-31): '4,31);
YEAR:: INTIN(' YEAR (ex: 1983): ',0,9999);
IF (YEAR/4 = YEAR DIV 4) AND (MONTH > 2) THEN JDATE:= DAY + MDAYMONTH + 1 (* Julian date - leapyear *)
ELSE JDATE:= DAY + MDAY[MONTH]; (* or not leapyear *)
WRITELN;WRITELN('Enter LATITUDE of sunrise/sunset location:');
WRITELN;
DEG:: INTIN(' Degrees (0-60): ',0,60);

(* of the 12 months in array MDAY *)
(* Store Julian date for the beginning *)
(* of the 12 months in array MDAY *)
LATM := INTIN(' Minutes (0-60): ', 0, 60);
MINUTES := LATM;
LAT := DEC + MINUTES/60;
WRITE (' Direction from equator (N/S): ');
REPEAT GET_NEXT_KEY_PRESSED(LATD); (* Accept n,N,s, or S answer *)
UNTIL LATD IN ['S', 'N']; (* only. GETKEY converts lower *)
WRITE(LATD);
READLN;
IF LATD = 'S' THEN LAT := -1*LAT;
WRITELN;
WRITELN('Enter LONGITUDE for sunrise/sunset location:');
READLN;
DEG := INTIN(' Degrees (0-180): ', 0, 180);
LONGM := INTIN(' Minutes (0-60): ', 0, 60);
WRITE (' Direction from Greenwich (E/W): ');
REPEAT GET_NEXT_KEY_PRESSED(LONGD); (* Get a E or W as above *)
UNTIL LONGD IN ['E', 'W'];
WRITE(LONGD);
READLN;
MINUTES := LONGM;
LONG := DEG + MINUTES / 60;
IF LONGD = 'E' THEN LONG := -1 *LONG;
END;
MODEND.
PROGRAM SUNRISE;

VAR
MONTH, DAY, YEAR, JDATE: INTEGER;
LAT, MINUTES, LONG: REAL;
LATM, LONGM: INTEGER;
LATD, LONGD, ANSWER: CHAR;
T, MA, LA, RA, D, Z, H1, H2, T1, T2, SPAN: REAL;
S1, S2, H1, H2, M1, M2: INTEGER;
W: STRING;
CHAINFIL: FILE;

EXTERNAL FUNCTION ACOS(ARG: REAL) : REAL; (* BEESLIB *)
EXTERNAL FUNCTION ASIN(ARG: REAL) : REAL; (* " *)
EXTERNAL FUNCTION DTR(ARG: REAL) : REAL; (* " *)
EXTERNAL FUNCTION RMOD(ARG, MODF: REAL) : REAL; (* " *)
EXTERNAL FUNCTION RTD(ARG: REAL) : REAL; (* " *)
EXTERNAL PROCEDURE CLRSCR; (* CURLIB *)
EXTERNAL PROCEDURE GETLOCALTIME(LONG: REAL); (* BEESLIB *)
EXTERNAL PROCEDURE GET_INPUTS; (* INSUN *)
EXTERNAL PROCEDURE GET_A_YESNO_ANSWER(VAR C: CHAR); (* BEESLIB *)

PROCEDURE COMPUTE_PHENOMENA_TIMES; (* computation loop for four *)
(* sun phenomena *)

VAR
I: INTEGER;

BEGIN
FOR I := 1 TO 4 DO
BEGIN
IF I = 1 THEN BEGIN
Z := -0.0145439;
W := 'Sunrise and Sunset
END;
IF I = 2 THEN BEGIN

(* computation loop for four *)
(* sun phenomena *)
61: \[ Z := -0.1045285; \]
62: \[ W := 'Civil Twilight \]
63: END;
64: IF I = 3 THEN BEGIN
65: \[ Z := -0.2079117; \]
66: \[ W := 'Nautical Twilight \]
67: END;
68: IF I = 4 THEN BEGIN
69: \[ Z := -0.309017; \]
70: \[ W := 'Astronomical Twilight \]
71: END;
72: (* Compute sun's local hour angle. D is already in radians *)
73: (* Latitude and longitude must be changed to radians from degrees *)
74: \[ H := (Z - \sin(D) \times \sin(DTR(LAT))) / (\cos(D) \times \cos(DTR(LAT))); \]
75: IF ABS(H) = 1 THEN
76: BEGIN
77: \[ Hz := \text{ACOS}(H); \]
78: \[ Ti := (360 - \text{RTD}(H)) / 15 + RA - 0.06571 \times T - 6.622 + LONG / 15; \]
79: T1 := \text{RTR}(T1, 24); (* rising phenom's local time = T1 MOD 24 *)
80: T2 := (RTD(H) / 15 + RA - 0.06571 \times T - 6.622 + LONG / 15);
81: T2 := \text{RTR}(T2, 24); (* setting phenom local time *)
82: IF T1 < T2 THEN SPAN := ABS(T1 - T2);
83: ELSE SPAN := ABS(T2 + 24 - T1); (* time span computed *)
84: S1 := \text{TRUNC}(SPAN);
85: S2 := \text{TRUNC}((SPAN - \text{TRUNC}(SPAN)) \times 60);
86: H1 := \text{TRUNC}(T1);
87: H2 := \text{TRUNC}(T2);
88: M1 := \text{TRUNC}((T1 - \text{TRUNC}(T1)) \times 60);
89: M2 := \text{TRUNC}((T2 - \text{TRUNC}(T2)) \times 60);
90: (* printout for time of phenomena *)
91: WRITELN(W, H1:2, ':', M1:2, ' ZULU ', H2:2, ':', M2:2, ' ZULU ', S1:2, ' HOURS', S2:2, ' MINUTES');
92: (* FOR *)
93: ELSE (* printout for absence of phenomenon (/cos(h)/ ) 0 ) *)
94: WRITELN(W, ' Twilight lasts all night.' );
95: END; (* FOR *)
96: WRITELN(W, ' Values are accurate to within + or - 5 minutes.' );
97: WRITELN;
98: (* clear screen and print a heading *)
99: (* for this program *)
100: PRINT_SUN_HEADING; (* MICROFIX BEES: SUNRISE/SUNSET/TWILIGHT TIMES *)
101: (Valid for latitudes 60degN to 60degS ')
102: WRITE('----------------------------------------------', ' ')
103: WRITE('----- ');
104: WRITE('----------------------------------------------', ' ')
105: WRITE('----- ');
106: END;
107: PROCEDURE PRINT_SUN_HEADING; (* clear screen and print a heading *)
108: (* for this program *)
109: BEGIN
110: CLRSCR;
111: WRITE( ' MICROFIX BEES: SUNRISE/SUNSET/TWILIGHT TIMES ' ) ;
112: WRITE( ' (Valid for latitudes 60degN to 60degS)' ) ;
113: WRITE('----------------------------------------------', ' ')
114: WRITE('----- ');
115: WRITE('----------------------------------------------', ' ')
116: WRITE('----- ');
117: END;
118:
119:
120: PROCEDURE Find_right_ascension_and_declination;
BEGIN
T := JDATE + (6 + LONG/15) / 24; (* approx days since 0 Jan, 0 hrs UT *)
MA := 0.9856 * T - 3.289; (* sun's mean anomaly *)

(* Compute sun's true longitude *)
(* Arguments of sin and cos must be converted to radians *)
L := (MA + 1.916 * SIN(DTR(MA)) + 0.02 * SIN(DTR(2*MA)) + 282.634);
L := RMOD(L,360); (* MOD 360 *)
IF L = 180 THEN L := 179.999999
ELSE IF L = 270 THEN L := 269.999999;

(* Compute sun's right ascension (RA) *)
RA := ARCTAN(0.91746 * SIN(DTR(L))/COS(DTR(L)));
RA := RTD (RA); (* Convert radian output from ARCTAN to degrees *)

(* Place RA in same quadrant as L *)
IF (L ) 90) AND (L < 180) THEN RA := 90 - ABS(RA) + 90;
IF (L )= 180) AND (L < 270) THEN RA := ABS(RA) + 180;
IF L ) 270 THEN RA := 90 - ABS(RA) + 270;
RA := RA/15; (* Convert hours to degrees *)
D := 0.39782 * SIN(DTR(L));
D := ASIN(D); (* Sun's declination = arcsin(D) *)
END; (* Find RA and D *)

PROCEDURE GET_DATE_AND_LOCATION; (* Get inputs from user and print *)
(* preliminary output. The times of *)
(* phenomena are printed as they are *)
(* computed. *)
BEGIN
PRINT SUN HEADING;
GET_INPUTS;
PRINT_SUN_HEADING;
WRITELN;
WRITE(' Date: ',MONTH:2,'/',TRUNC(DAY):2:0,'/',YEAR:4);
WRITE(' Latitude: ',ABS(TRUNC(LAT)):4:0,' Degrees',LATM:3,' Minutes ',LATD);
WRITE(' Longitude: ',ABS(TRUNC(LONG)):4:0,' Degrees',LONGM:3,' Minutes ',LONGD);
GETLOCALTIME(LONG);
WRITELN;
END;

(* MAIN PROCEDURE *)
BEGIN
REPEAT
GET_DATE_AND_LOCATION;
FIND_RIGHT_ASCENSION_AND_DECLINATION;
END;
COMPUTE_PHENOMENA_TIMES;
WRITE('Do you want sunrise/sunset for another location? (Y/N) ');
GET_A_YESNO_ANSWER(ANSWER)
UNTIL ANSWER = 'N';
ASSIGN(CHAINFIL,'BEES.COM'); (* chain back to main menu *)
RESET(CHAINFIL);
CHAIN(CHAINFIL)
END(CHAINFIL)
Section 4

UTILITIES
Section 5

TECHNICAL ITEMS
Almanac for Computers
1984

Nautical Almanac Office
United States Naval Observatory
Washington, D. C. 20390
### Table of Contents

#### Explanation
- Introduction ........................................... A1
- Calendar .............................................. A2
- Explanation of Navigational Tables (Section C) .......... A5
- Explanation of Astronomical Tables (Section D) ......... A8
- Explanation of Stellar Tables (Section E) ............... A13

#### Applications
- Introduction .......................................... B1
- Day of the Year ...................................... B1
- Julian Date .......................................... B2
- Sidereal Time ........................................ B3
- Hour Angles ......................................... B4
- Altitude and Azimuth ................................. B4
- Sunrise, Sunset & Twilight ......................... B5
- Solar Coordinates ..................................... B8
- Equation of Time and Time of Solar Transit .......... B8
- Moonrise and Moonset ................................ B9
- Polaris ................................................. B11
- Equation of Position Line ......................... B12
- Motion of Body and Motion of Observer .............. B12
- Sextant Altitude Corrections ....................... B13
- Dip of Horizon ...................................... B13
- Coriolis Correction ................................ B14
- Atmospheric Refraction ................................ B14
- Semidiameter of Sun and Planets .................. B16
- Semidiameter of Moon ................................ B16
- Parallax in Altitude ................................ B16
- Correction for the Phase of Venus .................. B17

#### Navigational Tables
- Aries, Sun and Navigational Planets .................. C2
- Moon .................................................. C8

#### Astronomical Tables
- Full Precision:
  - Sidereal Time, Equation of Equinoxes, Nutation ...... D2
  - Sun ................................................ D6
  - Moon .............................................. D8
  - Mercury and Venus ................................ D20
  - Mars and Jupiter .................................. D24
  - Saturn and Uranus ................................. D26
  - Neptune and Pluto ................................ D28
- Low Precision:
  - Sidereal Time, Equation of Equinoxes, Nutation ...... D30
  - Sun ................................................ D30
  - Mercury and Venus ................................ D31
  - Mars, Jupiter, Saturn and Uranus ................... D32
  - Neptune and Pluto ................................ D33

#### Stellar Tables
- Index of the 57 Standard Navigation Stars ........... E2
- 176 Stars ............................................ E3
Section A: EXPLANATION

Introduction (with material that should be read)

Incorporated in A/C 84 are a number of changes from previous editions. Most obvious is the use of the day of the month rather than day of the year as the time argument in the navigation tables (Section C). To facilitate this feature the power series for lunar coordinates now cover spans of eight rather than six days. The price to be paid is that eight rather than six terms are required for the power series. In Section B formulas involving geographic longitudes have been revised so that longitudes east of Greenwich are positive and west longitudes are negative. This conforms to the convention that has been common in many fields and has been newly adopted in astronomy. In Section E apparent places of the stars are now computed from mean places for the middle rather than the beginning of the year. But since this change is accounted for in adjusted values of the constants, no reprogramming should be necessary.

The almanacs for 1984 are based on a new system of planetary ephemerides, precession and nutation expressions, time scale (dynamical time replacing ephemeris time), standard epoch (J2000.0 replacing 1950.0 and 1900.0), equinox (FK5 replacing FK4), and definition of astrometric position. This new system forms the basis for A/C 84. Information on the new system is given in The Astronomical Almanac 1984, the Supplement to the Astronomical Almanac 1984, and U. S. Naval Observatory Circular 163.

For most efficient use with computers, the data in Sections D and E are available on magnetic tape (data in Section C are not available in machine readable form). Inquiries about this service, as well as comments and suggestions concerning this volume, should be addressed to The Director, Nautical Almanac Office, U. S. Naval Observatory, Washington, D. C. 20390.

This volume was produced by LeRoy E. Doggett and Patrick E. Protacio, Ensign, USN.
### CALENDAR, 1984

