IKS INTERNSHIP PROGRAM

PROJECT REPORT

ON

Relation between the ancient calendar and Suryasiddhanta based

SUBMITTED BY

Prashant Sharma

BSMS Math Major-4th Year

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH

THIRUVANANTHAPURAM

UNDER THE GUIDANCE OF

Dr. Anil Kumar Gourishetty

Associate Professor Department of Physics

IIT Roorkee, Uttarakhand.

14/08/2022

Contents

Abstract

The Sun, Moon, planets, and other celestial bodies appear to be resting on the surface of a massive sphere known as the Celestial Sphere. The celestial sphere seems to spin westward around Earth in 24 hours due to Earth's eastward revolution around its axis. The Celestial Equator is created by infinitely stretching the Earth's equator into space until it seems to cross the celestial sphere and form a circle

The purpose of the project was to Reviewing Spherical Trigonometry theorems and methods using books and then verify the results to solve the position of the celestial bodies in the current phase of the project we are solving from the general problems and ideal conditions such as considering every orbit to be spherical and follows the spherical geometry.

The Scope of the project is to find the relation between ancient calendar and the modern day calculation using computation and ancient methods suryasiddhanta is one of the method that we will see latter on the project

The Scientific Software MATLAB did verification of the different methods of the spherical geometry.

Preliminary results

In current phase of the project all the efforts were made to solve the essential problems to understand the spherical trigonometry in terms of degree decimals format.

the Basic Idea is to calculate the time and date by observing the position of celestial bodies in the current phase of the project using the point coordinates to refer to the celestial bodies is essential for learning the concept.

Problem-solving and verification using the computational software MATLAB were done and all the data and solutions are attached in the end of the appendix. At this moment we are only able to visualize the solution in text format. Latter on it will be graphical format.

Introduction

the project was conducted to figure out the relation between the ancient algorithms to predict the date and time in the calendar a good spherical trigonometry background is a must.
So, we started with learning the basics of spherical geometry.
Then we solved for the spherical trigonometry in deep derived all the relations and formulas required and found all the methods. These all are the basic prerequisite for the project. And our first month leads us to prepare the basic tools we need to compare our main goal, i.e., to find a calendar for a given period of time and compare it with the ancient method for the calendar.
As we move, we figure out that the problems in the real world are with more complicated numbers and consist of thousands of data points. So we need a program to solve this problem using computers as we have very less time to solve it manually. So a MATLAB script was built to solve the problems, and verification for the manual calculation was done.
These tools will help us to solve the position of the stars and the celestial bodies in the latter stage of the project.
All of the work was done with the help of the INTERNET, which allowed us to find many resources such as books and software. To learn the concepts, we need not go outside, and a computer was a must to solve all the problems, so the project is still going on the computers under the guidance of our mentor.

Background(250)
In the early Indian age, the Vedas and Purana were used as references for keeping records of the time and position of the stars.
We use calendars daily to see our schedule, recall significant occasions, and, most importantly, reflect on the past and plan for the future. We have been doing this since the dawn of civilization for humans. There is extensive written evidence from before humans began counting days, but the calculation has always been accurate. Ancient Indian astronomers are still the pride of the whole world. The introduction of astronomical calculations based on spherical trigonometry marked the beginning of a new age of siddhantic astronomy, then Mayasura's Surya Siddhanta later revolutionized Indian astronomy. Mayasura observed rare conjunction of the sun and the moon, and all planets except their nodes and subsidies were in conjunction with Aries (Mesha) on the new moon day of Chaitra month at the end of the 28th Krita yuga. Software simulations using the JPL Horizons (a web application) and the ephemeris system established that such conjunction on Chaitra Shukla Pratipada occurred only once in the last 16000 years, that is, on 22nd February 678 BCE. It is thus established that Mayasura wrote Surya Siddhanta in 6778 BCE.
The calendar of Surya Siddhanta completes 8800 years in the year 2022.
All these calculations were done way back in time without using computers or advanced astronomical equipment. If our ancestors could do this back in time, why not us? So, this project aims to figure out how Surya Siddhanta can be helpful in quickly figuring out the dates and then verifying their accuracy and relatedness with the modern calendar.
The first requirement for understanding the concepts in Surya Siddhanta is to learn Spherical Trigonometry, which is the initial step of this project and then we would explore Surya Siddhanta in depth. As we know, Astronomy is a field of study that encompasses everything outside of the surface of the earth. Every planetary motion and the solar event takes place in three-dimensional physical space, and most planets and their orbits are spherical or elliptical. However, certain planets have odd but well-defined orbits. With the aid of contemporary technology, it is possible to correctly predict objects' past and future positions by treating these motions as motion in spherical dimensions. We would calculate and verify the calendar system by determining the planetary motions and positions. Based on Mayasura's Surya

Siddhanta from 6778 BCE, Lata Deva wrote the Surya Siddhanta, now known as A Synopsis of Mayasura's Surya Siddhanta.The primary distinction between these two Siddhantas is that Lata Deva described a deva yuga of 4 lakh 32,000 years, whereas Mayasura described an asura yuga of 1 lakh 80 000 years.Surya Siddhanta was written by Lata Deva on Chaitra Prapada, or February 18th, 3101 BCE, while observing the approximate conjunction of the planets. Here we will figure out which of the Siddhanta is more accurate. Learning and solving spherical trigonometry will help us see how some Ancient civilizations had the concept of periods that were truly enormous by contemporary standards at that point of time. Apart from the Mayans, the ancient Hindus appear to be the only people who dared to think beyond a few thousand years.

Description of Project

Suryasidhanta is still one of the methods that we can fit on many of the spherical trigonometry problems perhaps it's an ancient method to calculate the position of the stars it works.
Here we have solved the problems and understand the derivation of the formulas that are given in the book by W.M. Smart. Based on the formulation, algorithms have been made to solve the problems for different conditions. All the problems were done using the normal user computer, and manual verification using simple calculator was done.
There’s a lot of difference between the manual calculations and the pre-programmed solution we are getting after attempting eight times to solve the problems using manual methods, it was still not as accurate as it should be, but in the end, we found that all the errors were due to unit conversion that is very difficult for a normal human so only the MATLAB program was giving the right and accurate solution here is the output we are getting in the FIGURE-1.

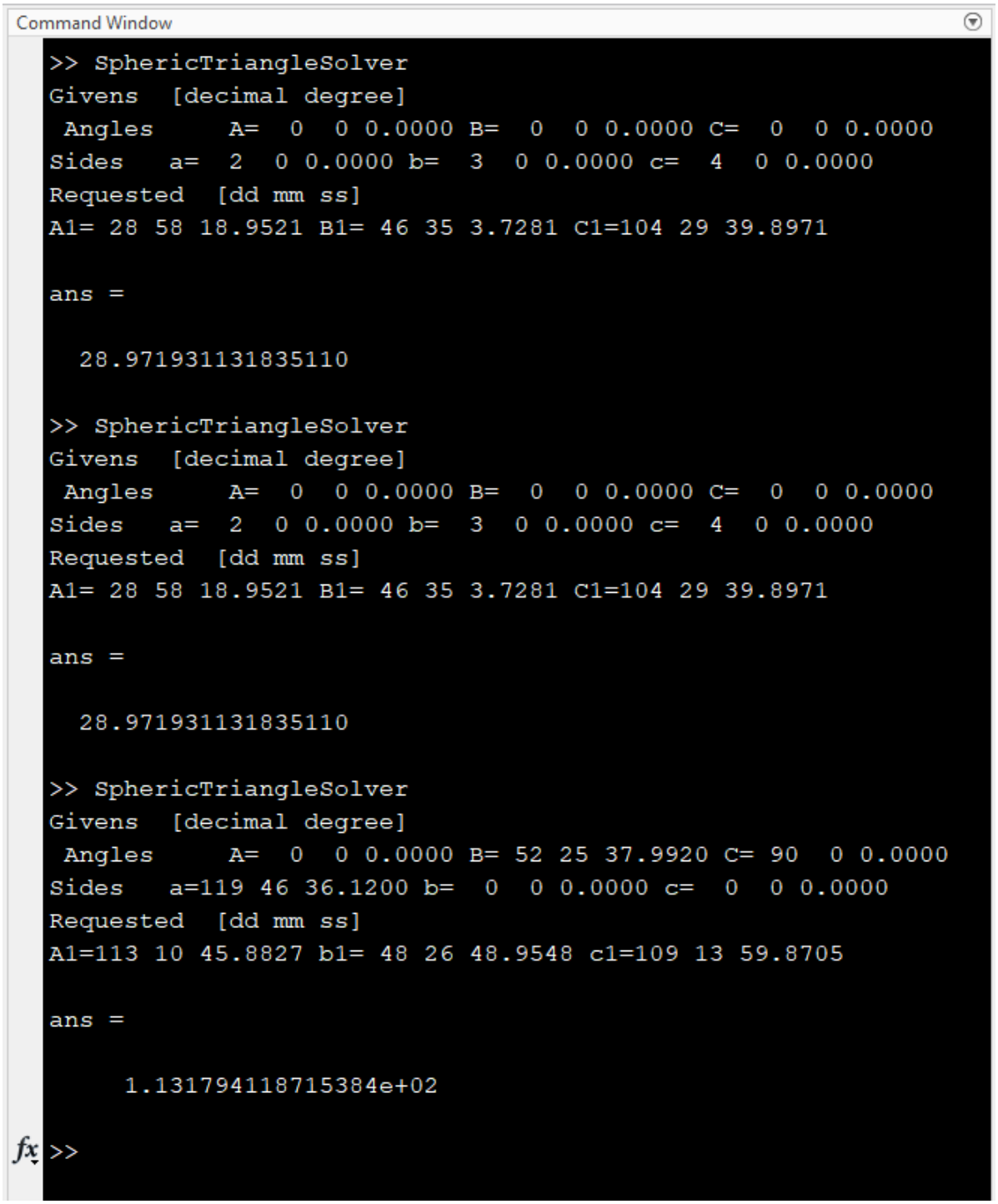

FIGURE-1

The most difficult part was writing the algorithm for all the conditions of spherical trigonometry problems.
So if we go in detail
A spherical triangle is made up of the area enclosed by the three great circular arcs intersecting pairwise in three vertices on the surface of the sphere Figure-2

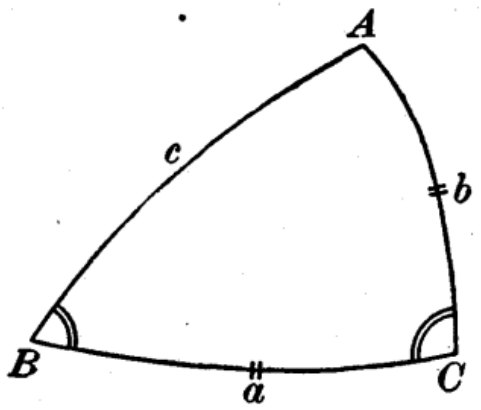

Figure-2

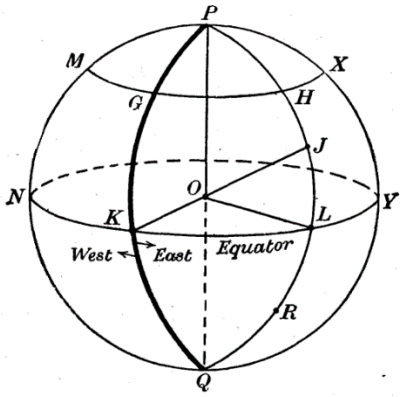

Figure-3

These are the derivations and topics covered in the past three weeks

- Intro to spherical trigonometry
- Spherical triangle
- Length of a small circle arc
- Terrestrial latitude and longitude
- The fundamental formula of trigonometry (Figure-3)
- The sine-formula
- Cosine-formula
- The Four-parts formula
- Right angled and quadrilateral triangle
- Polar formula
- Numerical example
- Haversine formula
- Trigonometrical ratios for small angles
- Delambre's and Napier's analogies.