<table>
<thead>
<tr>
<th>JANUARY</th>
<th>FEBRUARY</th>
<th>MARCH</th>
<th>APRIL</th>
<th>MAY</th>
<th>JUNE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day of Month</td>
<td>Day of Week</td>
<td>Day of Year</td>
<td>Day of Week</td>
<td>Day of Year</td>
<td>Day of Week</td>
</tr>
<tr>
<td>Sun. 1</td>
<td>Wed. 32</td>
<td>Thu. 61</td>
<td>Sun. 92</td>
<td>Tue. 122</td>
<td>Fri. 153</td>
</tr>
<tr>
<td>Mon. 2</td>
<td>Thu. 33</td>
<td>Fri. 62</td>
<td>Mon. 93</td>
<td>Wed. 123</td>
<td>Sat. 154</td>
</tr>
<tr>
<td>Tue. 3</td>
<td>Fri. 34</td>
<td>Sat. 63</td>
<td>Tue. 94</td>
<td>Thu. 124</td>
<td>Sun. 155</td>
</tr>
<tr>
<td>Wed. 4</td>
<td>Sat. 35</td>
<td>Sun. 64</td>
<td>Wed. 95</td>
<td>Fri. 125</td>
<td>Mon. 156</td>
</tr>
<tr>
<td>Thu. 5</td>
<td>Sun. 36</td>
<td>Mon. 65</td>
<td>Thu. 96</td>
<td>Sat. 126</td>
<td>Tue. 157</td>
</tr>
<tr>
<td>Fri. 6</td>
<td>Mon. 37</td>
<td>Tue. 66</td>
<td>Fri. 97</td>
<td>Sun. 127</td>
<td>Wed. 158</td>
</tr>
<tr>
<td>Sat. 7</td>
<td>Tue. 38</td>
<td>Wed. 67</td>
<td>Sat. 98</td>
<td>Mon. 128</td>
<td>Thu. 159</td>
</tr>
<tr>
<td>Sun. 8</td>
<td>Wed. 39</td>
<td>Thu. 68</td>
<td>Sun. 99</td>
<td>Tue. 129</td>
<td>Fri. 160</td>
</tr>
<tr>
<td>Mon. 9</td>
<td>Thu. 40</td>
<td>Fri. 69</td>
<td>Mon. 100</td>
<td>Wed. 130</td>
<td>Sat. 161</td>
</tr>
<tr>
<td>Tue. 10</td>
<td>Fri. 41</td>
<td>Sat. 70</td>
<td>Tue. 101</td>
<td>Thu. 131</td>
<td>Sun. 162</td>
</tr>
<tr>
<td>Wed. 11</td>
<td>Sat. 42</td>
<td>Sun. 71</td>
<td>Wed. 102</td>
<td>Fri. 132</td>
<td>Mon. 163</td>
</tr>
<tr>
<td>Thu. 12</td>
<td>Sun. 43</td>
<td>Mon. 72</td>
<td>Thu. 103</td>
<td>Sat. 133</td>
<td>Tue. 164</td>
</tr>
<tr>
<td>Fri. 13</td>
<td>Mon. 44</td>
<td>Tue. 73</td>
<td>Fri. 104</td>
<td>Sun. 134</td>
<td>Wed. 165</td>
</tr>
<tr>
<td>Sat. 14</td>
<td>Tue. 45</td>
<td>Wed. 74</td>
<td>Sat. 105</td>
<td>Mon. 135</td>
<td>Thu. 166</td>
</tr>
<tr>
<td>Sun. 15</td>
<td>Wed. 46</td>
<td>Thu. 75</td>
<td>Sun. 106</td>
<td>Tue. 136</td>
<td>Fri. 167</td>
</tr>
<tr>
<td>Mon. 16</td>
<td>Thu. 47</td>
<td>Fri. 76</td>
<td>Mon. 107</td>
<td>Wed. 137</td>
<td>Sat. 168</td>
</tr>
<tr>
<td>Tue. 17</td>
<td>Fri. 48</td>
<td>Sat. 77</td>
<td>Tue. 108</td>
<td>Thu. 138</td>
<td>Sun. 169</td>
</tr>
<tr>
<td>Wed. 18</td>
<td>Sat. 49</td>
<td>Sun. 78</td>
<td>Wed. 109</td>
<td>Fri. 139</td>
<td>Mon. 170</td>
</tr>
<tr>
<td>Thu. 19</td>
<td>Sun. 50</td>
<td>Mon. 79</td>
<td>Thu. 110</td>
<td>Sat. 140</td>
<td>Tue. 171</td>
</tr>
<tr>
<td>Fri. 20</td>
<td>Mon. 51</td>
<td>Tue. 80</td>
<td>Fri. 111</td>
<td>Sun. 141</td>
<td>Wed. 172</td>
</tr>
<tr>
<td>Sat. 21</td>
<td>Tue. 52</td>
<td>Wed. 81</td>
<td>Sat. 112</td>
<td>Mon. 142</td>
<td>Thu. 173</td>
</tr>
<tr>
<td>Sun. 22</td>
<td>Wed. 53</td>
<td>Thu. 82</td>
<td>Sun. 113</td>
<td>Tue. 143</td>
<td>Fri. 174</td>
</tr>
<tr>
<td>Mon. 23</td>
<td>Thu. 54</td>
<td>Fri. 83</td>
<td>Mon. 114</td>
<td>Wed. 144</td>
<td>Sat. 175</td>
</tr>
<tr>
<td>Tue. 24</td>
<td>Fri. 55</td>
<td>Sat. 84</td>
<td>Tue. 115</td>
<td>Thu. 145</td>
<td>Sun. 176</td>
</tr>
<tr>
<td>Wed. 25</td>
<td>Sat. 56</td>
<td>Sun. 85</td>
<td>Wed. 116</td>
<td>Fri. 146</td>
<td>Mon. 177</td>
</tr>
<tr>
<td>Thu. 26</td>
<td>Sun. 57</td>
<td>Mon. 86</td>
<td>Thu. 117</td>
<td>Sat. 147</td>
<td>Tue. 178</td>
</tr>
<tr>
<td>Fri. 27</td>
<td>Mon. 58</td>
<td>Fri. 87</td>
<td>Fri. 118</td>
<td>Sun. 148</td>
<td>Wed. 179</td>
</tr>
<tr>
<td>Sat. 28</td>
<td>Tue. 59</td>
<td>Wed. 88</td>
<td>Sat. 119</td>
<td>Mon. 149</td>
<td>Thu. 180</td>
</tr>
<tr>
<td>Sun. 29</td>
<td>Wed. 60</td>
<td>Thu. 89</td>
<td>Sun. 120</td>
<td>Tue. 150</td>
<td>Fri. 181</td>
</tr>
<tr>
<td>Mon. 30</td>
<td>Fri. 90</td>
<td>Mon. 121</td>
<td>Wed. 151</td>
<td>Sat. 182</td>
<td></td>
</tr>
<tr>
<td>Tue. 31</td>
<td>Sun. 91</td>
<td>Sat. 152</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### JULIAN DATE, 1984

<table>
<thead>
<tr>
<th>0° UT</th>
<th>0° UT</th>
<th>0° UT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan.</td>
<td>0</td>
<td>244 5699.5</td>
</tr>
<tr>
<td>Feb.</td>
<td>0</td>
<td>244 5730.5</td>
</tr>
<tr>
<td>Mar.</td>
<td>0</td>
<td>244 5759.5</td>
</tr>
</tbody>
</table>

400-day date: JD 244 6000.5 = 1984 October 27 0
## Calendar, 1984

<table>
<thead>
<tr>
<th>Day of Month</th>
<th>Day of Week</th>
<th>Day of Year</th>
<th>Day of Week</th>
<th>Day of Year</th>
<th>Day of Week</th>
<th>Day of Year</th>
<th>Day of Week</th>
<th>Day of Year</th>
<th>Day of Week</th>
<th>Day of Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>JULY</td>
<td>AUGUST</td>
<td>SEPTEMBER</td>
<td>OCTOBER</td>
<td>NOVEMBER</td>
<td>DECEMBER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Sun.</td>
<td>Wed. 214</td>
<td>Sat. 245</td>
<td>Mon. 275</td>
<td>Thu. 306</td>
<td>Sat. 336</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Mon.</td>
<td>Thu. 215</td>
<td>Sun. 246</td>
<td>Tue. 276</td>
<td>Fri. 307</td>
<td>Sun. 337</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Tue.</td>
<td>Fri. 216</td>
<td>Mon. 247</td>
<td>Wed. 277</td>
<td>Sat. 308</td>
<td>Mon. 338</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Wed.</td>
<td>Sat. 217</td>
<td>Tue. 248</td>
<td>Thu. 278</td>
<td>Sun. 309</td>
<td>Tue. 339</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Thu.</td>
<td>Sun. 218</td>
<td>Wed. 249</td>
<td>Fri. 279</td>
<td>Mon. 310</td>
<td>Wed. 340</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Fri.</td>
<td>Mon. 219</td>
<td>Thu. 251</td>
<td>Sat. 281</td>
<td>Wed. 312</td>
<td>Fri. 342</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Sat.</td>
<td>Tue. 220</td>
<td>Fri. 252</td>
<td>Sun. 282</td>
<td>Thu. 313</td>
<td>Sat. 343</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Sun.</td>
<td>Wed. 221</td>
<td>Sat. 253</td>
<td>Tue. 283</td>
<td>Fri. 314</td>
<td>Sun. 344</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 Mon.</td>
<td>Thu. 222</td>
<td>Sun. 254</td>
<td>Wed. 284</td>
<td>Sat. 315</td>
<td>Mon. 345</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Tue.</td>
<td>Fri. 223</td>
<td>Mon. 255</td>
<td>Thu. 285</td>
<td>Sun. 316</td>
<td>Tue. 346</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Wed.</td>
<td>Sat. 224</td>
<td>Tue. 255</td>
<td>Fri. 286</td>
<td>Mon. 317</td>
<td>Wed. 347</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Thu.</td>
<td>Sun. 225</td>
<td>Wed. 256</td>
<td>Fri. 287</td>
<td>Tue. 318</td>
<td>Thu. 348</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 Fri.</td>
<td>Mon. 226</td>
<td>Thu. 257</td>
<td>Sat. 287</td>
<td>Wed. 319</td>
<td>Fri. 349</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 Sat.</td>
<td>Tue. 227</td>
<td>Fri. 258</td>
<td>Sun. 288</td>
<td>Thu. 320</td>
<td>Sat. 350</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 Sun.</td>
<td>Wed. 228</td>
<td>Sat. 259</td>
<td>Mon. 289</td>
<td>Fri. 321</td>
<td>Sun. 351</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Mon.</td>
<td>Thu. 229</td>
<td>Sun. 260</td>
<td>Tue. 290</td>
<td>Fri. 322</td>
<td>Mon. 352</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 Tue.</td>
<td>Fri. 230</td>
<td>Mon. 261</td>
<td>Wed. 291</td>
<td>Sat. 323</td>
<td>Tue. 353</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 Wed.</td>
<td>Sat. 231</td>
<td>Tue. 262</td>
<td>Thu. 292</td>
<td>Sun. 324</td>
<td>Wed. 354</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 Thu.</td>
<td>Sun. 232</td>
<td>Wed. 263</td>
<td>Fri. 293</td>
<td>Mon. 325</td>
<td>Thu. 355</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 Fri.</td>
<td>Mon. 233</td>
<td>Thu. 264</td>
<td>Sat. 294</td>
<td>Tue. 326</td>
<td>Fri. 356</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 Sat.</td>
<td>Tue. 234</td>
<td>Fri. 265</td>
<td>Sun. 295</td>
<td>Wed. 327</td>
<td>Sat. 357</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22 Sun.</td>
<td>Wed. 235</td>
<td>Sat. 266</td>
<td>Mon. 296</td>
<td>Thu. 328</td>
<td>Sun. 358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23 Mon.</td>
<td>Thu. 236</td>
<td>Sun. 267</td>
<td>Tue. 297</td>
<td>Fri. 329</td>
<td>Mon. 359</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 Tue.</td>
<td>Fri. 237</td>
<td>Mon. 268</td>
<td>Wed. 298</td>
<td>Sat. 330</td>
<td>Tue. 360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 Wed.</td>
<td>Sat. 238</td>
<td>Tue. 269</td>
<td>Thu. 299</td>
<td>Sun. 331</td>
<td>Wed. 361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 Thu.</td>
<td>Sun. 239</td>
<td>Wed. 270</td>
<td>Fri. 300</td>
<td>Mon. 332</td>
<td>Thu. 362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27 Fri.</td>
<td>Mon. 240</td>
<td>Thu. 271</td>
<td>Sat. 301</td>
<td>Tue. 333</td>
<td>Fri. 363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 Sat.</td>
<td>Tue. 241</td>
<td>Fri. 272</td>
<td>Sun. 302</td>
<td>Wed. 334</td>
<td>Sat. 364</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29 Sun.</td>
<td>Wed. 242</td>
<td>Sat. 273</td>
<td>Mon. 303</td>
<td>Thu. 335</td>
<td>Sun. 365</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 Mon.</td>
<td>Thu. 243</td>
<td>Sun. 274</td>
<td>Tue. 304</td>
<td>Fri. 336</td>
<td>Mon. 366</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Mean Sidereal Time, 1984

Greenwich mean sidereal time at 0h UT

<table>
<thead>
<tr>
<th>Jan.</th>
<th>06:59:06</th>
<th>Apr.</th>
<th>0:57:02</th>
<th>July</th>
<th>18:54:98</th>
</tr>
</thead>
</table>

A – Explanation: Calendar
Navigational Tables

Section C contains mathematical representations of the following functions that are tabulated in the *Nautical Almanac* (NA): the GHA of Aries, the GHA and declination of the Sun, Moon and navigational planets, the semidiameter of the Sun and Moon, and the horizontal parallax of the Moon. Except in the case of the Moon, these functions are expressed for a specified time span by a power series of the form

\[
f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5
\]

For the Moon there are two additional terms, \(a_6 x^6\) and \(a_7 x^7\), that must be added to the series. In the series \(x\) is a time-like variable that takes on values between \(-1\) and \(+1\) over the specified time span; \(a_0, a_1, a_2, \ldots\), are coefficients that are tabulated in Section C for the specified time span; and \(f(x)\) represents the value of the function (e.g., the GHA of Aries) evaluated at time \(x\).

To evaluate the series for one of the navigational functions, one must first find the set of coefficients in Section C that is applicable for the desired date. Constants \(A\) and \(W\) are given for the purpose of converting the calendar date and GMT to the time-like variable \(x\). First compute \(t\), the GMT measured in days and fractions thereof from the beginning of the month: \(t = d + \text{GMT}/24\), where \(d\) is the day of the month at Greenwich and GMT is the Greenwich Mean Time expressed in hours. A calendar is provided on pages A2—A3. Once \(t\) has been determined, \(x\) can be computed from the relation \(x = (t - W)/A - 1\). If computed correctly, the value of \(x\) will fall in the range \(-1 \leq x \leq +1\).

Example 1: Compute \(x\) for later use in computing the position of the Moon on 24 July at 22h 30m 16s GMT (=049376852).

\(t = 24 + 0.9376852\). Constants for this date are found on page C14: \(A = 4\) and \(W = 17\). Therefore

\[x = (24.9376852 - 17)/4 - 1 = +0.9844213.\]

Once the variable \(x\) has been computed and the coefficients \(a_i\) have been found, the series for the desired function can be evaluated. The series can be evaluated most efficiently by computing a set of auxiliary variables, \(b_1, b_2, b_3, b_4, b_5\) (with additional variables \(b_6, b_7\) for the Moon), in the following order:

For the Sun, Aries, and planets:  \[b_5 = xa_5\]

Then for all objects:

\[
\begin{align*}
    b_4 &= x(a_4 + b_5) \\
    b_3 &= x(a_3 + b_4) \\
    b_2 &= x(a_2 + b_3) \\
    b_1 &= x(a_1 + b_2) \\
    f(x) &= a_0 + b_1
\end{align*}
\]

For the Moon:

\[
\begin{align*}
    b_7 &= xa_7 \\
    b_6 &= x(a_6 + b_7) \\
    b_5 &= x(a_5 + b_6)
\end{align*}
\]
By using this algorithm, the series is evaluated in its nested form. For the Sun, Aries and the planets, which have six coefficients per series, the nested series may be written in the form

\[ f(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + x(a_4 + xa_5)))). \]

Example 2: Compute the declination of the Moon at 22h 30m 16s GMT on 24 July 1984.

From the previous example \(x = +0.9844213\). The coefficients for the Moon's declination are found on page C14.

\[
\begin{align*}
b_7 &= 0.9844213 (+0.0021) = +0.0021 \\
b_6 &= 0.9844213 (+0.0183 + 0.0021) = +0.0201 \\
b_5 &= 0.9844213 (-0.1435 + 0.0201) = -0.1215 \\
b_4 &= 0.9844213 (-0.3371 - 0.1215) = -0.4515 \\
b_3 &= 0.9844213 (-1.8210 - 0.4515) = -2.2371 \\
b_2 &= 0.9844213 (-1.5186 - 2.2371) = -3.6972 \\
b_1 &= 0.9844213 (+20.3310 - 3.6972) = +16.3747 \\
f(+0.9844213) &= 7.2656 + 16.3747 = +23.6403 \\
\text{Therefore declination} &= +23^\circ 38'4 \\
\end{align*}
\]

Example 3: Compute the Sun's GHA at 15h 12m 10s GMT on 31 August 1984.

The constants \(A\) and \(W\) and the series coefficients are found on page C4.

\[
\begin{align*}
x &= (t - W)/A - 1 = (31.6334491 - 1)/16 - 1 = +0.9145906 \\
b_5 &= 0.9145906 (+0.0016) = +0.0015 \\
b_4 &= 0.9145906 (-0.0035 + 0.0015) = -0.0018 \\
b_3 &= 0.9145906 (-0.0334 - 0.0018) = -0.0322 \\
b_2 &= 0.9145906 (+0.2680 - 0.0322) = +0.2157 \\
b_1 &= 0.9145906 (+5760.8525 + 0.2157) = +5269.0188 \\
f(+0.9145906) &= 5938.9845 + 5269.0188 = +11289.0033 \\
\text{GHA} &= 48^\circ 00'33 = 48^\circ 00'2 \\
\end{align*}
\]

Note that when computing the GHA, it may be necessary to reduce the final result to the range 0°–360° by subtracting multiples of 360°.
Although the series are designed to provide precision comparable to that published in the *NA*, there will be small discrepancies between the tabulated values and the values computed from the series. In such cases it should be understood that the *NA* represents the standard. Table 1 lists the largest discrepancies found from evaluating and comparing the series with the data in the *NA*.