Along with solving the problems, verification was done using coding on the MATLAB program one of the interesting problems was this
one.

- **4. Two ships X and Y are steaming along the parallels of latitude 48° N and 15° S respectively, in such a way that at any given moment the two ships are on the same meridian of longitude. If the speed of X is 15 knots,* find the speed of Y.**

Here is one of the most important practical examples we can see in ancient times. When voyagers used to travel through the sea horizon, the only way to find their position and the time elapsed was by knowing the position of the stars. The same information helps them to determine the speed of the boat, and they can determine the end of their journey.

Methods

haversine(θ) = sin²(θ/2). The haversine formula is a very accurate way of computing distances between two points on the surface of a sphere using the latitude and longitude of the two points.

$$\text{hav}\,\theta = \tfrac{1}{2}(1-\cos\theta) = \sin^2\frac{\theta}{2}$$

Using the above equation, algorithms were made to solve any spherical trigonometry problem. A spherical triangle has 6 basic elements: three angles (A, B, C) and three sides (a,b,c).
FIGURE-4

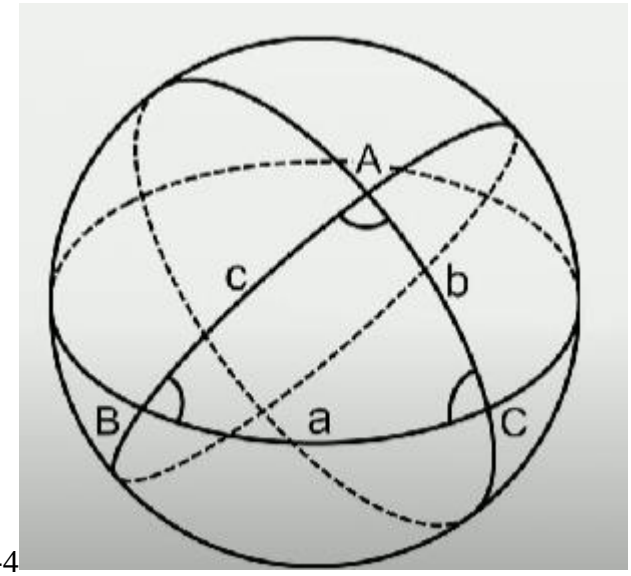

Figure-4

Given any three of the 6 basic elements, the function finds the other three. Usage of function for example, Given 3 elements of a spherical triangle:
a=2, b=3, C=104.4944158
[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver(A,B,C,a,b,c)
[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver(0,0,104.4944158,2,3,0)
NOTE That the unknown (A, B, c) 3-elements of the spherical triangle should be replaced with zero.
A spherical triangle has 6 basic elements: three angles (A, B, C) and three sides (a,b,c).
Given any three of the 6 basic elements, the function finds the other three.
So the written algorithm works as follows
It takes the Input inside the code as
INPUT:

- A,B,C,a,b,c element of spherical triangle. Only 3 values should be entered.
- The others must be zero.
- Inputs units must be [decimal degree]

And gives the OUTPUT in the terminal :

- A1, B1,C1,a1,b1,c1 are 1. solutions of the spherical triangle.

- A2, B2,C2,a2,b2,c2 are 2. solutions of the spherical triangle.
- Outputs units are degrees, minutes, seconds [dd mm ss]

Now going in further detail, here is algorithm that is working for now
Firstly the programm ask for the input in the format of ANGLES and sides. Angles and side must be given in degree decimal format. NOTE that the sides are nothing but the arc of the sphere so that is also amust be written in terms of degree decimals format
After that we store and assign the given values to some variables A, B, C, a , b and c.
These values are angles and arcs respectively. And then we look for the missing variables.
In order to solve for the all ather angles and sides we need to have atleast three of them known.
So the triangle have now different format of inputs such as SSS,SAS,ASA,AAS,AAA (S- side, A-angle) we classify them on the basis of these parameters and pass the argue to the function for solving the other values.
We have already derived and illustrated the formulas for the sine and cosine functions $\cos a = \cos b \cos c + \sin b \sin c \cos A$

$$\frac{\sin A}{\sin a} = \frac{\sin B}{\sin b} = \frac{\sin C}{\sin c}$$

Results
A working algorithm is designed to solve the spherical trigonometry problem and equip us for the future working of the project. Now we can use this algorithm to solve the position of the stars and verify the calendar with the one determined by the ancient suryasiddhante method

Discussion
Approximations for the small angles in sin and cosine functions leads to have a bit error in calculation that we need to consider if we looking for the accurate date and time calculations and aslo the four decimal point is not that accurate we need to have in rocket landing but it is okay for calculating the time by observing the position of the stars in the sky.

References
1. Textbook on Spherical Astronomy, Smart, W. M. Published by Cambridge University Press (1977) ISBN 10: 0521291801 / ISBN 13: 9780521291804
2. Bektas S. (2021), Jeodezi 1 Küre yüzeyinde uygulamalar, Atlas Akademi, ISBN:978-605-7839-87-9

Appendices

While preparing the function, Rody P.S. Oldenhuis subfunctions are used.