Under no circumstances should the series be used to extrapolate data beyond the specified time intervals. Such extrapolation will lead to erroneous and useless results.

In accordance with standard practice for navigational almanacs, the time argument used in this almanac is Greenwich Mean Time (GMT), or more specifically UT1. To obtain full precision in the determined positions, the radio time signals in UTC must be corrected to UT1, or GMT, according to standard procedures. (See the paper by R.L. Duncombe and P.K. Seidelmann, 'The New UTC Time Signals', *Navigation*, 24, 160–165, 1977.)

Beneath each set of coefficients in Section C is printed the sum of the coefficients. As a check on whether the coefficients have been entered accurately into the calculator, it is recommended that the coefficients be summed and that the resulting sum be compared with the printed sum.

Table 1: Comparison of Almanac for Computers with *NA*

<table>
<thead>
<tr>
<th>Function</th>
<th>No. of Terms</th>
<th>Span of Validity</th>
<th>Maximum Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHA of Aries</td>
<td>6</td>
<td>32 days</td>
<td>0.2</td>
</tr>
<tr>
<td>Sun:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GHA</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Semidiameter</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Moon:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GHA</td>
<td>8</td>
<td>8 days</td>
<td>0.2</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Horizontal Parallax</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Semidiameter</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
<tr>
<td>Navigational Planets:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GHA</td>
<td>6</td>
<td>32 days</td>
<td>0.1</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Astronomical Tables

Section D contains mathematical representations of data published in the Astronomical Almanac ($A^2$). Chebyshev expansions have been chosen as the means of representation since they provide efficient and accurate expressions that can be easily evaluated with a small computer. The coefficients $a_i$ of the Chebyshev expansion

$$f(x) = a_0/2 + \sum_{i=1}^{n} a_i T_i(x)$$

are tabulated for prescribed time spans, where $f(x)$ is the function being represented, $T_i(x)$ is the Chebyshev polynomial of the first kind of the $i$-th degree, and $x$ is the normalized time variable. Although Chebyshev polynomials appear in the series expansions, the series can be evaluated without explicitly computing these polynomials. No a priori knowledge of Chebyshev analysis is required to use the series in this almanac. Interested readers can find information on Chebyshev analysis in Applied Analysis by C. Lanczos and Chebyshev Polynomials in Numerical Analysis by L. Fox and I. B. Parker.

It must be emphasized that the series are valid only over the specified time intervals. Attempts to extrapolate data using these series will yield erroneous and useless results.

If precision comparable to that of the $A^2$ is required, the series on pages D2-D29 should be used. With the exception of the series for the Moon, these series are valid for time spans of approximately three months; for the Moon the span of validity is approximately one month. Table 2 lists the largest errors found by evaluating these series and comparing the results with data printed in the $A^2$.

It is possible to develop series that are valid for longer time spans if the precision requirements are relaxed. Such series, valid for one full year, are given on pages D30–D33. Precision criteria of these less precise series are summarized in Table 3.

To evaluate a Chebyshev series, one must first normalize the time variable on the interval for which the series is valid. The normalized time $x$ can be determined from the relation $x = (t - W)/A - 1$, where $t$ is reckoned in days and fractions thereof from 0 January. As in previous editions, constants $A$ and $W$ are given for each set of coefficients. If correctly computed, the value of $x$ will fall in the range $-1 \leq x \leq +1$.

For the functions Apparent Sidereal Time at 0 UT, Equation of the Equinoxes, Nutation in Longitude and Nutation in Obliquity, the variable $t$ is measured in days of universal time (UT1 to be precise) from 0 January, 0 UT. For all other functions in Section D, $t$ is measured in days of terrestrial dynamical time (TDT) from 0 January, 0 UT. These latter functions can be evaluated for universal times, however, using the normalizing relation $x = ((t' + \Delta T) - W)/A - 1$, where $t'$ is the universal time measured in days from 0 January, 0 UT. As this volume goes to
Table 2: Comparison of *Almanac for Computers* and *A²*  
(High Precision Series, pp. D2-D29)

<table>
<thead>
<tr>
<th>Function</th>
<th>No. of Terms</th>
<th>Span of Validity</th>
<th>Maximum Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apparent Sidereal Time at 0h UT</td>
<td>34</td>
<td>95 days</td>
<td>0:001</td>
</tr>
<tr>
<td>Equation of the Equinoxes</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:001</td>
</tr>
<tr>
<td>Nutation in Longitude</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:02</td>
</tr>
<tr>
<td>Nutation in Obliquity</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:01</td>
</tr>
<tr>
<td>Sun:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>22</td>
<td>95 days</td>
<td>0:02</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:1</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>4x10⁻⁷ AU</td>
</tr>
<tr>
<td>Semidiameter</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:01</td>
</tr>
<tr>
<td>Ephemeris Transit</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:01</td>
</tr>
<tr>
<td>Moon:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>32</td>
<td>32 days</td>
<td>0:04</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:3</td>
</tr>
<tr>
<td>Horizontal Parallax</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:01</td>
</tr>
<tr>
<td>Geocentric Rectangular Coords.</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁶ Earth radii</td>
</tr>
<tr>
<td>Mercury:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>32</td>
<td>95 days</td>
<td>0:08</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:3</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁶ AU</td>
</tr>
<tr>
<td>Venus:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>32</td>
<td>95 days</td>
<td>0:02</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:6</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁶ AU</td>
</tr>
<tr>
<td>Mars:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>16</td>
<td>95 days</td>
<td>0:03</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:2</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁶ AU</td>
</tr>
<tr>
<td>Jupiter:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>16</td>
<td>95 days</td>
<td>0:03</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:2</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁶ AU</td>
</tr>
<tr>
<td>Saturn:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>14</td>
<td>95 days</td>
<td>0:02</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:2</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>3x10⁻⁶ AU</td>
</tr>
<tr>
<td>Uranus:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>14</td>
<td>95 days</td>
<td>0:1</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:2</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>3x10⁻⁶ AU</td>
</tr>
<tr>
<td>Neptune:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>12</td>
<td>95 days</td>
<td>0:03</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:4</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁵ AU</td>
</tr>
<tr>
<td>Pluto:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension (astrometric)</td>
<td>12</td>
<td>95 days</td>
<td>0:004</td>
</tr>
<tr>
<td>Declination (astrometric)</td>
<td>&quot;</td>
<td>&quot;</td>
<td>0:04</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>&quot;</td>
<td>1x10⁻⁵ AU</td>
</tr>
</tbody>
</table>
press, $\Delta T = 54.2 (=0^d 0000627)$ appears to be a reliable value to use in 1984. Care should be taken to verify that the sum $t' + \Delta T$ falls within the time span for which the series is valid; if it falls outside, the series and constants for the next span should be used.

Once the normalized time variable $x$ is determined, the series can be evaluated as follows:

Let $b_{n+1} = b_{n+2} = 0$,

compute $b_i = 2xb_{i+1} - b_{i+2} + a_i$, for $i=n,n-1,\ldots,0$,

then $f(x) = (b_0 - b_2)/2$.

Example: Compute the equation of the equinoxes to a precision of $\pm 0.05$ at $14^h 51^m 37^s$ UT ($=0^d 619178$) on 26 January 1984.

As shown in Table 3 the low precision series on page D30 provide the required precision. Since universal time is the independent variable for the series for the equation of the equinoxes,

$t = 26^d + 0^d 619178 = 26^d 619178$

Constants for the series are $A = 183.0$ and $W = 1$.

$x = (26.619178 - 1)/183.0 - 1 = -0.860004$

$\begin{align*}
  b_{n+2} &= b_{11} = 0 \\
  b_{n+1} &= b_{10} = 0 \\
  b_n &= b_9 = 2xb_{10} - b_{11} + a_9 = +0.0029 \\
  b_8 &= 2xb_9 - b_{10} + a_8 = +0.0042 \\
  b_7 &= 2xb_8 - b_9 + a_7 = -0.0321 \\
  b_6 &= 2xb_7 - b_8 + a_6 = +0.0335 \\
  b_5 &= 2xb_6 - b_7 + a_5 = +0.0275 \\
  b_4 &= 2xb_5 - b_6 + a_4 = -0.0601 \\
  b_3 &= 2xb_4 - b_5 + a_3 = +0.0787 \\
  b_2 &= 2xb_3 - b_4 + a_2 = -0.0557 \\
  b_1 &= 2xb_2 - b_3 + a_1 = +0.0572 \\
  b_0 &= 2xb_1 - b_2 + a_0 = -1.9023 \\
\end{align*}$

$f(x) = (b_0 - b_2)/2 = (-1.9023 + 0.0557)/2$

equation of the equinoxes = $0.92$

Beneath each set of coefficients is printed the sum of the coefficients. This may be used as an easy means of verifying the accuracy with which the coefficients have been entered in the computer.

The series for Apparent Sidereal Time are designed to reproduce the table 'Apparent Sidereal Time at 0th Universal Time' in *The Astronomical Almanac*. To compute the Greenwich apparent sidereal time for any universal time,
Table 3: Comparison of *Almanac for Computers* and $A^2$
(Low Precision Series, pp. D30-D33)

<table>
<thead>
<tr>
<th>Function</th>
<th>No. of Terms</th>
<th>Maximum Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apparent Sidereal Time at 0h UT</td>
<td>10</td>
<td>0'03</td>
</tr>
<tr>
<td>Equation of the Equinoxes</td>
<td>&quot;</td>
<td>0'03</td>
</tr>
<tr>
<td>Nutation in Longitude</td>
<td>&quot;</td>
<td>0'5</td>
</tr>
<tr>
<td>Nutation in Obliquity</td>
<td>&quot;</td>
<td>0'3</td>
</tr>
<tr>
<td>Sun:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>22</td>
<td>0'5</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>3&quot;</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-6} AU</td>
</tr>
<tr>
<td>Semi-diameter</td>
<td>&quot;</td>
<td>0'04</td>
</tr>
<tr>
<td>Ephemeris Transit</td>
<td>&quot;</td>
<td>0'5</td>
</tr>
<tr>
<td>Mercury:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>50</td>
<td>8°</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>2'</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-4} AU</td>
</tr>
<tr>
<td>Venus:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>50</td>
<td>0'05</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>0'7</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>1x10^{-6} AU</td>
</tr>
<tr>
<td>Mars:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>34</td>
<td>0'8</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>5&quot;</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-5} AU</td>
</tr>
<tr>
<td>Jupiter:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>34</td>
<td>0'1</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>0'2</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-5} AU</td>
</tr>
<tr>
<td>Saturn:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension</td>
<td>16</td>
<td>0'1</td>
</tr>
<tr>
<td>Declination</td>
<td>&quot;</td>
<td>0'6</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-5} AU</td>
</tr>
<tr>
<td>Uranus:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension (astrometric)</td>
<td>16</td>
<td>0'2</td>
</tr>
<tr>
<td>Declination (astrometric)</td>
<td>&quot;</td>
<td>0'3</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>4x10^{-5} AU</td>
</tr>
<tr>
<td>Neptune:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension (astrometric)</td>
<td>12</td>
<td>0'1</td>
</tr>
<tr>
<td>Declination (astrometric)</td>
<td>&quot;</td>
<td>0'5</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>7x10^{-5} AU</td>
</tr>
<tr>
<td>Pluto:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Ascension (astrometric)</td>
<td>12</td>
<td>0'03</td>
</tr>
<tr>
<td>Declination (astrometric)</td>
<td>&quot;</td>
<td>0'3</td>
</tr>
<tr>
<td>Distance</td>
<td>&quot;</td>
<td>7x10^{-5} AU</td>
</tr>
</tbody>
</table>
(1) evaluate the series for the desired UT,
(2) add the desired UT to the result of step (1).

Local apparent sidereal time may be obtained by adding the local longitude to the
Greenwich apparent sidereal time, where east longitudes are considered positive.

With two exceptions the series in Section D provide data referred to the true
equinox and equator of date. The exceptions are
(1) the Moon's geocentric, rectangular coordinates \((X, Y, Z)\), which are referred
to the mean equator and equinox of B1950.0;
(2) the right ascension and declination of Pluto, which are astrometric (i.e.,
free of the effect of stellar aberration) and are referred to the mean equi-
nox and equator of J2000.0.

The unit of distance for the Sun and planets is the astronomical unit; the unit of
distance for the Moon is the Earth's equatorial radius.
Stellar Tables

The Stellar Tables (Section E) list the mean and apparent places of 176 stars for the current year, along with coefficients for converting from mean to apparent place for any date in the year. The selection of stars is essentially that of the star tables on pages 268–273 of the *Nautical Almanac*. Stars are arranged in order of increasing right ascension (decreasing sidereal hour angle), except where both components of a binary system are listed. For binary stars that can be resolved in small instruments, the position of one or both components is listed rather than the position of the center of gravity or the center of light. When both components of a binary system are included, the brighter star is listed first. For convenience of navigators the sidereal hour angle (SHA) is tabulated rather than right ascension (RA); right ascension in degrees can be obtained from the relation

\[ RA = 360^\circ - SHA. \]

The quantities tabulated for each star are, from left to right on the page:

1. Identification number.
2. Navigational star number, provided the star is one of the 57 selected navigational stars listed in the *Nautical Almanac* and *Air Almanac*.
3. Star name. The Bayer designation is on the first line and the proper name, if any, is on the second line.
4. Magnitude and spectral type. The visual magnitude is on the first line, and the spectral type is on the second line. A composite spectrum is denoted by *.
5. Mean place of the star for J1984.5. The SHA in degrees is on the first line; the declination in degrees is on the second line.
6. Four coefficients \((H, R, S, C)\) used in computing the apparent place of the star. The coefficients on the first line are for the computation of apparent SHA; these will hereafter be designated \(H_s, R_s, S_s, C_s\). The coefficients on the second line are for the computation of apparent declination; these will be designated with the subscript \(D: H_D, etc.\)
7. The sum of the mean SHA or declination and the coefficients in the line. This may be used to verify that the numbers have been entered correctly in the computer.

The mean place of a star is a fundamental reference point with no simple geometric or observational significance. The apparent place of a star is the geocentric position, referred to the true equinox and equator of date, at which the star is observed. Thus the apparent place is the position needed for navigation, calibration of telescope setting circles, computation of transit times, etc. Except for Polaris the tabulated mean places for the middle of the year can be used to an accuracy of \(\pm 1^\circ.3\) for any date during the year. To obtain apparent places to greater accuracy, the following procedures should be used:
For the desired date in the current year determine \( \tau \), the fraction of the year elapsed. If \( t \) denotes the day of the year, \( \tau \) can be computed from the relation 
\[
\tau = \frac{(t - W)}{A}
\]
As in previous editions the constants \( A \) and \( W \) are given at the top of page E3.

Except for Polaris, star positions accurate to better than \( \pm 0.5 \) can be obtained from the following formulas:

\[
\begin{align*}
\text{apparent SHA} & = \text{mean SHA} + H_S + R_S \tau \\
\text{apparent decl.} & = \text{mean decl.} + H_D + R_D \tau 
\end{align*}
\]

Except for Polaris, star positions accurate to better than \( \pm 0.1 \) (and usually better than \( \pm 0.05 \)) can be obtained from the following formulas:

\[
\begin{align*}
\text{apparent SHA} & = \text{mean SHA} + H_S + R_S \tau + S_S \sin (360^\circ \tau) + C_S \cos (360^\circ \tau) \\
\text{apparent decl.} & = \text{mean decl.} + H_D + R_D \tau + S_D \sin (360^\circ \tau) + C_D \cos (360^\circ \tau)
\end{align*}
\]

To facilitate identification of the 57 standard navigational stars, an index for these stars is provided on page E2.