These are the things I have done with the python library for the calculations.
And plotting graphs a python library pygeodis and NumPy for prior calculations

Some failed attempts were also made to animate the points on the surface of the sphere. MANIM library for the animation and pictures to illustrate the problems and find solutions were used.
A brief part of the code to solve the spherical trigonometry
Problem is given here

```
This function solves any type of spherical triangle
% A spherical triangle has 6 basic elements: three angles (A,B,C) and three sides (a,b,c). % Given any three of the 6 basic elements, the function finds the other three.
%
% Usage of function
% for example, Given 3 elements of a spherical triangle: a=2, b=3,
C=104.4944158
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver(A,B,C, a,b,c)
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver
(0,0,104.4944158,2,3,0)

% Attention, the unknown (A,B,c) 3 elements of the spherical triangle should be replaced with zero.

% A spherical triangle has 6 basic elements: three angles (A,B,C) and three sides (a,b,c). % Given any three of the 6 basic elements, the function finds the other three.
%
%
% INPUT:
% A,B,C,a,b,c   element of spherical triangle. Only 3 values should be entered.
% the others must be zero. % Inputs units must be [decimal degree]



%OUTPUT:
%
%  A1, B1,C1,a1,b1,c1  are 1. solutions of the spherical triangle.
%  A2, B2,C2,a2,b2,c2  are 2. solutions of the spherical triangle.
% outputs units are degrees minutes seconds [dd mm ss]

% Usage of function
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver
(0,0,104.4944158,2,3,0)
% Attention, the unknown, desired 3 elements of the spherical triangle should be replaced with zero.

% While preparing this function, Rody P.S. Oldenhuis subfunctions are used.
function [A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] =
SphericTriangleSolver(A,B,C,a,b,c)
 format long
% ro=180/pi;
A=0;,B=0;,C=0;     a=2;,b=3;,c=4; %A=28.97193110;,B=46.5843689140;,C=104.4944158550;     a=0;,b=0;,c=0;
%A=0;,B=46.5843689140;,C=0;     a=2;,b=0;,c=4;
%A=0;,B=0;,C=104.49441585500;     a=0;,b=3;,c=4;
%A=0;,B=46.58436891400;,C=104.4944158550;     a=0;,b=0;,c=4;

%A=112.62944444;,B=65.30972222220;,C=0;     a=0;,b=70.874166666;,c=0;
%A=0;,B=0;,C=0;     a=73.701111110;,b=70.874166666;,c=7.0355555550;
%A=112.62944444;,B=65.30972222220;,C=0;     a=0;,b=70.874166666;,c=0;
%A=120.57333330;,B=0;,C=0;     a=105.4811111110;,b=75.7294444440;,c=0;
%A=112.62944444;,B=65.30972222220;,C=0;     a=0;,b=70.874166666;,c=0;
%A=60.46011667;,B=42.55760065;,C=80.6425750;     a=0;,b=0;,c=0;

%A=25;,B=0;,C=0;     a=0;,b=50;,c=70; %A=76.255;,B=115.795;,C=0;     a=0;,b=0;,c=81.4367777770;
%A=30;,B=0;,C=0;     a=70;,b=0;,c=80;

%a=148.573333;,b=142.1933333;c=0;     A=153.12666666;,B=0;,C=0;
%A=132.695;,B=107.1163889;,C=0;     a=146.345833333;,b=0;,c=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A1=0;A2=0;B1=0;B2=0 ;C1=0;C2=0;a1=0;a2=0;b1=0;b2=0 ;c1=0;c2=0;
 fprintf('Givens  [decimal degree]   \n') fprintf(' Angles     A=%3d %2.0f %6.4f B=%3d %2.0f %6.4f C=%3d %2.0f %
6.4f\n', degrees2dms(A), degrees2dms(B), degrees2dms(C))
```

```
fprintf('Sides   a=%3d %2.0f %6.4f b=%3d %2.0f %6.4f c=%3d %2.0f %6.4  f\n', degrees2dms(a), degrees2dms(b), de-
grees2dms(c)) fprintf('Requested  [dd mm ss]   \n')
A=A/ro;,B=B/ro;,C=C/ro;  a=a/ro;,b=b/ro;,c=c/ro;


%1.option KKK
  if (A==0 & B==0 & C==0)
[A1,B1, C1, A2,B2,C2] = sss(a,b,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(A1), degrees2dms(B1), de-
grees2dms(C1)) %fprintf('A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(A2), de-
grees2dms(B2), degrees2dms(C2))
 end
%2.option AAA
 if (a==0 & b==0 & c==0)
[a1, b1, c1, a2, b2, c2] = aaa(A,B,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('a1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(b1), degrees2dms(c1))
%fprintf('a2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(b2), de-
grees2dms(c2))
 end

%3.option KAK  [c1, A1, B1, c2, A2, B2] = sas(a, C, b)
 if (c==0 & A==0 & B==0)
[c1, A1, B1, c2, A2, B2] = sas(a,C,b); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('c1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(A1), de-
grees2dms(B1)) %fprintf('c2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(A2),
degrees2dms(B2))
 end  if (a==0 & B==0 & C==0)
[a1, B1, C1, a2, B2, C2] = sas(b,A,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('a1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(B1), de-
grees2dms(C1)) %fprintf('a2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(B2), degrees2dms(C2)) end
%[c1, A1, B1, c2, A2, B2] = sas(a, C, b)
 if (b==0 & A==0 & C==0)
[b1,C1, A1, b2, C2, A2] = sas(c,B,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('b1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f\n', degrees2dms(b1), degrees2dms(C1), de-
grees2dms(A1)) %fprintf('b2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f\n', degrees2dms(b2), de-
grees2dms(C2), degrees2dms(A2))
 end

%4.option AKA   [C1, a1, b1, C2, a2, b2] = asa(A, B, c)
 if (C==0 & a==0 & b==0)
[C1,a1, b1, C2 a2, b2] = asa(A,B,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('C1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(C1), degrees2dms(a1), de-
grees2dms(b1))
%fprintf('C2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(C2), degrees2dms(a2), de-
grees2dms(b2))
 end  if (A==0 & b==0 & c==0)
[A1, b1, c1, A2, b2, c2] = asa(B,C,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(A1), degrees2dms(b1), de-
grees2dms(c1)) %fprintf('A2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(A2), degrees2dms(b2),
degrees2dms(c2))
 end  if (B==0 & a==0 & c==0)
[B1,  c1,a1, B2,c2, a2] = asa(C,A,b);[B1,  c1,a1, B2,c2, a2] =[B1,  c1, a1, B2,c2, a2] *ro; fprintf('a1=%3d %2.0f %6.4f c1=%3d %2.0f
%6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(c1), degrees2dms(B1)) %fprintf('a2=%3d %2.0f %6.4f c2=%3d
%2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(c2), degrees2dms(B2))
 end

%5.option KKA
 if (c==0 & B==0 & C==0)
```

```
[c1, B1, C1, c2, B2, C2] = ssa(a,b,A); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(B1), de-
grees2dms(C1))
fprintf('B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(B2), de-
grees2dms(C2))
 end  if (a==0 & A==0 & C==0) [a1,A1, C1, a2, A2, C2] = ssa(b,c,B); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
  fprintf('C1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(A1), de-
grees2dms(C1)) fprintf('C2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(A2),
degrees2dms(C2))
 end  if (b==0 & A==0 & B==0)
[A1, B1, b1, A2, B2, b2] = ssa(c,a,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('b1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(b1), degrees2dms(A1), de-
grees2dms(B1)) fprintf('b2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(b2), degrees2dms(A2),
degrees2dms(B2))
 end

%6.option AAK
 if (C==0 & c==0 & b==0)
[b1, c1, C1, b2, c2, C2] = aas(A,B,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('c1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(b1), de-
grees2dms(C1))
fprintf('c2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(b2), degrees2dms(C2))
 end  if (A==0 & a==0 & c==0)
[A1, a1, c1, A2, a2, c2] = aas(B,C,b);
A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(A1), de-
grees2dms(a1)) fprintf('A2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(A2),
degrees2dms(a2))
 end  if (B==0 & a==0 & b==0)
[a1, B1, b1, a2, B2, b2] = aas(C,A,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('a1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(b1), de-
grees2dms(B1)) fprintf('a2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(b2),
degrees2dms(B2))
 end

%%%%%=================================================
%7.option  AKK
 if (b==0 & B==0 & C==0)
[c1, B1, C1, c2, B2, C2] = ssa(a,c,A); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2; fprintf('C1=%3d %2.0f %6.4f B1=%3d %2.0f
%6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(B1), degrees2dms(C1))
fprintf('C2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(B2), de-
grees2dms(C2)) end
if (c==0 & A==0 & C==0)
[c1, A1, C1, c2, A2, C2] = ssa(b,a,B); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('A1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(C1), de-
grees2dms(A1))
fprintf('A2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(C2), de-
grees2dms(A2)) end  if (a==0 & A==0 & B==0)
[a1, A1, B1, a2, A2, B2] = ssa(c,b,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('B1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(B1), de-
grees2dms(A1)) fprintf('B2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(B2),
degrees2dms(A2)) end

% 8.option  KAA
 if (c==0 & B==0 & b==0)
[c1, b1, B1, c2, b2,B2] = aas2(A,C,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
```

```
 fprintf('c1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(B1), de-
grees2dms(b1)) fprintf('c2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(B2),
degrees2dms(b2)) end  if (c==0 & a==0 & C==0)
[c1, a1, C1, c2, a2,C2] = aas2(A,B,b); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('c1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(c1), de-
grees2dms(C1)) fprintf('c2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(c2),
degrees2dms(C2)) end  if (b==0 & a==0 & A==0)
[b1, a1, A1, b2, a2,A2] = aas2(C,B,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;  a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('a1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(A1), de-
grees2dms(b1)) fprintf('a2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(A2),
degrees2dms(b2)) end

%a1, B1, c1degrees2dms function [a1, b1, c1, a2, b2, c2] = aaa(A, B, C)
%AAA  gives both solutions to the angle-angle-angle problem, in radians.
%
%   AAA(A, B, C) will result in NaNs if the existence condition
%   |pi - |A|-|B|| <= |C| <= pi - ||A| - |B||
%   is not met.
%


   % first solution     a1 = acos2( (cos(A) + cos(B).*cos(C)) ./ (sin(B).*sin(C)),  A );     b1 = acos2( (cos(B) + cos(A).*cos(C)) ./
(sin(A).*sin(C)),  B );     c1 = acos2( (cos(C) + cos(A).*cos(B)) ./ (sin(A).*sin(B)),  C );
   % second solution     a2 = 2*pi - a1;     b2 = 2*pi - b1;     c2 = 2*pi - c1;

   % check constraints     indices = ( ...         abs(pi - abs(A)-abs(B)) > abs(C) | ...         abs(C) > pi - abs(abs(A)-abs(B)) );     a1(indi-
ces) = NaN;    a2(indices) = NaN;     b1(indices) = NaN;    b2(indices) = NaN;     c1(indices) = NaN;    c2(indices) = NaN;
 end
function [b1, c1, C1, b2, c2, C2] = aas(A, B, a)
%AAS   gives both solutions to the angle-angle-side problem, in radians. %
%   AAS(A, B, a) may result in a vector filled with NaNs if the existence