Example: Compute the apparent place of Regulus (\( \alpha \) Leonis) on 5 August to an accuracy of \( \pm 0.1 \).

From the calendar on pages A2-A3 or the formulas on pages B1-B2, 6 October is found to be day 218. From page E3, \( A = 366.0 \) and \( W' = 184 \). Data for Regulus (Nav. No. 26; A/C ID 74) are found on page E6.

\[
\tau = \frac{(218 - 184)}{366.0} = +0.0929
\]

\[
\begin{array}{c|c|c}
\text{Mean place} & \text{SHA} & \text{decl.} \\
\hline
208^\circ 1131 & +12^\circ 0434 \\
+H & +0.0039 & +0.0020 \\
+R \tau & -0.0013 & -0.0004 \\
+S \sin (360^\circ \tau) & +0.0023 & +0.0008 \\
+C \cos (360^\circ \tau) & +0.0029 & +0.0011 \\
\hline
\text{Apparent place} & 208^\circ 121 & +12^\circ 047 \\
\end{array}
\]

Because of the close proximity of Polaris to the north celestial pole, a small change in the position of Polaris on the celestial sphere causes a large change in the value of the SHA (or right ascension). This is purely due to the nature of the coordinate system rather than to extraordinary physical motion. Though the formulas given above will yield the declination of Polaris to an accuracy comparable to that of other stars, errors in SHA can reach \( \pm 1^\circ 2 \), even if the more accurate formula is used.
Section B: APPLICATIONS

Introduction

In this section reference will be made to the following functions:

Sign function. The sign function serves to extract the algebraic sign from a number. The notation sign(x) is defined to be \( \text{sign}(x) = 1 \) for \( x \geq 0 \), sign(x) = -1 for \( x < 0 \). An equivalent definition is \( \text{sign}(x) = x / |x| \) for \( x \neq 0 \), \( \text{sign}(x) = 1 \) for \( x = 0 \).

Examples: \( \text{sign}(247) = 1 \), \( \text{sign}(-6.28) = -1 \).

Truncation or largest-integer function. The truncation function extracts the integral part of a number. The algebraic sign of the result is the same as that of the original number. \( \langle x \rangle \) is defined to be \( \langle x \rangle = \text{sign}(x) \cdot N \), where \( N \) is the largest nonnegative integer such that \( N \leq |x| \).

Examples: \( \langle 17.835 \rangle = 17 \), \( \langle -3.1416 \rangle = -3 \).

Modulus or remainder function. The modulus function yields the remainder of division \( x/y \), when the quotient is constrained to be an integer value. Thus \( \text{mod}(x,y) \) is defined to be \( \text{mod}(x,y) = x - \langle x/y \rangle \cdot y \).

Examples: \( \text{mod}(11,3) = 2 \), \( \text{mod}(-764.3,360.0) = -44.3 \).

Note that \( \langle x \rangle = x - \text{mod}(x,1) \). Therefore the truncation function can be defined in terms of the modulus function and vice versa. If either modulus or truncation is available on a calculator or computer, the other function can be simply obtained.

In this almanac universal time (UT) is to be identified with UT1, which is equivalent to the standard navigational time argument Greenwich Mean Time (GMT). The symbols UT and GMT may therefore be considered interchangeable. For detailed information on time systems the reader should consult the Explanation of a current edition of The Astronomical Almanac.

Day of the Year

The day of the year (N) is defined as the integer \( N = \langle t \rangle \), where \( t \) is the time elapsed in days since 0 January of the current year. Thus \( N \) is an integer running from 1 through 365 (or 366 in leap years). The day of the year can be computed from either of the following formulas:

\[
N = \left\lfloor \frac{275M}{9} \right\rfloor + \left\lfloor \frac{M+9}{12} \right\rfloor \left( 1 + \left\lfloor \frac{K - 4\{K/4\} + 2}{3} \right\rfloor \right) - I - 30
\]

\[
N = \left\lfloor \frac{275M}{9} \right\rfloor + \left\lfloor \frac{M+9}{12} \right\rfloor \left( 1 + \left\lfloor \frac{\text{mod}(K,4) + 2}{3} \right\rfloor \right) - I - 30
\]

where \( N \) is the day of the year, \( K \) is the year (e.g., 1981), \( M \) is the month (1 \( \leq M \leq 12 \)), and \( I \) is the day of the month (1 \( \leq I \leq 31 \)).

These formulas are equivalent and are valid for any year, except those centennial years that are not evenly divisible by 400. Therefore the formulas given above are
Sidereal Time

The following formulas are relevant to the computation of sidereal time:

1. \( \text{GMST} = 6 \text{h} 5905966 + 0 \text{h} 0657098242 \, N + 1.00273791 \, UT \)
2. \( \text{GMST} = 6 \text{h} 69737456 + 2400 \text{h} 051336 \, T_0 + 0 \text{h} 0000258622 \, T_0^2 + 1.002737909 \, UT \)
3. \( \Omega = 74 \text{h} 5658 - 0 \text{h} 0529539 \, (N + UT/24) \)
4. \( \Omega = 125 \text{h} 01452 - 1924 \text{h} 13626 \, T + 0 \text{h} 002074 \, T^2 \)
5. \( E = -0 \text{h} 00029 \sin \Omega \)
6. \( \text{GAST} = \text{GMST} + E \)
7. \( \text{GAST} = \Sigma(t_0) + 1.002737909 \, UT = \Sigma(t) + UT \)
8. \( \text{LAST} = \text{GAST} + \lambda/15 \)

where

- \( \text{GMST} \) is the Greenwich mean sidereal time in hours;
- \( \Omega \) is the mean longitude of the ascending node of the Moon's orbit, measured in degrees;
- \( E \) is the equation of the equinoxes in hours;
- \( \text{GAST} \) is the Greenwich apparent sidereal time in hours;
- \( \text{LAST} \) is the local apparent sidereal time in hours;
- \( N \) is the day of the year (1 \( \leq N \leq 365 \) or, during a leap year, 1 \( \leq N \leq 366 \));
- \( T_0 \) and \( T \) are time intervals in Julian centuries from J2000.0:
  \( T_0 = (JDO - 2451545.0)/36525 \)
  \( T = (JD - 2451545.0)/36525 \)
- \( UT \) is the universal time in hours;
- \( JDO \) and \( JD \) are the Julian Dates at 0h UT and at an arbitrary time of the day, respectively;
- \( \Sigma(t_0) \) and \( \Sigma(t) \) are values obtained by evaluating the Chebyshev series for Apparent Sidereal Time (pp. D2-D5 or D30) at 0h UT and at an arbitrary time of the day, respectively; (see page A12 for notes about evaluating the series for sidereal time);
- \( \lambda \) is the local longitude in degrees (east is positive; west is negative).

When using the formulas given above, it may be necessary to reduce the results to the range 0h -- 24h by adding or subtracting multiples of 24h.

Formulas (1) and (3) are specifically for the current year; the other formulas are valid at least over the latter half of this century. Formula (5) is an approximation that is accurate to about ±0°2. If more accuracy is required, the Chebyshev series for the Equation of the Equinoxes (pp. D2-D5 or D30) can be used in place of Formula (5). If sidereal time \( \Omega \) is to be computed to an accuracy better than ±0°2 (rarely justified for practical applications), then either the Chebyshev series for the Equation of the Equinoxes should be used in place of Formula (5) or Formula (7) should be used in place of Formula (6).
Hour Angles

The following formulas are useful if astronomical data, such as that given in Sections C and E, are applied to navigational purposes:

- \( \text{GHA} = 15 (\text{GAST} - \text{RA}) \)
- \( \text{LHA} = 15 (\text{LAST} - \text{RA}) = \text{GHA} + \lambda \)
- \( \text{GHA Aries} = 15 \text{GAST} \)
- \( \text{SHA} = 360^\circ - 15 \text{RA} \)
- \( \text{GHA} = \text{GHA Aries} + \text{SHA} \)

where

- \( \text{GHA} \) is the Greenwich hour angle in degrees;
- \( \text{LHA} \) is the local hour angle in degrees;
- \( \text{GHA Aries} \) is the Greenwich hour angle of the First Point of Aries (the origin of right ascension) in degrees;
- \( \text{SHA} \) is the sidereal hour angle in degrees;
- \( \text{RA} \) is the apparent right ascension (referred to the true equator and equinox of date) in hours;
- \( \lambda \) is the local longitude in degrees (east is positive; west is negative);
- \( \text{GAST} \) is the Greenwich apparent sidereal time in hours;
- \( \text{LAST} \) is the local apparent sidereal time in hours.

When using the above formulas, it may be necessary to add or subtract \(360^\circ\) to reduce the resulting hour angles to the range \(0^\circ - 360^\circ\). Often the local hour angle values are reduced to the range \(-180^\circ\) to \(+180^\circ\), in which case they are called meridian angles. In all cases positive hour angle values are measured westward from the meridian.

Altitude and Azimuth

The following formulas can be used to compute the altitude \((a)\) and azimuth \((A)\) of a celestial body:

1. \( \sin a = \cos z = \sin \phi \sin \delta + \cos \phi \cos \delta \cos \text{LHA} \)
2. \( x = \tan A = \sin \text{LHA} / (\cos \text{LHA} \sin \phi - \tan \delta \cos \phi) \)

Since computers and calculators normally give the arctangent in the range \(-90^\circ\) to \(+90^\circ\), the correct quadrant for \(A\) can be selected according to the following rules:

If \(0^\circ \leq \text{LHA} \leq 180^\circ\),
\[ A = 180^\circ + \arctan x, \text{ if } x \text{ is positive,} \]
\[ A = 360^\circ + \arctan x, \text{ if } x \text{ is negative.} \]

If \(180^\circ \leq \text{LHA} < 360^\circ\),
\[ A = \arctan x, \text{ if } x \text{ is positive,} \]
\[ A = 180^\circ + \arctan x, \text{ if } x \text{ is negative.} \]
Notation:

\[ a = \text{altitude of body above (if } \sin a > 0 \text{) or below (if } \sin a < 0 \text{) the horizon;} \]
\[ A = \text{azimuth of body measured eastward from north over the range } 0^\circ < A < 360^\circ; \]
\[ \phi = \text{latitude of observer (north is positive; south is negative);} \]
\[ \delta = \text{declination of body (north is positive; south is negative);} \]
\[ \text{LHA} = \text{local hour angle of body;} \]
\[ z = \text{zenith distance of body (} z = 90^\circ - a). \]

In standard navigational notation altitude and azimuth are denoted \( He \) and \( Zo \), respectively. Equations (1) and (2) are the basic formulas used in preparing sight reduction tables; they do not include the effect of refraction.

Example: Compute the altitude and azimuth of the Sun at 17h 58m 45s UT on 24 May 1984 at Annapolis, Maryland.

Latitude: \( \phi = +38^\circ 59 \)  \( \sin \phi = +0.62374 \)  \( \cos \phi = +0.78163 \)

Longitude: \( \lambda = -76^\circ 30 \)

Using the power series on page C4, the Sun’s GHA and \( \delta \) are found to be

\[ \text{GHA} = 90^\circ 483 \quad \text{hence} \quad \text{LHA} = 90^\circ 483 - 76^\circ 30 = 14^\circ 183 \]

\[ \sin \text{LHA} = +0.24502 \quad \cos \text{LHA} = +0.96952 \]

\[ \delta = +20^\circ 899 \quad \sin \delta = +0.35672 \quad \cos \delta = +0.93421 \quad \tan \delta = +0.38184 \]

\[ \sin a = \cos z = (0.62374)(0.35672) + (0.78163)(0.93421)(0.96952) \]
\[ = +0.93045 \]

\[ a = 68^\circ 5 \]

\[ x = \tan A = 0.24502/((0.96952)(0.62374) - (0.38184)(0.78163)) \]
\[ = +0.80001 \quad \arctan x = +38^\circ 7 \]

Since \( \text{LHA} \) is greater than \( 0^\circ \) and less than \( 180^\circ \), and since \( x \) is positive,
\[ A = 180^\circ + 38^\circ 7 = 218^\circ 7 \]

Sunrise, Sunset and Twilight

For locations between latitudes \( 65^\circ \) North and \( 65^\circ \) South, the following algorithm provides times of sunrise, sunset and twilight to an accuracy of \( \pm 2^m \), for any date in the latter half of the twentieth century. Because the phenomena depend on local meteorological conditions, attempts to attain higher accuracy are seldom justified. Although the algorithm can be used at higher latitudes, its accuracy deteriorates near dates on which the Sun remains above or below the horizon for more than twenty-four hours.

Notation:

\[ \phi = \text{latitude of observer (north is positive; south is negative)} \]
\[ \lambda = \text{longitude of observer (east is positive; west is negative)} \]
\[ M = \text{Sun’s mean anomaly} \]
\[ L = \text{Sun’s true longitude} \]
RA = Sun's right ascension
δ = Sun's declination
H = Sun's local hour angle
z = Sun's zenith distance at rise, set or twilight*
t = approximate time of phenomenon in days since 0 Jan., 0 UT
T = local mean time of phenomenon
UT = universal time of phenomenon

*The proper value of z should be chosen from the following:

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>z</th>
<th>cos z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunrise and Sunset</td>
<td>90°50'</td>
<td>-0.01454</td>
</tr>
<tr>
<td>Civil Twilight</td>
<td>96°</td>
<td>-0.10453</td>
</tr>
<tr>
<td>Nautical Twilight</td>
<td>102°</td>
<td>-0.20791</td>
</tr>
<tr>
<td>Astronomical Twilight</td>
<td>108°</td>
<td>-0.30902</td>
</tr>
</tbody>
</table>

Formulas:

1. \( M = 0°985600t - 3°289 \)
2. \( L = M + 1°916 \sin M + 0°020 \sin 2M + 282°634 \)
3. \( \tan RA = 0.91746 \tan L \)
4. \( \sin \delta = 0.39782 \sin L \)
5. \( x = \cos H = (\cos z - \sin \delta \sin \phi) / (\cos \delta \cos \phi) \)
6. \( T = H + RA - 0°065710t - 0°622 \)
7. \( UT = T + \lambda \)

Procedure:

1. With an initial value of \( t \), compute \( M \) from Eq. (1) and then \( L \) from Eq. (2). If a morning phenomenon (sunrise or the beginning of morning twilight) is being computed, construct an initial value of \( t \) from the formula
   \( t = N + (6h - \lambda)/24 \)
   where \( N \) is the day of the year (see the calendar on pages A2–A3 or the formulas on page B1) and \( \lambda \) is the observer's longitude expressed in hours. If an evening phenomenon is being computed, use
   \( t = N + (18h - \lambda)/24 \)
2. Solve Eq. (3) for \( RA \), noting that \( RA \) is in the same quadrant as \( L \). Transform \( RA \) to hours for later use in Eq. (6).
3. Solve Eq. (4) for \( \sin \delta \) which appears in Eq. (5); \( \cos \delta \), which also is required in Eq. (5), should be determined from \( \sin \delta \). While \( \sin \delta \) may be positive or negative, \( \cos \delta \) is always positive.
4. Solve Eq. (5) for \( H \). Since computers and calculators normally give the arccosine in the range 0°–180°, the correct quadrant for \( H \) can be selected according to the following rules:
   (a) rising phenomena, \( H = 360° - \arccos x \);
   (b) setting phenomena, \( H = \arccos x \).
In other words, for rising phenomena $H$ must be either in quadrant 3 or 4 (depending on the sign of $\cos H$), whereas $H$ must be either in quadrant 1 or 2 for setting phenomena. Convert $H$ from degrees to hours for use in Eq. (6).

5. Compute $T$ from Eq. (6), recalling that $H$ and $RA$ must be expressed hours. If $T$ is negative or greater than 24, it should be converted to the range $0-24$ by adding or subtracting multiples of 24.

6. Compute $UT$ from Eq. (7), where $\lambda$ must be expressed in hours. $UT$ is an approximation to the time of sunrise, sunset or twilight, referred to the Greenwich meridian. If $UT$ is greater than 24, the phenomenon occurs on the following day, Greenwich time. If $UT$ is negative, the phenomenon occurs on the previous day, Greenwich time.

To ensure that precision is not lost during the computations, $t$ should be carried to four decimal places. Angles should be expressed to three decimals of a degree and, upon conversion, to three decimals of an hour. Five significant digits should be carried for the trigonometric functions.

Under certain conditions Eq. (5) will yield a value of $|\cos H| > 1$, indicating the absence of the phenomenon on that day. At far northern latitudes, for example, there is continuous illumination during certain summer days and continuous darkness during winter days.

Example: Compute the time of sunrise on 25 June at Wayne, New Jersey.

Latitude: 40°9 North Longitude: 74°3 West

$\varphi = +40^\circ 9 \quad \sin \varphi = +0.65474 \quad \cos \varphi = +0.75585$

$\lambda = -74^\circ 3/15 = -4^h 55'$

For sunrise: $z = 90^\circ 50' \quad \cos z = -0.01454$

$t = 176^d + (6^h + 4^m 95')/24 = 176^d 456$

$M = 0^\circ 985600 (176^d 456) - 3^\circ 289 = 170^\circ 626$

$L = 170^\circ 626 + 1^\circ 916 (0.16288) + 0^\circ 020 (-0.32141) + 282^d 634$

$= 453^d 566 = 93^d 566$

$\tan RA = 0.91746 (-16.047) = -14.723$

$RA = 93^\circ 886/15 = 6^h 259$ Since $L$ is in quadrant 2, so is $RA$.

$\sin \delta = 0.39782 (0.99806) = 0.39705$

$\cos \delta = 0.91780$

$x = \cos H = [-0.01454 - (0.39705) (0.65474)] / [(0.91780) (0.75585)]$

$= -0.39570 \quad \arccos x = 113^\circ 310$

Since sunrise is being computed, $H = 360^\circ - 113^\circ 310 = 246^\circ 690$

$H = 246^\circ 690/15 = 16^h 446$

$T = 16^h 446 + 6^h 259 - 0^\circ 065710 (176^d 456) - 6^h 622 = 4^h 488$

$UT = 4^h 488 + 4^h 95 = 9^h 44$

Sunrise occurs at 9h 26m UT = 5h 26m EDT
Solar Coordinates

The true geocentric longitude of the Sun \((L)\) can be computed to an accuracy of ±1 minute of arc from the following formulas:

\[
\begin{align*}
M &= 357.529 + 35999.050 T \\
\mathcal{L} &= 280^\circ + 36000^\circ T \\
L &= \mathcal{L} + (1591.15 - 0.0067 T) \sin M + 0.020 \sin 2M
\end{align*}
\]

where \(T = (JD - 2451545.0)/36525\) and \(JD\) is the Julian Date (see page B21.

If we consider the Sun's latitude to be identically zero, the right ascension \((RA)\) and declination \((\delta)\) of the Sun can also be computed to ±1 minute of arc from

\[
\begin{align*}
\tan RA &= \cos \varepsilon \tan L \\
\sin \delta &= \sin e \sin L
\end{align*}
\]

where \(\varepsilon\), the obliquity of the ecliptic, can be computed from

\[
\varepsilon = 23^\circ.439 - 0.013 T.
\]

Because the obliquity varies slowly, a single value can be used for an extended period of time. During the last quarter of the twentieth century, \(\varepsilon = 23^\circ.441\) is sufficiently accurate. Similarly the coefficient of \(\sin M\) in the equation for \(L\) changes slowly; for the last half of the twentieth century a value of 1°.916 can be safely used.

Although there is no rigorous limit on the time span for which these formulas are valid, their accuracy gradually deteriorates for values of \(T\) greater than a couple of centuries.

Equation of Time and Time of Solar Transit

The equation of time \((\text{Eq}T)\) is the hour angle of the true Sun minus the hour angle of the mean sun. Thus it is the difference: apparent solar (sundial) time minus mean solar (clock) time.

For the current year \(\text{Eq}T\) can be computed to an accuracy of ±0.8 minute from the following formula:

\[
\begin{align*}
\text{Eq}T &= -764 \sin (0^\circ.9856 t) + 0^\circ.57 \cos (0^\circ.9856 t) \\
&\quad -97\sin (1^\circ.9712 t) - 2\cos (1^\circ.9712 t)
\end{align*}
\]

where \(t\) is the number of days since 0 January, 0\(^{h}\) UT.

If higher accuracy is required the following formulas will give \(\text{Eq}T\) to an accuracy of ±2 seconds during the current year:

\[
\begin{align*}
\theta &= 8^\circ.855 + 0^\circ.98561 t + 1^\circ.916 \sin (0^\circ.9856 t - 3^\circ.819) \\
&\quad + 0.020 \sin (1^\circ.9712 t - 7^\circ.638)
\end{align*}
\]

\[
\begin{align*}
\text{Eq}T &= 35^\circ.421 + 3^\circ.94244 t - 4^\circ.0 \arctan [(\tan \theta)/0.91747]
\end{align*}
\]

where \(t\) is the number of days, and fractions thereof, since 0 January, 0\(^{h}\) UT. In Eq. (3) the arctangent should yield a result in degrees that is in the same quadrant as \(\theta\). Near the end of the year \(\theta\) becomes greater than 360°. When this occurs the arctangent in Eq. (3) should also be greater than 360°.
Eqs. (2) and (3) can be used to compute the time at which the Sun transits the local meridian. First use Eqs. (2) and (3) to compute \( EqT \) for \( t = N + (12^h - \lambda)/24 \), where \( N \) is the day of the year (see the calendar on pages A2-A3 or the formulas on pages B1-B2) and \( \lambda \) is the longitude (east positive, west negative) expressed in hours. Then the local mean time (LMT) of transit is given to an accuracy of ±2 seconds by

\[
LMT = 12^h - EqT.
\]

The universal time of local transit is then obtained from

\[
UT = LMT - \lambda.
\]

Example: Compute the time of solar transit at longitude 9°07'2 East on 13 December 1984.

\[
\lambda = +9°12'/15 = +0^h 60^m 80^s = +0^h 36^m 48^s
\]

For solar transit:

\[
t = 348^d + (12^h - 0^h 60^m 80^s)/24 = 348^d 47^m 47^s
\]

\[
\theta = 8°85^m + 0°985^m (348^d 47^m 47^s) + 1°916^m (-0.3480)
\]

\[
+ 0°020^m (-0.6524) = 351°635^m
\]

\[
EqT = 35°421^m + 3°94244^m (348^d 47^m 47^s) - 4°0 arctan \left( -0.14704/0.91747\right)
\]

\[
= 35°421^m + 3°94244^m - 4°0 (350°895) = +5°68
\]

\[
LMT = 12^h 00^m - 5°68 = 11^h 54^m 32^s
\]

\[
UT = 11^h 54^m 32^s - 0^h 36^m 48^s = 11^h 17^m 50^s UT
\]

Moonrise and Moonset

Times of moonrise and moonset can be computed for specified locations using the following algorithm. Between latitudes 60° North and 60° South, the phenomena can be computed to an accuracy of ±5'. Although the algorithm can be used at higher latitudes, its accuracy deteriorates near dates on which the Moon remains above or below the horizon for more than twenty-four hours.

Notation:

\( \phi = \) latitude of observer (north is positive; south is negative)

\( \lambda = \) longitude of observer (east is positive; west is negative)

\( t_i = i\text{-th approximation to universal time of phenomenon, expressed in fractions of a day from } 0^h UT \)

\( GHA_i = \) Moon’s GHA at time \( t_i \)

\( \delta_i = \) Moon’s declination at time \( t_i \) (north is positive; south is negative)

\( \tau_i = i\text{-th correction to } t_0, \text{ thus } t_i = t_0 + \tau_i \)

\( \Delta H_i = i\text{-th approximation to Moon’s LHA at time of rise or set} \)

\( \Delta H_i = i\text{-th approximation to Moon’s daily rate of change in GHA} \)

Formulas:

1. \( \Delta H_i = (GHA_i - GHA_0) / \tau_i \) for \( i = 0 \), let \( \Delta H_0 = 347°.81 \)
2. \( x_{i+1} = \cos H_{i+1} = \left( .00233 - \sin \phi \sin \delta_i \right) / \left( \cos \phi \cos \delta_i \right) \)
3. \( \tau_{i+1} = \left( H_{i+1} - H_0 \right) / \Delta H_i \)
4. \( t_{i+1} = t_0 + \tau_{i+1} \)
Procedure:

1. Let \( t_0 = \frac{(12^h - \lambda)}{24} \), where \( \lambda \) is the observer's longitude expressed in hours. Set \( i = 0 \) and begin the following iterative process.

2. For time \( t_i \) compute the Moon's GHA and declination to navigational precision (±0'1). Label these quantities \( \text{GHA}_i \) and \( \delta_i \), respectively, where \( i \) specifies the iteration number. For \( i = 0 \), compute \( H_0 = \text{GHA}_0 + \lambda \).

3. If \( i = 0 \), let \( \Delta H_0 = 347^\circ 81 \). Otherwise compute \( \Delta H_i \) from Eq. (1). If \( \text{GHA}_i < GHA_0 \), add 360° to \( \text{GHA}_i \) before computing \( \Delta H_i \).

4. Solve Eq. (2) for \( \tau_{i+1} \). Since computers and calculators normally give the arc-cosine in the range \( 0^\circ - 180^\circ \), the correct quadrant for \( \tau_{i+1} \) can be selected according to the following rules:
   (a) moonrise computations, \( \tau_{i+1} = 360^\circ - \arccos x_{i+1} \)
   (b) moonset computations, \( \tau_{i+1} = \arccos x_{i+1} \).

5. Compute \( \tau_{i+1} \) from Eq. (3). If \( |\tau_{i+1}| < 0^\circ 5 \), proceed to Step 6. If \( |\tau_{i+1}| > 0^\circ 5 \), the phenomenon being computed occurs on the day prior to the day desired (if \( \tau_{i+1} \) is negative) or on the day following the day desired (if \( \tau_{i+1} \) is positive). Normally the phenomenon on the desired day can be obtained by adding to \( \tau_{i+1} \) (if \( \tau_{i+1} \) is negative), or subtracting from \( \tau_{i+1} \) (if \( \tau_{i+1} \) is positive). 360°, \( \Delta H_i \). If successful this technique will produce a new value of \( \tau_{i+1} \) in the required range. However, two conditions may prevent the reduction to \( |\tau_{i+1}| < 0^\circ 5 \):
   (a) for low values of \( i \), \( \tau_{i+1} \) may be a fairly crude approximation to the ultimate value, \( \tau_{i+1} \).
   (b) each month there is one day (near last quarter) on which there is no moonrise, and another day (near first quarter) on which there is no moonset.

6. Compute \( \tau_{i+1} \) from Eq. (4). If \( |\tau_{i+1} - \tau_i| < 0^\circ 01 \), \( \tau_{i+1} \) is accurate to ±5'. Otherwise it is necessary to iterate the solution by setting \( i = i + 1 \) and executing Steps 2 through 6 again.

Example: Compute moonrise on 15 October 1984 at San Francisco, California.

\[ \phi = +37^\circ 45 \quad \sin \phi = +0.60807 \quad \cos \phi = +0.79388 \]
\[ \lambda = -122^\circ 27 = -8^h 151 \]
\[ t_0 = \frac{12^h + 8^m15^s}{24} = 0^d83963 \]

\[ i = 0: \text{ Evaluating the power series on page C17 for } t_0 \text{ on 15 October,} \]
\[ GHA_0 = 239^\circ075 \quad \delta_0 = +26^\circ045 \quad \cdot \cdot \cdot \]
\[ H_0 = 239^\circ075 - 122^\circ270 = 116^\circ805 \]
\[ \Delta H_0 = 347^\circ81 \]
\[ x_1 = \cos H_1 = \frac{0.00233 - (0.60807)(0.43908)}{((0.79388)(0.89845))} = -0.37106 \]
\[ \arccos x_1 = 111^\circ781 \]

Since moonrise is sought, \( H_1 \) is in quadrant 3 or 4:
\[ H_1 = 360^\circ - 111^\circ781 = 248^\circ219 \]
\[ \tau_1 = \frac{248^\circ219 - 116^\circ805}{347^\circ81} = +0^\circ37783 \]
\[ |\tau_1| < 0^\circ5, \text{ as required.} \]
\[ t_1 = 0^d83963 + 0^\circ37783 = 1^d21746 = 5^h13^m \text{ UT on 16 October} \]

\[ i = 1: \text{ Evaluating the power series on page C17 for } t_1 \text{ on 16 October,} \]
\[ GHA_1 = 10^\circ087 \quad \delta_1 = +26^\circ402 \]
\[ \Delta H_1 = \frac{370^\circ087 - 239^\circ075}{0^\circ437783} = 346^\circ749 \]
\[ x_2 = \cos H_2 = \frac{10.00233 - (0.60807)(0.44467)}{((0.79388)(0.89570))} = -0.37698 \]
\[ \arccos x_2 = 112^\circ147 \]

Since moonrise is sought, \( H_2 \) is in quadrant 3 or 4:
\[ H_2 = 360^\circ - 112^\circ147 = 247^\circ853 \]
\[ \tau_2 = \frac{247^\circ853 - 116^\circ805}{346^\circ749} = +0^\circ37793 \]
\[ t_2 = 0^d83963 + 0^\circ37793 = 1^d21756 = 5^h13^m \text{ UT on 16 October} \]
\[ = 10:13 \text{ p.m., Pacific Daylight Time, on 15 October} \]
\[ |t_2 - t_1| = 0^\circ0001 < 0^\circ01 \]

The extremely rapid convergence illustrated in this example occurs frequently but not invariably. Although the first approximation \( t_1 \) will often give adequate precision for most purposes, it is recommended that the solution be iterated and that the convergence criterion \( |t_{i+1} - t_i| < 0^\circ01 \) be tested.

**Polaris (Pole Star)**

The following formulas are relevant to observations of Polaris:

1. \[ \phi = a - p \cosh + 0.5p \sin p \sin^2 h \tan \phi \]
2. \[ A \cos \phi = -p \sin h - p \sin h \cosh \tan \phi \]

where \( p \) is the polar distance of Polaris: \( p = 90^\circ \) - declination of Polaris

\( h \) is the LHA of Polaris: \( h = \text{GHA Aries} + \text{SHA Polaris} + \text{east (-west) longitude of observer} \)

\( \phi \) is the observer's latitude;

\( A \) is the azimuth of Polaris;

\( a \) is the corrected altitude of Polaris.
Eq. (1) permits the observer's latitude to be determined from an observation of the altitude of Polaris (corrected for refraction, dip, etc.). Assumed values of the observer's latitude and longitude can be used for the right side of Eq. (1). Eq. (2) yields the azimuth of Polaris if the observer's position is known. These expressions are accurate only for Polaris, since they depend on $p$ being a small quantity. The SHA and declination of Polaris to be used in these formulas should be referred to the true equator and equinox of date; i.e., the apparent place of Polaris should be computed (see Section E: 'Stellar Tables'; Polaris is star number 17).

Equation of Position Line

The following formula can be used to obtain a line of position (LOP) directly from an observation of the altitude of a celestial body:

$$X = \text{GHA} \pm \arccos \left[ \frac{\sin a - \sin \phi \sin d}{\cos \phi \cos d} \right]$$

where:

- $X$ is the computed longitude;
- GHA is the GHA of the body for the time of observation;
- $a$ is the corrected altitude of the body;
- $d$ is the declination of the body for the time of observation;
- $\phi$ is an estimate of the observer's latitude.

North latitudes and west longitudes are positive; south latitudes and east longitudes are negative. Longitudes with absolute values greater than 180° may be encountered.

In the above formula, $+$ is used for bodies east of the meridian (rising) and $-$ for bodies west of the meridian (setting).