%   condition |sin(B)sin(a)| <= |sin(A)| is not met. This function uses the
%   Middle Side Law function MSL.m and Middle Angle Law function MAL.m to
%   determine the solutions.
%

   % first solution     b0 = asin( (sin(B).*sin(a))./sin(A) );     b0(imag(b0) ~= 0) = NaN;
    b1 = mod(b0, 2*pi);     c1 = msl(a, b1, A, B);     C1 = mal(A, B, a, b1);

   % second solution     b2 = mod(pi - b1, 2*pi);     c2 = msl(a, b2, A, B);     C2 = mal(A, B, a, b2);



   % check constraints
   indices = ( abs(sin(B).*sin(a)) > abs(sin(A)) );     b1(indices) = NaN;    c1(indices) = NaN;     C1(indices) = NaN;    b2(indices) =
NaN;     c2(indices) = NaN;    C2(indices) = NaN;
 end

% Middle-angle-law function C = mal(A, B, a, b)
%MAL    Computes the missing angle in a spherical triangle, in radians.
%
%   MAL(A, B, a, b) is the implementation of the Middle Angle Law, and %   returns the missing angle C.
%
%   See also MSL, MALD.

   % sine & cosine of C
   % NOTE: denomenator not needed     sinC  =  sin(A).*cos(B).*cos(b) + sin(B).*cos(A).*cos(a);     cosC  = -cos(A).*cos(B) +
sin(A).*sin(B).*cos(a).*cos(b);

   % C is the arctangent of the ratio of these two
   C = mod( atan2(sinC, cosC), 2*pi);
 end

% Middle-side-law function c = msl(a, b, A, B)
%MSL    Computes the missing side in a spherical triangle, in radians.
%
```

```
%   MSL(a, b, A, B) is the implementation of the Middle Side Law, and %   returns the missing angular side c.
    sinc  = (sin(a).*cos(b).*cos(B) + sin(b).*cos(a).*cos(A));     cosc  = (cos(a).*cos(b) - sin(a).*sin(b).*cos(A).*cos(B));

   % c is the arctangent of the sine over the cosine     c = mod( atan2(sinc, cosc), 2*pi);
 end
function signedcos = acos2(alpha, beta)
%ACOS2      4-quadrant arccosine function, in radians.
%
%   ACOS2(alpha, beta) computes the four-quadrant arccosine of the amgle
%   [alpha]. For arguments |alpha| > 1, the result is NaN. The resulting
%   angle is not uniquely determined by alpha, nor by the lengths or
%   order of the sides of the triangle (as in ATAN2), so an additional
%   argument [beta] is required. If [beta] < pi/2, the small angle
%   (0 <= alpha <= pi/2) is returned. If [beta] > pi/2, the large angle %   (pi/2 < alpha < pi) is returned.
%
%   See also acos2d.



   H = 2*( mod(beta, 2*pi) < pi ) - 1;     H(~isreal(H)) = NaN;

   % compute signed arc-cosine     signedcos = H .* acos(alpha);

   % set complex results to NaN & take the modulus     signedcos(imag(signedcos) ~= 0) = NaN;     signedcos = mod(signedcos,
2*pi);

   % determine alphaues for zero-alphaued acos     ind1 = (signedcos == 0);     ind2     = (H < 0);                ind3     = (H > 0);     indi-
ces1 = ((ind1 + ind2) == 2);     indices2 = ((ind1 + ind3) == 2);     signedcos(indices1) = pi;          signedcos(indices2) = 0;
 end  function [C1, a1, b1, C2, a2, b2] = asa(A, B, c)
%ASA   gives both solutions to the angle-side-angle problem, in radians.
%
%   ASA(A, B, c) returns the missing values C, a, b. It uses the
%   four-quadrant arccosine function ACOS2 to determine these values.
%

   % first solution
   % NOTE: normal acos (in stead of acos2) is indeed correct.     C1 = acos( -cos(A) .*cos(B) + sin(A).*sin(B).*cos(c));     a1 = acos(
(cos(A) + cos(B).*cos(C1)) ./ (sin(B).*sin(C1)));     b1 = acos( (cos(B) + cos(A).*cos(C1)) ./ (sin(A).*sin(C1)));

   C1(imag(C1) ~= 0) = NaN;     a1(imag(a1) ~= 0) = NaN;     b1(imag(b1) ~= 0) = NaN;

   % second solution     C2 = 2*pi - C1;     a2 = mod(a1 + pi, 2*pi);     b2 = mod(b1 + pi, 2*pi);
 end function [c1, A1, B1, c2, A2, B2] = sas(a, C, b)
%SAS   gives both solutions to the side-angle-side problem, in radians.
%
%   SAS(a, C, b) returns the remaining unknowns of the spherical triangle,
%   [c1, A1, B1, c2, A2, B2].
%


   % first solution
   c1 = acos2( cos(a).*cos(b) + sin(a).*sin(b).*cos(C),  C );     A1 = acos2( (cos(a) - cos(b).*cos(c1))./(sin(b).*sin(c1)),  a );     B1 =
acos2( (cos(b) - cos(a).*cos(c1))./(sin(a).*sin(c1)),  b );
   % second solution     c2 = 2*pi - c1;     A2 = mod(A1 + pi, 2*pi);
   B2 = mod(B1 + pi, 2*pi);
 end  function [B1, C1, c1, B2, C2, c2] = ssa(a, b, A)
%SSA   gives both solutions to the side-side-angle problem, in radians.
%
%   SSA(a, b, A) will result in NaNs if the existence condition %   |sin b * sin A| <= | sin a |  is not met.
%
%

   % first solution
   B0 = asin(sin(b).*sin(A)./sin(a));     B0(imag(B0) ~= 0) = NaN;

   B1 = mod(B0, 2*pi);     C1 = mal(A, B1, a, b);     c1 = msl(a, b, A, B1);

   % second solution
   B2 = mod(pi - B1, 2*pi);     C2 = mal(A, B2, a, b);     c2 = msl(a, b, A, B2);
```

```
    % check constraints
    indices = ( abs(sin(b).*sin(A)) > abs(sin(a)) );     B1(indices) = NaN;    C1(indices) = NaN;     c1(indices) = NaN;    B2(indices) =
NaN;     C2(indices) = NaN;    c2(indices) = NaN;
 end

  function [A1, B1, C1, A2, B2, C2] = sss(a, b, c)
%SSS   gives both solutions to the side-side-side problem, in radians.
%
%   SSS(a, b, c) results in NaNs for those indices where the existence %   condition |pi - a| - |pi - b| <= |pi - c| <= |pi - a| + |pi -b| is
not
%   met.
%

    % first solution
    A1 = acos2( (cos(a) - cos(b).*cos(c))./(sin(b).*sin(c)), a);
    B1 = acos2( (cos(b) - cos(a).*cos(c))./(sin(a).*sin(c)), b);     C1 = acos2( (cos(c) - cos(a).*cos(b))./(sin(a).*sin(b)), c);