The formula gives the longitude $\lambda$ at which the position line crosses the parallel of latitude $\phi$. Repeated application of the formula using different values of latitude yields a locus of points all lying in the LOP. Note that no assumed position is necessary, although an estimate of the observer's latitude is helpful in reducing the number of times the formula is applied.

The formula becomes indeterminate at the transit time of a body and for latitudes that the position line does not cross at any point.

Motion of Body and Motion of Observer

During the time interval $\Delta t$ (e.g., the interval between a sextant observation and the time of a fix), the rotation of the Earth causes a change in the altitude of a celestial body. To permit the use of a common assumed position and LHA Aries for observations made at different times, the following correction can be applied to the observed altitude:

$$\text{MOB} = 15.04 \Delta t \cos \phi \sin A$$

where MOB is the altitude correction in minutes of arc, $\Delta t$ is the time difference in minutes, $\phi$ is the latitude of the observer, and $A$ is the azimuth of the observed body. If the time of the fix is later than the time of observation, MOB should be added to the observed altitude. It should be noted that the formula for MOB is an approximation that becomes unreliable for values of $\Delta t$ greater than 5 minutes.
The following formula gives the change of altitude of a celestial body due to the motion of the observer in the time interval $\Delta t$ (e.g., the interval between a sextant observation and the time of a fix). Though this formula is only an approximation to the physical phenomenon, it is the exact mathematical equivalent of advancing or retiring a line of position.

$$MOO = \frac{v \Delta t}{60} \cos (A - C)$$

where $MOO$ is the altitude correction in minutes of arc, $\Delta t$ is the time difference in minutes, $A$ is the azimuth of the observed body, $C$ is the track/course angle, and $v$ is the ground speed in knots. If the time of the fix is later than the time of observation, $MOO$ should be added to the observed altitude.

Sextant Altitude Corrections

Several corrections must be applied to a sextant altitude ($hs$) in order to obtain a corrected altitude ($Ho$). $Ho$ can then be either (a) compared with the computed altitude ($hc$) to obtain the altitude difference ($\Delta h$); or (b) used in the 'Equation of Position Line' (see p. B12) to obtain directly the location of the LOP for the sight.

The corrections, in the order in which they should be applied, are:

1. Instrument and/or index correction, IC;
2. Dip of Horizon, $D$ (marine sextant); or Coriolis correction, $Az$ (bubble sextant);
3. Atmospheric refraction, $R$;
4. Semidiameter, $SD$ (marine sextant, Sun and Moon observations);
5. Parallax in altitude, $PA$ (Moon, Venus and Mars observation).

In mathematical notation:

$$Ho = hs + IC + (D \text{ or } Az) - R + SD + PA$$

If Venus is observed, an additional correction for the phase of the planet may be necessary. This correction can be made either to the sextant altitude or to the GHA or LHA of Venus.

Descriptions and formulas for $D$, $Az$, $R$, $SD$, $PA$ and the phase correction for Venus are given on the following pages.

Dip of Horizon

The dip of the apparent horizon from a horizontal plane is given by

$$D = -0.97\sqrt{h}$$

where $h$ is the height of eye level of the observer in feet and $D$ is the dip of the horizon in minutes of arc. For observations of a celestial body made with a marine sextant or similar instrument, $D$ should be added to the observed altitude to obtain the corrected altitude. This formula is an approximation; the apparent dip varies with atmospheric conditions.
Coriolis Correction

Any object moving across or above the surface of the rotating Earth is subject to an apparent force tending to push the object to the right in the northern hemisphere and to the left in the southern hemisphere. This Coriolis acceleration manifests itself as a deflection of the apparent vertical by an amount \( Z \):

\[
Z = 2.62 V \sin \phi + 0.146 V^2 \sin C \tan \phi - 5.25 V C'
\]

where: \( Z \) is the deflection in minutes of arc;
\( V \) is the speed in hundreds of knots;
\( \phi \) is the latitude;
\( C \) is the true track/course angle;
\( C' \) is the rate of change of true track/course angle in degrees per minute of time.

The 'Coriolis (Z) Correction' tabulated in the Air Almanac consists of only the first term in Eq. (1). The second term is known as 'Rhumb Line Correction', and the third term is the 'Wander Correction'. Usually only the first term is significant.

Observations of the altitudes of celestial bodies made with bubble sextants or similar artificial horizon instruments must be corrected for the Coriolis effect. The correction \( \Delta z \), which can be added to the observed (e.g., bubble sextant) altitude, is given approximately by

\[
\Delta z = Z \sin(A - C)
\]

where: \( \Delta z \) is the altitude correction in minutes of arc;
\( Z \) is the deflection of the vertical determined from Eq. (1);
\( A \) is the azimuth of observed body;
\( C \) is the true track/course angle.

In the northern hemisphere the correction \( \Delta z \) is positive for stars on the right and negative for stars on the left of the aircraft. In the southern hemisphere the correction is negative for stars on the right and positive for stars on the left.

Atmospheric Refraction

The Earth's atmosphere tends to refract light in such a way that celestial bodies appear slightly higher in the sky than they would if there were no atmosphere. The formulas below can be used to determine \( R \), the angle of refraction. \( R \) should be subtracted from an observed (e.g., sextant) altitude to obtain the corrected altitude.

\[
R = \frac{P}{273 + T} [3.430289 (z - \arcsin [0.9986047 \sin (0.9967614z)]) - 0.01115929z]
\]

\[
R = \exp(-h/27000) \tan z = 1/[\exp(h/27000) \tan a]
\]

where: \( R \) is the refraction correction in minutes of arc;
\( a \) is the observed altitude;
\( z \) is the observed zenith distance in degrees: \( z = 90^\circ - a; \)
$T$ is the temperature in degrees Celsius;
$P$ is the atmospheric pressure in millibars;
$h$ is the height of the observer above sea level in feet.

The formulas given above are approximations and are not equivalent to a complete theory of refraction. Eq. (1), which is better suited to surface observations, is accurate to ±0.1 for altitudes greater than 15°; between altitudes 3° and 15°, errors can reach ±1'; for altitudes less than 3°, errors between ±1' and ±3' will be encountered. Eq. (2), which is better suited to observations from aircraft, should only be used for altitudes greater than 10°. Above 10° this formula is accurate to ±0.2.

For surface observations under standard atmospheric conditions, the following Chebyshev series represents refraction for altitudes from 0° to 90° with errors not exceeding ±0.1.

$$R = \frac{a_0}{2} + \sum_{i=1}^{9} a_i T_i(x)$$

where $x$ is related to the observed altitude by $x = 0.442837 \log_e (a + 1.5) - 1$.

The coefficients $a_i$ in the series are

$$a_0 = +28.891741 \quad a_5 = +0.340097$$
$$a_1 = -20.516167 \quad a_6 = -0.024576$$
$$a_2 = +7.291562 \quad a_7 = -0.050041$$
$$a_3 = -0.813492 \quad a_8 = +0.023252$$
$$a_4 = -0.690042 \quad a_9 = -0.009406$$

The sum of these coefficients is +14.442928.

For a given value of the altitude $a$, compute $x$. Then the series can be evaluated as follows:

let $b_{10} = b_{11} = 0$,
compute $b_i = 2x b_{i+1} - b_{i+2} + a_i$, for $i = 9, 8, \ldots, 0$,
then $R = (b_0 - b_2)/2$.

Example: A star is observed from the Earth's surface under standard atmospheric conditions to be at altitude 10°. Use the Chebyshev series to compute the refraction correction.

$x = 0.442837 \log_e (10.0 + 1.5) - 1 = +0.081562$

$R = (18.296200 - 7.660919)/2 = 5.3'$
Semidiameter of the Sun and Planets

The semidiameters of the Sun and planets can be computed from
\[ SD = S/d = S\pi/8.794 \]
where:
- \( SD \) is the semidiameter in seconds of arc;
- \( S \) is the semidiameter at unit distance (1 AU) in seconds of arc;
- \( d \) is the geocentric distance in AU;
- \( \pi \) is the horizontal parallax in seconds of arc.

The following values of \( S \) should be used:

<table>
<thead>
<tr>
<th>Planet</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>959.63</td>
</tr>
<tr>
<td>Jupiter</td>
<td>98.47</td>
</tr>
<tr>
<td>Mercury</td>
<td>3.34</td>
</tr>
<tr>
<td>Venus</td>
<td>8.41</td>
</tr>
<tr>
<td>Mars</td>
<td>4.68</td>
</tr>
<tr>
<td>Uranus</td>
<td>34.28</td>
</tr>
<tr>
<td>Saturn</td>
<td>83.33</td>
</tr>
<tr>
<td>Neptune</td>
<td>36.56</td>
</tr>
</tbody>
</table>

These values apply to the equatorial dimensions of the bodies and do not include any adjustments for irradiation.

Semidiameter of the Moon

The geocentric semidiameter of the Moon can be computed from
\[ SD = 56204.92/d = 0.272476\pi \]
where \( SD \) is the geocentric semidiameter in seconds of arc, \( d \) is the geocentric distance of the Moon in units of the Earth's equatorial radius, and \( \pi \) is the horizontal parallax of the Moon in seconds of arc.

Since observations are made from the Earth's surface rather than from its center, the observed, topocentric semidiameter is slightly greater than the geocentric semidiameter. For navigation and certain other purposes the augmented semidiameter of the Moon should be used:
\[ SD_{aug} = SD[1 + (\sin a)/d] \]
where \( SD_{aug} \) is the augmented semidiameter in seconds of arc, \( a \) is the altitude of the Moon (for navigational purposes \( a = H_0 \), but \( H_0 \) or \( H_c \) can be used instead with negligible error), \( d \) is the geocentric distance of the Moon in units of the Earth's equatorial radius, and \( SD \) is the geocentric semidiameter computed from Eq.(1). For navigational purposes a constant value of \( d = 60.27 \) can be used to sufficient accuracy. The increase in the Moon's semidiameter due to augmentation is zero when the Moon is on the horizon and is about 0.3 when the Moon is at the zenith.

Parallax in Altitude

The finite size of the Earth causes a parallactic shift in the apparent positions of nearby celestial objects. The resulting parallax in altitude can be computed from
\[ \sin PA = \sin \pi \cos a \]
where PA is the parallax in altitude, $\pi$ is the horizontal parallax, and $\alpha$ is the observed altitude. When the horizontal parallax of a body is not available, it can be computed from the relation $\pi = 8.794/d$, where $d$ is the geocentric distance of the body in astronomical units. Except for the Moon, parallax in altitude does not exceed 1'. Since parallax tends to decrease the apparent altitude of a body, the quantity PA should be added to an observed (e.g., sextant) altitude in order to obtain the corrected altitude. To a reasonable approximation, PA can also be computed from

$$PA = \pi \cos \alpha$$

Correction for the Phase of Venus

When the altitude of Venus is observed with a small instrument, a correction to the observed altitude is required to account for the fact that the center of light, rather than the center of the disk, is observed. This correction has the form $-k \cos \theta$, where $k$ is a correction factor (given below) and $\theta$ is the angle on the celestial sphere, at the position of Venus, between the observer's vertical and the direction of the Sun. The correction, which should be added to the observed (e.g., sextant) altitude, is positive when the Sun is lower than Venus, zero when they have the same altitude, and negative when the Sun is higher.

In sight reduction this effect can be approximately taken into account by correcting the GHA (or LHA) of Venus rather than correcting the observed altitude. Simply add $k$ to the GHA (or LHA) of Venus when Venus is east of the Sun (i.e., when Venus is an evening planet), and subtract $k$ from the GHA (or LHA) when Venus is west of the Sun (morning planet). The correction should not be applied in this way near the time of superior or inferior conjunction.

In 1984 Venus is in the morning sky from the beginning of the year until mid-May, when it becomes too close to the Sun for observation. Late in May it reappears in the evening sky, where it is visible for the rest of the year.