    % second solution
    A2 = 2*pi - A1;
    B2 = 2*pi - B1;
    C2 = 2*pi - C1;

    % check constraints     indices = ( ...
        (abs(pi-a) - abs(pi-b)) > abs(pi-c) | ...          abs(pi-c) > (abs(pi-a) + abs(pi-b)) );     A1(indices) = NaN;   B1(indices) =
NaN;   C1(indices) = NaN;     A2(indices) = NaN;   B2(indices) = NaN;   C2(indices) = NaN; end function [a1, c1, C1, a2, c2, C2] =
aas2(A, B, b)

%benimki    aas2
 a1=asin(sin(A)/sin(B)*sin(b)); a2=pi-a1; %1.çözüm

C1=2* atan(cos((a1-b)/2)/tan((A+B)/2)/cos((a1+b)/2));
C2=2* atan(cos((a2-b)/2)/tan((A+B)/2)/cos((a2+b)/2));
 c1=2* atan(tan((a1+b)/2)*cos((A+B)/2)/cos((A-B)/2)); c2=2* atan(tan((a2+b)/2)*cos((A+B)/2)/cos((A-B)/2));
  end end

%
% This function solves any type of spherical triangle
% A spherical triangle has 6 basic elements: three angles (A,B,C) and three sides (a,b,c). % Given any three of the 6 basic ele-
ments, the function finds the other three.
%
% Usage of function
% for example, Given 3 elements of a spherical triangle: a=2, b=3,
C=104.4944158
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver(A,B,C, a,b,c)
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver
(0,0,104.4944158,2,3,0)

% Attention, the unknown (A,B,c) 3 elements of the spherical triangle should be replaced with zero.

% A spherical triangle has 6 basic elements: three angles (A,B,C) and three sides (a,b,c). % Given any three of the 6 basic ele-
ments, the function finds the other three.
%
%
% INPUT:
% A,B,C,a,b,c   element of spherical triangle. Only 3 values should be entered.
% the others must be zero. % Inputs units must be [decimal de-
gree]



%OUTPUT:
%
%  A1, B1,C1,a1,b1,c1  are 1. solutions of the spherical triangle.
%  A2, B2,C2,a2,b2,c2  are 2. solutions of the spherical triangle.
% outputs units are degrees minutes seconds [dd mm ss]

% Usage of function
%[A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] = SphericTriangleSolver
```

```
(0,0,104.4944158,2,3,0)
% Attention, the unknown, desired 3 elements of the spherical triangle should be replaced with zero.

% While preparing this function, Rody P.S. Oldenhuis subfunctions are used.
function [A1, B1, A2, B2,C1,C2,a1,b1,a2,b2,c1,c2] =
SphericTriangleSolver(A,B,C,a,b,c)
 format
long
%
ro=180/
pi;
A=0;,B=0;,C=0;       a=2;,b=3;,c=4; %A=28.97193110;,B=46.5843689140;,C=104.4944158550;          a=0;,b=0;,c=0;
%A=0;,B=46.5843689140;,C=0;              a=2;,b=0;,c=4;
%A=0;,B=0;,C=104.49441585500;         a=0;,b=3;,c=4;
%A=0;,B=46.58436891400;,C=104.4944158550;   a=0;,b=0;,c=4;

%A=112.62944444;,B=65.30972222220;,C=0;       a=0;,b=70.874166666;,c=0;
%A=0;,B=0;,C=0;     a=73.701111110;,b=70.874166666;,c=7.0355555550;
%A=112.62944444;,B=65.30972222220;,C=0;     a=0;,b=70.874166666;,c=0;
%A=120.57333330;,B=0;,C=0;               a=105.4811111110;,b=75.7294444440;,c=0;
%A=112.62944444;,B=65.30972222220;,C=0;     a=0;,b=70.874166666;,c=0;
%A=60.46011667;,B=42.55760065;,C=80.6425750;             a=0;,b=0;,c=0;

%A=25;,B=0;,C=0;  a=0;,b=50;,c=70; %A=76.255;,B=115.795;,C=0;     a=0;,b=0;,c=81.4367777770;
%A=30;,B=0;,C=0;  a=70;,b=0;,c=80;

%a=148.573333;,b=142.1933333;c=0;   A=153.12666666;,B=0;,C=0;
%A=132.695;,B=107.1163889;,C=0;       a=146.345833333;,b=0;,c=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A1=0;A2=0;B1=0;B2=0 ;C1=0;C2=0;a1=0;a2=0;b1=0;b2=0 ;c1=0;c2=0;
 fprintf('Givens  [decimal degree]   \n') fprintf(' Angles            A=%3d %2.0f %6.4f B=%3d %2.0f %6.4f C=%3d %2.0f %
6.4f\n', degrees2dms(A), degrees2dms(B), degrees2dms(C))

fprintf('Sides     a=%3d  %2.0f  %6.4f  b=%3d  %2.0f  %6.4f  c=%3d  %2.0f  %6.4  f\n',  degrees2dms(a),  degrees2dms(b),  de-
grees2dms(c)) fprintf('Requested  [dd mm ss]   \n')
A=A/ro;,B=B/ro;,C=C/ro;  a=a/ro;,b=b/ro;,c=c/ro;


%1.option KKK
  if (A==0 & B==0 & C==0)
[A1,B1, C1, A2,B2,C2] = sss(a,b,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(A1), degrees2dms(B1), de-
grees2dms(C1)) %fprintf('A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(A2), de-
grees2dms(B2), degrees2dms(C2))

e
n
d
%2.option AAA
 if (a==0 & b==0 & c==0)
[a1, b1, c1, a2, b2, c2] = aaa(A,B,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('a1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(b1), de-
grees2dms(c1)) %fprintf('a2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(b2), degrees2dms(c2))

e
n
d

%3.option KAK  [c1, A1, B1, c2, A2, B2] = sas(a, C, b)
 if (c==0 & A==0 & B==0)
[c1, A1, B1, c2, A2, B2] = sas(a,C,b); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
```

```
fprintf('c1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(A1), de-
grees2dms(B1)) %fprintf('c2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(c2), de-
grees2dms(A2), degrees2dms(B2))
 end  if (a==0 & B==0 & C==0)
[a1, B1, C1, a2, B2, C2] = sas(b,A,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('a1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(B1), de-
grees2dms(C1)) %fprintf('a2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(B2), degrees2dms(C2)) end
%[c1, A1, B1, c2, A2, B2] = sas(a, C, b)
 if (b==0 & A==0 & C==0)
[b1,C1, A1, b2, C2, A2] = sas(c,B,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('b1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f\n', degrees2dms(b1), degrees2dms(C1), de-
grees2dms(A1)) %fprintf('b2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f\n', degrees2dms(b2), de-
grees2dms(C2), degrees2dms(A2))

e
n
d

%4.option AKA   [C1, a1, b1, C2, a2, b2] = asa(A, B, c)
 if (C==0 & a==0 & b==0)
[C1,a1, b1, C2 a2, b2] = asa(A,B,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('C1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(C1), degrees2dms(a1), de-
grees2dms(b1))
%fprintf('C2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(C2), degrees2dms(a2), de-
grees2dms(b2))
 end  if (A==0 & b==0 & c==0)
[A1, b1, c1, A2, b2, c2] = asa(B,C,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(A1), degrees2dms(b1), de-
grees2dms(c1)) %fprintf('A2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(A2), de-
grees2dms(b2), degrees2dms(c2))
 end  if (B==0 & a==0 & c==0)
[B1,  c1,a1, B2,c2, a2] = asa(C,A,b);[B1,  c1,a1, B2,c2, a2] =[B1,  c1, a1, B2,c2, a2] *ro; fprintf('a1=%3d %2.0f %6.4f c1=%3d
%2.0f %6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(c1), degrees2dms(B1)) %fprintf('a2=%3d %2.0f %6.4f
c2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(a2), degrees2dms(c2), degrees2dms(B2))

e
n
d

%5.option KKA
 if (c==0 & B==0 & C==0)
[c1, B1, C1, c2, B2, C2] = ssa(a,b,A); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('B1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(B1), de-
grees2dms(C1))
fprintf('B2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(B2), de-
grees2dms(C2))
 end  if (a==0 & A==0 & C==0)
[a1,A1, C1, a2, A2, C2] =
ssa(b,c,B);
A1=ro*A1;A2=ro*A2;B1=ro*B1;B
2=ro*B2
;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2
;c1=ro*c1;c2=ro*c2;
  fprintf('C1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(A1), de-
grees2dms(C1)) fprintf('C2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(A2), degrees2dms(C2))
 end  if (b==0 & A==0 & B==0)
[A1, B1, b1, A2, B2, b2] = ssa(c,a,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
```

```
 fprintf('b1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f\n', degrees2dms(b1), degrees2dms(A1), de-
grees2dms(B1)) fprintf('b2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f\n', degrees2dms(b2), de-
grees2dms(A2), degrees2dms(B2))

e
n
d

%6.option AAK
 if (C==0 & c==0 & b==0)
[b1, c1, C1, b2, c2, C2] = aas(A,B,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('c1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(b1), de-
grees2dms(C1))
fprintf('c2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(b2), de-
grees2dms(C2))
 end  if (A==0 & a==0 & c==0)
[A1, a1, c1, A2, a2, c2] = aas(B,C,b);
A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1; a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('A1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(A1), de-
grees2dms(a1)) fprintf('A2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f\n', degrees2dms(c2), de-
grees2dms(A2), degrees2dms(a2))
 end  if (B==0 & a==0 & b==0)
[a1, B1, b1, a2, B2, b2] = aas(C,A,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('a1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(b1), de-
grees2dms(B1)) fprintf('a2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(b2), degrees2dms(B2))

e
n
d

%%%%%==============================================
%7.option  AKK
 if (b==0 & B==0 & C==0)
[c1, B1, C1, c2, B2, C2] = ssa(a,c,A); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2; fprintf('C1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n',
degrees2dms(c1), degrees2dms(B1), degrees2dms(C1))
fprintf('C2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(B2), de-
grees2dms(C2)) end
if (c==0 & A==0 & C==0)
[c1, A1, C1, c2, A2, C2] = ssa(b,a,B); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;

fprintf('A1=%3d %2.0f %6.4f c1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(C1), de-
grees2dms(A1))
fprintf('A2=%3d %2.0f %6.4f c2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(c2), degrees2dms(C2), de-
grees2dms(A2)) end  if (a==0 & A==0 & B==0)
[a1, A1, B1, a2, A2, B2] = ssa(c,b,C); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('B1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(B1), de-
grees2dms(A1)) fprintf('B2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(B2), degrees2dms(A2)) end

% 8.option  KAA
 if (c==0 & B==0 & b==0)
[c1, b1, B1, c2, b2,B2] = aas2(A,C,a); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('c1=%3d %2.0f %6.4f B1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(c1), degrees2dms(B1), de-
grees2dms(b1)) fprintf('c2=%3d %2.0f %6.4f B2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(c2), de-
grees2dms(B2), degrees2dms(b2)) end  if (c==0 & a==0 & C==0)
[c1, a1, C1, c2, a2,C2] = aas2(A,B,b); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
 fprintf('c1=%3d %2.0f %6.4f a1=%3d %2.0f %6.4f C1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(c1), de-
grees2dms(C1)) fprintf('c2=%3d %2.0f %6.4f a2=%3d %2.0f %6.4f C2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(c2), degrees2dms(C2)) end  if (b==0 & a==0 & A==0)
[b1, a1, A1, b2, a2,A2] = aas2(C,B,c); A1=ro*A1;A2=ro*A2;B1=ro*B1;B2=ro*B2 ;C1=ro*C1;C2=ro*C2;a1=ro*a1;
a2=ro*a2;b1=ro*b1;b2=ro*b2 ;c1=ro*c1;c2=ro*c2;
```

```
 fprintf('a1=%3d %2.0f %6.4f A1=%3d %2.0f %6.4f b1=%3d %2.0f %6.4f\n', degrees2dms(a1), degrees2dms(A1), de-
grees2dms(b1)) fprintf('a2=%3d %2.0f %6.4f A2=%3d %2.0f %6.4f b2=%3d %2.0f %6.4f\n', degrees2dms(a2), de-
grees2dms(A2), degrees2dms(b2)) end

%a1, B1, c1degrees2dms .
```