The values of $k$ for 1984 are

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
</tr>
</tbody>
</table>
1011984.TEMP
MOONRISE/MOONSET/MOON PHASE (NAVAL OBSERVATORY, VALUES GOOD THRU CT 1984)
201VALID FOR LOCATIONS 60degN THRU 60degS LATITUDE
40 CCRear & MSCALE 0,0 @ CSIZE 9 @ MOVE 4,55 @ LABEL "MOONRISE/MOON/PHASE"
50 MOVE 45,49 @ LABEL "PROGRAM"
60 CSIZE S @ MOVE 17,43 @ LABEL "(VALID FOR LATITUDES 60degN TO 60degS)"
70 MOVE 33,9 @ LABEL "PRESS 'CONT' WHEN READY..." @ PAUSE
80 GCLEAR
90 PRINTER IS 580,132 @ DEG & DIM D(140)
100 PRINT "A11V" & &G8" @ FOR I=1 TO 25 @ PRINT "MOON","NEXT" I @ PRINT "MOON","611V"
110 CLEAR & PRINT "&K38,\ tabs(32),"MOONRISE/MOONSET/MOON PHASE--1984"
120 PRINT TAB (27),"
130 PRINT "&K5" & PRINT TAB (32),"Nautical Almanac Office, US Naval Observatory" & PRINT & PRINT
140 PRINT TAB (32),"(VALID FOR LATITUDES 60degN TO 60degS)" @ PRINT " A11V"
150 C=1 @ Q3=0 @ Q2=0
160 CLEAR & D="ENTER THE MONTH, DAY (example: 10,20)" @ DISP HGL% (D4,1) @ INPUT MONTH,DAY
170 YEAR=1984 VALUE TO BE UPDATED ANNUALLY
180 IF MONTH=12 OR DAY=31 THEN BEEP & CLEAR & GOTO 160
190 PRINT " &K35"
200 PRINT USING 212, MONTH,DAY,YEAR
210 IMAGE "DATE: ","D D,","Z/Z/","DDDD"
220 IF YEAR=4*print (YEAR/4) THEN RESTORE 240 ELSE RESTORE 250 @ DETERMINES LEAP YEAR
230 L=LINE 240 = LEAP YEAR (366 DAYS) LINE 258 = 365 DAY YEAR
240 DATA 31,60,91,121,152,182,213,244,274,305,335
250 DATA 31,59,90,120,151,181,212,243,273,304,334
260 T=0 & FOR I=1 TO MONTH-1 & READ Y & NEXT I & GOTO 350
270 PRINT "JULIAN DATE: ","JDATE"
280 CLEAR & D=" LATITUDE (degrees,minutes) example: 37deg 14min = 37.24 " @ DISP HGL% (D4,1) & INPUT LAT1,MINUTES1
290 IF LAT1=60 OR MINUTES1=60 THEN BEEP & CLEAR & GOTO 260
300 LAT=LAT1+MINUTES1/60
310 DISP & D=" DIRECTION FROM EQUATOR (N/S)" @ DISP HGL% (D4,1) & INPUT H & IF H=11="S" THEN LAT="-LAT
320 PRINT USING 330, LAT1,MINUTES1,H
330 IMAGE "LATITUDE: ","DDD,","Degrees",X,D,"Minutes",X,A
340 IMAGE "LONGITUDE: ","DDD,","Degrees",X,D,"Minutes",X,A
350 CLEAR & D=" LONGITUDE (degrees,minutes) example: 34deg 14min = 34.24 " @ DISP HGL% (D4,1) & INPUT LONG1,MINUTES2
360 IF LONG1=180 OR MINUTES2=60 THEN BEEP & CLEAR & GOTO 350
370 LONG=LONG1+MINUTES2/60
380 DISP & D=" DIRECTION FROM GREENWICH (E/W)" @ DISP HGL% (D4,1) & INPUT J & IF J=11="W" THEN LONG=LONG-W
390 PRINT USING 340, LONG1,MINUTES2,J
400 CLEAR & D=" MOON RISE AND SET TIMES BEING COMPUTED " & AWRITE 9,15,HGL% (D4,1)
410 T=(12-LONG/15)/24 @ T1=T0
420 DS=347.01 @ A=A+1 VALUE TO BE ANNUALLY UPDATED
430 COSU=630
440 23=Z @ 24=Z1
450 H0=Z+LONG
460 GOTO 490
470 GOSUB 630
480 Z3=Z @ 24=Z1
490 H0=Z+LONG
500 GOTO 460
510 IF D5=0 THEN DS=DS+360/ABS(XB)
520 X2=(0.0233-SIN(LAT)*SIN(Z1))/COS(LAT)*COS(Z1)
530 IF C=2 THEN GOTO 530
540 H1=360-AC3(X2)
550 GOTO 540
560 H1=AC3 (X2)
570 X0=(H1-H0)/DS
580 IF X0<5 THEN XB=X0-360/DS
590 IF X0>5 THEN XB=X0+360/DS
600 T2=T0
601 IF X0<(T2-T1)/.01 THEN GOTO 1010
602 GOTO 470
660  U=M8+1
670  P=MONT*4-3-N
680  I CHOOSE PROPER DATA TO USE IN SERIES EXPANSION CALCULATION.
690  IF P>12 THEN 710
700  RESTORE 1800 GOTO 760
710  IF P<24 THEN 730
720  RESTORE 2670 GOTO 760
730  IF P>36 THEN 750
740  RESTORE 2340 GOTO 760
750  RESTORE 2610
760  FOR LOOP=1 TO P MOD 12
770   READ 88,88,CB,DD,EB,GB,HB,IB,JB,KB,KB,MB,MB,MB,KB,KB
780  NEXT LOOP
790  RESTORE 1470
800  I SERIES EXPANSIONS (7TH ORDER) FOR GHA AND DEC
810  X=(DAY+T1-V)/A-1
820  D7=X%HB
830  D6=X%(GB+D7)
840  D5=X%(FD+D6)
850  D4=X%(EB+D5)
860  D3=X%(DD+D4)
870  D2=X%(CB+D3)
880  B1=X%(JB+D2)
890  Z1=(A+811) MOD 360 M OON'S GHA AT TIME TO
900  B7=X%PB
910  B6=X%(GB+D7)
920  B5=X%(HB+D6)
930  B4=X%(MB+D5)
940  B3=X%(LB+D4)
950  B2=X%(KB+D3)
960  B1=X%(JB+D2)
970  Z1=(X+81) MOD 360 M OON'S DECLINATION AT TIME TO
980  RETURN
990 1600 IF C=2 THEN GOTO 1060
990 1610 IF C=1 OR C=2 THEN COSUB 1060
1000 1620 GOTO 1030
1010 1630 C=2 GOTO 410
1020 1640 PAUSE
1030 1650 END
1060 1660 DAYS=T2JDATE
1070 1670 HOURS=(DAYS-IP(DAYS))#24
1080 1680 MINUTES=INT((HOURS-IP(HOURS))#60)
1090 1690 IF C=1 THEN A="MOONRISE"
1100 IF C=2 THEN A="MOONSET" GOTO 1150
1110 1120 PRINT " GHA" SET I/O 7,16,1 @ PRINT " AIGB" LINES 4951 THRU 4180 PRINTS RESULTS
1130 1140 CLEAR @ D=" MOONRISE AND SET TIMES BEING PRINTED" @ AURITE 9.15,HGL? (D$1)
1150 1160 PRINT " DATE",TAB(27),"ZULU TIME"
1170 1180 PRINT TAB(12),"T A I R (12) JULIAN DATE",TAB(27)," ZULU TIME"
1190 1190 IF Q>0 THEN 1160 ELSE 1180
1200 1210 PRINT USING 1170 ;A,*IP(DAYS),IP(HOURS)*IP(MINUTES)
1220 1230 IMAGE 15A,DDD,10X,DD,="",Z2
1240 1250 IF Q<0 THEN PRINT A,TAB(12),"NONE-CHECK PREVIOUS AND FOLLOWING DAYS"
1260 1270 IF Q>0 THEN GOTO 1220
1280 1290 Q=0 @ Q=0
1300 1310 RETURN
1320 1330 PRINT " TIMES ARE ACCURATE TO + OR - 5 MINUTES" @ PRINT " AIGS"
1340 1350 CLEAR @ D=" DO YOU WANT MOONRISE AND MOONSET TIMES FOR OTHER DATES OR LOCATIONS (Y/N) " @ DISP HGL? (D$1) @ INPUT U66
1360 1370 IF U66(1,1)="Y" THEN CLEAR @ GOTO 140
1380 1390 PRINT " =HGL"
1400 1410 DIM X$(75),D$(5)
1420 1430 CLEAR @ D=" PHASE DATES BEING PRINTED" @ AURITE 9.26,HGL? (D$1)
1440 1450 IF JDATE=1 AND JDATE=31 THEN G="JANUARY" RESTORE 1470
1460 IF JDATE=2 AND JDATE=30 THEN G="FEBRUARY" RESTORE 1470
1470 IF JDATE=3 AND JDATE=31 THEN G="MARCH" RESTORE 1470
1480 IF JDATE=4 AND JDATE=30 THEN G="APRIL" RESTORE 1470
1490 IF JDATE=5 AND JDATE=31 THEN G="MAY" RESTORE 1470
1500 IF JDATE=6 AND JDATE=30 THEN G="JUNE" RESTORE 1470
1510 IF JDATE=7 AND JDATE=31 THEN G="JULY" RESTORE 1470
1520 IF JDATE=8 AND JDATE=30 THEN G="AUGUST" RESTORE 1470
1530 IF JDATE=9 AND JDATE=31 THEN G="SEPTEMBER" RESTORE 1470
1540 IF JDATE=10 AND JDATE=30 THEN G="OCTOBER" RESTORE 1470
1550 IF JDATE=11 AND JDATE=31 THEN G="NOVEMBER" RESTORE 1470
1560 IF JDATE=12 AND JDATE=30 THEN G="DECEMBER" RESTORE 1470
1570 JDATE2=0
1580 JDATE2=JDATE2+1
1590 IF JDATE2=366 THEN JDATE2=1
THE FOLLOWING 12 DATA STATEMENTS CONTAIN THE ABOVE INFORMATION FOR THE TWELVE MONTHS, ONE STATEMENT PER MONTH:

```
1470 DATA 2,3,11,18,25
1480 DATA 2,1,10,17,23
1490 DATA 2,2,10,17,24
1500 DATA 2,1,9,15,23
1510 DATA 2,1,8,15,22
1520 DATA 3,6,13,21,29
1530 DATA 3,5,13,21,28
1540 DATA 3,4,11,19,26
1550 DATA 3,2,10,18,25
1560 DATA 3,8,1,9,17,24,31
1570 DATA 4,8,16,22,30
1580 DATA 4,8,15,22,30
```

1590 READ W
1600 IF W = 10 THEN 1620
1610 READ Q(1),Q(2),Q(3),Q(4),Q(5)
1620 GO TO 1630
1630 PRINT "THE APPROXIMATE DATES OF MOON PHASES FOR "'; W,'" ARE:" ; PRINT "PHASE/DATE"
1640 IF W = 1 OR W = 10 THEN X = "WAXING CRESCENT" WAXING CRESCENT FULL MOON"
1650 IF W = 2 OR W = 20 THEN X = "NEW MOON" WAXING CRESCENT FULL MOON"
1660 IF W = 3 OR W = 30 THEN X = "WAXING CRESCENT" WAXING CRESCENT FULL MOON"
1670 IF W = 4 OR W = 40 THEN X = "FULL MOON" WAXING CRESCENT FULL MOON"
1680 IF W = 10 THEN X = "WAXING CRESCENT"
1690 IF W = 20 THEN X = "NEW MOON"
1700 IF W = 30 THEN X = "WAXING CRESCENT"
1710 IF W = 40 THEN X = "FULL MOON"
1720 IF W = 70 THEN 1740
1730 FOR I = 1 TO 6 PRINT X(1),X(15),14,15,,/",Q(1) I NEXT I CTO 1750
1740 FDO I = 1 TO 4 PRINT X(1),X(15),14,15,,/",Q(1) I NEXT I
1750 CHAN "$MEN DISCl";
1760 DATA FOR FIRST 12 PERIODS OF SIXTEEN TERMS:
1770 MOON DATA FOR TERMS 0 - 7, LINES 1,3,5, AND 7
1780 MOON DEC FOR TERMS 0 - 7, LINES 2,4,6, AND 0
1790 FOR JANUARY:
1800 DATA 1950.697,1392,2494.4,7149,1,0.019,-1,3222,-0.041,1976,-0.365
1810 DATA -3.875,2,962,0,1968,-1.927,-3374,3,164,-0.239
1820 DATA 1512.978,2,1959,0,877,6,3971,-2.7110,4453,5095,1,491,-0.216
1830 DATA 12.612,19,3377,-2.9301,-2.6935,-7.391,-1616,1222,853
1840 DATA 1401.7515,1386,0,697,7,8169,-0.0841,0,1056,9104,3125,-0.3131
1850 DATA 14.5721,-0.1036,0,7123,4,9170,8,543,7,845,1066,-0.8362
1860 DATA 1664,3561,1309,577,-1,1272,2,0810,8073,-4597,1409,0797
1870 DATA -24.0059,-6.7991,0.9414,0,610,7,801,1,1219,0.0444,0.0213
1880 FOR FEBRUARY:
1890 DATA 1503.2810,1400,0.094,2,3500,-1,9301,-3.050,102,0.344,0.017
1900 DATA 9,3321,10,9236,2,7242,-1,7205,142,0,835,-0.326,0.082
1910 DATA 1494.3531,1382,9105,6,9639,3,6709,4,4793,-4,4741,-0.5055,0.039
1920 DATA 24.7242,5,010,0,11,0005,3,1141,1,1612,0.001,0.07,0.07,0.07
1930 DATA 1743,5005,1391,4612,61,-1,2693,5491,3460,-80,0.014
1940 DATA -7.1409,23,6776,4,4402,0,1930,7,645,0,781,0.322,-0.33
1950 DATA 1644,0099,19,0762,-0.04,-0.14,0.041,0.08,0.08,0.08
<table>
<thead>
<tr>
<th>Year</th>
<th>Data</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970</td>
<td>1759.35641, 1499.35641</td>
<td>-1.7492, 0.1049, -0.077, -0.024, 0.072</td>
</tr>
<tr>
<td>1980</td>
<td>1977.6145, 1399.35641</td>
<td>1.5777, 0.2934, 0.0537, 0.000, 0.010</td>
</tr>
<tr>
<td>1990</td>
<td>2020.3789, 1530.1234</td>
<td>1.5734, -0.2934, -0.0537, 0.000, -0.010</td>
</tr>
<tr>
<td>2000</td>
<td>2080.3789, 1599.35641</td>
<td>1.5797, -0.2934, -0.0537, 0.000, -0.010</td>
</tr>
<tr>
<td>2010</td>
<td>2140.3789, 1659.35641</td>
<td>1.5859, -0.2934, -0.0537, 0.000, -0.010</td>
</tr>
<tr>
<td>2020</td>
<td>2200.3789, 1719.35641</td>
<td>1.5921, -0.2934, -0.0537, 0.000, -0.010</td>
</tr>
<tr>
<td>Date</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>2640 DATA 18.632,16.5073,-1.5614,-2.4766, -2.006, 116, 6737, -004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2650 DATA 1667.4794,130.7195,3.0567,-1.3641,-1.0837,4.924, 1120, -0045</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2660 DATA 13.1807,-23.1656,-0.9943,3.7114, 7417, 1571, -0051, -011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2670 DATA 1561.2979,130.1831,3.2936,3.9844, -7066,-1.0591, 1616, 1610</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2680 DATA -26.5635,1.5412,12.5639,-1.8169,-1.4106,4.4804, 1302, -0682</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2690 DATA FOR NOVEMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2700 DATA 1481.0160,140.527,605,-2.4076, 0328,-0.0506,-0196, 0165</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2710 DATA -3.2735,21.0512,1.8681, -1.6591,-0.2597, 1586, 0681, 020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2720 DATA 1751.1928,130.825,-1.7277,3.7418, -0249,-0.0125, 0672, 1642</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2730 DATA 20.6394,-465,-1.8156,-6.751,1.2616, 147,-139,-0212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2740 DATA 1649.4163,130.853,-4.262,-1.9732,1.2611, 6079,-0315,-0083</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2750 DATA -19.0935,24.0168,4.4946,4.261, 6252,-3484,-1244,-0623</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2760 DATA 1541.8447,139.6763,7.0573,-1.7195,-1.101, 6523,-0114,-0627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2770 DATA -19.1407,15.9411,5.0256,-2.7344, 4990,-0101,-1422,-0256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2780 DATA FOR DECEMBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2790 DATA 1401.6577,139.2502,-4.2324,-2.3219, 3609, 2527,-0079,-0063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2800 DATA 10.0023,19.9927,-2.6355,-2.1368,-4263,-0002, 0697,-0020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2810 DATA 1745.7654,130.9731,4.6735, 5151,-2.0277, 564, 2988,-1162</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2820 DATA 20.6523,-16.1953,-9.4100,2.9479, 6085,-4204, 8436,-0079</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2830 DATA 1641.5314,130.1566,-4.4209,2.9322, 2.3456,-4208,-0711,-0073</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2840 DATA -23.1024,-13.2761,11.5975,2.0067,-1.412,-6415, 1165,-1216</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2850 DATA 1551.2149,139.0025,4.7572,-2.5992,-3171,-29,-1149,-1133</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2860 DATA -11.63,19.7582,2.8291,-2.1343, 584,-1035,-0520,-0255</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4264  | SUNRISE/SUNSET/TWILIGHT TIMES (VALID FOR APPROX 60degN TO 60degS LAT)
4271  |
4280  | DIM WH(211)
4281  | GCLR @ GRAPHICAL SCALE 0,0
4282  | CSIZE 9
4283  | MOVE 10,55 @ LABEL "SUNRISE/SUNSET/TWILIGHT TIMES"
4284  | CSIZE 5 @ MOVE 35,49 @ LABEL "(VALID FOR LATITUDES 60degN TO 60degS)"
4285  | MOVE 45,9 @ LABEL "PRESS 'CONT' WHEN READY..." @ PAUSE
4296  | GCLR
4297  | PRINT "AIU", 4458* FOR I=1 TO 32 @ PRINT "SUN", @ NEXT I @ PRINT "SUN", @ PRINT "AIU"
4298  | PRINT "AIU", 4458*, "SUNRISE/SUNSET/TWILIGHT TIMES", 4464* @ PRINT "AIU" @ ALPHA
4299  | PRINT "(VALID FOR LATITUDES 60degN TO 60degS)"*, "AIU" @ PRINT "RISE")
4300  | DEC @ CLEAR @ DISP "ENTER THE MONTH, DAY, YEAR (e.g. 10,20,1982)" @ INPUT MONTH, DAY, YEAR
4301  | PRINT USING 4320, "I, MONTH, DAY, YEAR"
4302  | IMAGE "DATE", 20",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",","
4085 IF T1=T2 THEN GOTO 4038 1 T1=RISETIME T2=SETTIME
4086 IF T1<T2 THEN GOTO 4015
4087 SPAN=ABS (T1-T2) GOTO 4020 / TIME SPAN COMPUTED
4088 SPAN=ABS (T2-T1) GOTO 4020 / TIME SPAN COMPUTED
4089 PRINTOUT FOR TIME OF PHENOMена
4090 U=ABS (T1) V =H2+ABS (T2+1) / SI=SPAN*1
4091 N=ABS (ABS (T1) MOD 10460) +N2=ABS (ABS (T2) MOD 10460) +N2=ABS (SPAN MOD 51860)
4092 IF T1<0 THEN B1=** ELSE B1=**
4093 IF T2<0 THEN B2=** ELSE B2=**
4094 H1=H1 MOD 24
4095 H2=H2 MOD 24
4096 PRINT USING 4908 ; W1, H1, H2, SC, HC, MC, SI, SI
4097 IMAGE 23A.A, A2, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", 

^---```
10 I OPTS458 DISC.
20 I CONTAINING MENU OPTIONS 4=DENSITY ALTITUDE, 5=ATMOSPHERIC STABILITY, and 8= SUNRISE/SUNSET/TWILIGHT TIMES
40 PRINTER IS 584,132
45 DIM D$1800,1800
50 ON X GOTO 60,2930,4250
60 I
70 I DENSITY ALTITUDE
80 I
81 GCLEAR @ GRAPH @ MSIZE 0,0
82 CSIZE 9
84 MVAL 15.55 @ LABEL "DENSITY ALTITUDE PROGRAM"
85 CSIZE 5 @ MOVE 33,9 @ LABEL "PRESS 'CONT' WHEN READY..."
86 PAUSE
87 GCLEAR
88 PRINT " A1U" @ FOR I=1 TO 11 @ PRINT "DENSITYALT=":; NEXT I @ PRINT "DENSITYALT=": &15U"
90 ON ERROR GOTO 98 @ CLEAR @ DISP "STATION ELEVATION (feet)",; INPUT ELEV
100 ON ERROR GOTO 100 @ CLEAR @ DISP "TEMPERATURE (degrees F)",; INPUT TEMP
110 DISP @ DISP "DEW POINT (degrees F)",; INPUT FDEWPT
120 CDEWPT=5*(FDEWPT-32)/9
130 ON ERROR GOTO 130 @ CLEAR @ DISP "ENTER THE UNCORRECTED STATION PRESSURE (inches of mercury). ENTER A '0' IF ONLY THE PRESSURE AT ALTITUDE IS KNOWN."
140 DISP @ INPUT BP @ BAROMETRIC PRESSURE
150 IF BP=0 THEN GOSUB 570 ELSE GOSUB 710
160 BP=INT (BP*100)/100 @ PA=INT (PA)
170 ICEPT=273.16
180 STDATM=1013.246
190 STEAMPT=373.16
200 IF CDEWPT=0 THEN GOTO 330
210 CDEWPT=CDEWPT+ICEPT
220 D=-(7.9929/S*(STEAMPT/CDEWPT-1))
230 E=5.029886LT (STEAMPT/CDEWPT)
240 R=1-CDEWPT/STEAMPT
250 F=1.39161214-7*10^(-11.4444R-1)
260 S=STEAMPT/CDEWPT-1
270 C=0.13323*10^(-310^(-(-3.49149S)-1))
280 H=LGT (STDATM)
290 I=DE+FG+H
300 Z=10^I
310 SUP=Z*0.2953
320 GOTO 418
330 D2=CDEWPT+ICEPT
340 D=-(9.0971/S*(ICEPT/D2-1))
350 F=-3.565746MGT (ICEPT/D2)
360 G=LGT (6.1071)
370 I=DE+FG+H
380 Z=10^I
390 SUP=Z*0.2953
410 CLEAR
420 T=(-0.02#SUP/(BP-SUP)) @ MIXING RATIO CALCULATION
430 I=(TEMP*459.69)/(1+1.614/F) @ VIRTUAL TEMPERATURE CALCULATION
440 DA=INT ((1-(17.326#BP/^1/.2351)/145366) @ DENSITY ALTITUDE FORMULA: SHMITHLIN HET TABLES
450 PRINT " %-15J @ PRINT " &13D"
460 SET 1/0 7,16,1
470 PKINI "DENSITY ALTITUDE CALCULATIONS:" @ PRINT " 
480 SET 1/0 7,16,2
490 PKINI "A1OS:" @ PRINT " A16D"
500 PRINT "STATION ELEVATION = ";ELEV, " feet"
510 PRINT "TEMPERATURE = °F, degrees fahrenheit"
520 PRINT "DEW POINT = °F, degrees fahrenheit"
530 PRINT "PRESSURE = °PSI, inches of mercury"
540 PRINT "PRESSURE ALTITUDE = °F, feet"
550 PRINT "&14D" @ PRINT " &AS"
560 PRINT @ PRINT "YOUR DENSITY ALTITUDE IS A/D, feet" @ GOTO 880
570 I
580 I DETERMINES PRESSURE ALT FROM BAROMETRIC PRESSURE
590 PA=PA*33.8639 @ CONVERTS INCHES HG TO MILLIBARS
610 IF PA=1013.25 THEN PA=26659.7342246-26.518500A @ GOTO 640
620 IF PA=760.69 THEN PA=50125.7923317-79.0246642072PA = 830000325969PA * 2 @ GOTO 640
630 PA=42682.3525246-57.4057111468PA = 51.5262796666-28PA * 2
640 IF PA<5000 THEN C=-1.47634664655-0.2646985PA + 0.001305643PA * 2 @ GOTO 690
650 IF PA<9000 THEN C=-1.175407305PA = 3.0000001622PA * 2 + 0.0000001622PA * 2-3.704655-13SA4PA * 4 @ GOTO 690
660 IF PA<3500 AND PA>9700 THEN C=11.747041.544653-30.19740PA = 0.140730PA * 2-0.000001456063PA * 3 @ GOTO 690
670 IF PA<9700 THEN PA=1273.730911.26668PA = 0.000010826PA * 2 + 0.05324E-10PA * 3
680 PA=PA+C
700 RETURN
710 I
720 I DETERMINES BAROMETRIC PRESSURE FROM PRESSURE ALT
730 I
740 CLEAR @ DISP "PRESSURE ALTITUDE (feet)" @ INPUT PA
750 BP=1813.0096413-3.683108562322-28PA + 0.3865310352633E-7PA * 2
760 C=-1.46342+0.005569578PA = 0.000001601921PA = 2+3.46716E-12PA * 3
770 BP=BP-C
780 BP=BP/33.8639 @ CONVERTS MILLIBARS TO INCHES HG
790 RETURN
800 I
810 I HELICOPTER LOADING CAPACITIES
820 I
830 CLEAR @ DISP "DO YOU WANT LOAD CAPACITIES FOR SELECTED HELICOPTERS AT THE COMPUTED DENSITY ALTITUDE (Y/N)" @ INPUT I
840 IF I=I THEN PRINT " &11V" @ CLEAR @ DISP "PLEASE WAIT" @ CHAIN "MENU.DISCI"
850 DIM HELIYP(150)
860 TEMP=(TEMP-32)*5/9 @ TEMP NOW IN DEG C
870 DO "" ENTER THE NUMBER OF THE HELICOPTER (1-14) "" @ DI="" "" FOR I=1 TO LEN (DO) @ DI=DI&CHR (NUM (DO(I,1))<128) @ NEXT I
880 ON ERROR GOTO 880 @ CLEAR @ DISP TAB (19),DI & DISP
890 DISP TAB (15): 1. AH-1G
900 DISP TAB (15): 2. AH-1G
910 DISP TAB (15): 3. CH-47A
920 DISP TAB (15): 4. CH-47C
930 DISP TAB (15): 5. CH-47G
940 DISP TAB (15): 6. CH-54A
950 DISP TAB (15): 7. CH-54B
960 ON H DOSUB 110,1200,1350,1470,1620,1770,1950,2620,2140,2260,2300,2590,2620,2750
970 PRINT " &A1S"," &A5V" @ SET I/O 7,16,1
980 PRINT "HELICOPTER LOADING CAPABILITY" @ PRINT " "
990 SET I/O 7,16,2 @ PRINT @ PRINT " " &A5S"," &A1D"
1000 TEMP=INT (TEMP/100)/100 @ PA=INT (PA) @ DA=INT (DA) @ WT=INT (WT/10)/10
1010 PRINT "HELICOPTER TYPE"," HELIYP"
1020 PRINT "OAT = °C, degrees Centigrade"," Pressure Altitude = °PA,
1030 PRINT "Approximate Maximum Allowable Load (Cargo, Fuel, and Passenger Mix) = °WT," LBS"
1040 PRINT " &A2S"," (ALLOWABLE LOAD ASSUMES AIRCRAFT BASIC WEIGHT = °CRAF"," LBS, OIL = °OIL,
1050 PRINT " &A1S", " °LBS," @ PRINT " °LBS," @ PRINT " °OIL"
1060 PRINT "DO YOU WANT LOAD CAPACITIES FOR OTHER HELICOPTERS AT THE SAME DENSITY ALTITUDE (Y/N)" @ INPUT B
1080 IF B=I THEN GOTO 880
1090 PRINT " &A1S"," &A5V" @ FOR I=1 TO 11 @ PRINT "DENSITYALT"," @ NEXT I @ PRINT "DENSITYALT"
1090 CLEAR @ DISP "PLEASE WAIT" @ CHAIN "MENU.DISCI"
CH-54A (TM 55-1520-217-10-1)  DATE: 6 APRIL 1977

2600  CRAFT=21895 @ OIL=15 @ CREW=2 @ HELTYPE="CH-54A (T73-P-1), TWO-ENGINE OPERATION" @ MAXWT=42000 @ MAXTORQUE=81.5 @ PERCENT
2610  IF TEMP>.-11 THEN 2640
2620  A=34707.769480-78.79067*TEMP +760A*TEMP^2
2630  B=-296.62085-2033*TEMP-0.8579*TEMP^2 & GOTO 2660
2640  A=35613.1609-1.22379*TEMP +56065*TEMP^2
2650  B=-296.125-1.66*TEMP-0.07690*TEMP^2
2660  TORQUE=(PA-A)/B @ IF TORQUE>MAXTORQUE THEN TORQUE=MAXTORQUE
2670  YSCALE=.05*TORQUE-1
2680  A=98398+.004560*TEMP+.00002996*TEMP^2
2690  B=-935-.0012610*TEMP
2700  Y=9*YSCALE+B
2710  A=25925.7756-1.67678*DA-.0000061351*DA^2
2720  B=4199-.0000075284*DA-.5.428E-11*DA^2
2730  GROSSWT=AY*5 & IF GROSSWT>MAXWT THEN GROSSWT=MAXWT
2740  WT=GROSSWT-(CRAFT-OIL+CREW=200)
2750  GOTO 1790
2760  CH-54B (TM 55-1520-217-10-2)  DATE: 15 APRIL 1977

2770  CRAFT=22625 @ OIL=15 @ CREW=2 @ HELTYPE="CH-54B (T73-P-700), TWO-ENGINE OPERATION" @ MAXWT=47000 @ MAXTORQUE=100 @ PERCENT
2780  IF TEMP>.-14 THEN 2820
2790  A=94594.5094-3884.3305*TEMP-184.65234*TEMP^2-3.56638*TEMP^3-8.2472*TEMP^4
2800  B=-21792.1913473.1486*TEMP +41736*TEMP^2+.17233*TEMP^3 & GOTO 2840
2810  A=121998.47092-95.3369*TEMP-7.31005*TEMP^2
2820  B=-26809.6487-30.9293*TEMP +84051*TEMP^2+.10238*TEMP^3-0.0172256*TEMP^4
2830  TORQUE=EXP ((PA-A)/B) @ IF TORQUE>MAXTORQUE THEN TORQUE=MAXTORQUE
2840  YSCALE=.05*TORQUE-1
2850  A=98398+.004560*TEMP+.00002996*TEMP^2
2860  B=-935-.0012610*TEMP
2870  Y=9*YSCALE+B
2880  A=16542.1559+.1629DA-.0000029365*DA^2-.000000006790*DA+3
2890  B=14187.5979-33958DA-.0000099711*DA^2+2.2315E-9*DA^3
2900  C=1253.1072+.03978DA-.0000010685*DA^2+.2325E-10*DA^3
2910  GROSSWT=AY*5 & IF GROSSWT>MAXWT THEN GROSSWT=MAXWT
2920  WT=GROSSWT-(CRAFT-OIL+CREW=200)
2930  GOTO 1790

2940  OH-6A (TM 55-1520-214-10)  DATE: 17 DECEMBER 1976

2950  CRAFT=1050 @ OIL=6 @ CREW=2 @ HELTYPE="OH-6A (T63-A-5A-700)" @ MAXWT=2550 @ MAXTORQUE=80.3 @ PSI
2960  A=10277.7233-304.236*TEMP
2970  B=-2196.807998-34.05994213*TEMP+1906795523*TEMP^2-0.00077529E-3*TEMP^3
2980  TORQUE=EXP ((PA-A)/B) & IF TORQUE>MAXTORQUE THEN TORQUE=MAXTORQUE
2990  YSCALE=.2*TORQUE-0
3000  A=1587.6791-818613*DA-.00000470*DA^2
3010  B=184.4292-.000666666*DA-.0000007266*DA^2
3020  GROSSWT=AY*5 & IF GROSSWT>MAXWT THEN GROSSWT=MAXWT
3030  WT=GROSSWT-(CRAFT-OIL+CREW=200)
3040  GOTO 1790

3050  OH-5DA (TM 55-1520-229-19)  DATE: 7 APRIL 1970

3060  CRAFT=1725 @ OIL=13 @ CREW=1 @ HELTYPE="OH-5D (T63-A-700)" @ MAXWT=3000 @ MAXTORQUE=78 @ PSI
3070  A=100574.314-335.296*TEMP
3080  B=-21792.1913473.1486*TEMP +41736*TEMP^2+.17233*TEMP^3
3090  TORQUE=EXP ((PA-A)/B) & IF TORQUE>MAXTORQUE THEN TORQUE=MAXTORQUE
3100  YSCALE=.2*TORQUE-0
3110  A=1566.74206-.0044052DA-.00000124579*DA^2
3120  B=-179.227-0.03480A
3130  GROSSWT=AY*5 & IF GROSSWT>MAXWT THEN GROSSWT=MAXWT
3140  WT=GROSSWT-(CRAFT-OIL+CREW=200)
3150  GOTO 1790

3160  OH-5BC (TM 55-1520-235-10)  DATE: 7 APRIL 1970
<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Type</td>
<td>CRAFT</td>
</tr>
<tr>
<td>Oil Type</td>
<td>SAE 10W-30</td>
</tr>
<tr>
<td>Oil Pint</td>
<td>15</td>
</tr>
<tr>
<td>Coolant Type</td>
<td>WATER</td>
</tr>
<tr>
<td>Coolant Pint</td>
<td>2 OUNCES</td>
</tr>
<tr>
<td>RPM</td>
<td>1500-2500</td>
</tr>
<tr>
<td>Coolant Temp</td>
<td>180-190°F</td>
</tr>
<tr>
<td>Air Temp</td>
<td>70°F</td>
</tr>
<tr>
<td>Fuel Type</td>
<td>87 OCTANE</td>
</tr>
<tr>
<td>Fuel Pint</td>
<td>10</td>
</tr>
<tr>
<td>Battery Size</td>
<td>950 CCA</td>
</tr>
<tr>
<td>Battery Pint</td>
<td>1000</td>
</tr>
<tr>
<td>Battery Type</td>
<td>SLA</td>
</tr>
<tr>
<td>Battery Voltage</td>
<td>12V</td>
</tr>
</tbody>
</table>

** Torque-Max Torque **

- Cylinder Heads: 180-200 ft-lb
- Crankshaft: 300-350 ft-lb
- Flywheel: 500-550 ft-lb

** Fuel Efficiency **

- Gasoline: 25-30 miles per gallon
- Diesel: 15-20 miles per gallon

** Maintenance Schedule **

- Oil Change: Every 3,000 miles
- Filter Change: Every 15,000 miles
- Coolant Change: Every 2 years
- Battery Check: Every 6 months

** Safety Tips **

- Wear proper safety gear when working on the car.
- Keep the area around the car clear of debris.
- Use a secure jack when working under the car.
- Test brakes before driving after any repair work.

** Troubleshooting **

- Check for leaks in the coolant system.
- Check for clogged air filters.
- Check for worn spark plugs.
- Check foriat drive belts.

** Contact Information **

- Service Department: 555-5555
- Parts Department: 555-5556
- General Manager: 555-5557

** Additional Notes **

- Regular maintenance is key to extending the life of your vehicle.
- Always use high-quality parts when repairing your vehicle.
- Regular checks can prevent costly repairs later on.