Hemispherical Iteration Tracking Method

Improved Data Reduction for Two-Station Optical Tracking of Model Rockets

National Association of Rocketry
Research & Development Report

by

John S. DeMar
NAR 52094

NARAM-38

August 1996
Evansville, IN

TABLE OF CONTENTS

1.0 INTRODUCTION

Historically, the choice of formulae for model rocketry altitude data reduction has been limited to linear closed-form equations. Using simple trigonometric functions, these equations require only a pocket calculator or, if necessary, could be derived by hand using sine and cosine tables. The two commonly used methods for two-station theodolite data reduction are Geodesic and Vertical Midpoint. In recent years the widespread availability of programmable calculators and portable computers has automated these calculations. In fact, the typical portable computer has more than enough computing power to perform vastly more complex methods of data reduction. Until recently, it would be impractical to consider other logical methods which would apply successive approximation, iterative methods, or other techniques requiring special rules.

The main objective of this project was to implement a computer software program for a more accurate method of determining a model rocket's likely altitude given two-station theodolite angles. The method should consider the real-world characteristics of optically tracking a model rocket to ejection (the virtually universal choice for marking its position). The algorithm may be as complex as necessary but should run reasonably quickly on the average laptop computer. The implementation should be capable of executing on a variety of low-cost computer platforms with a universal text-based interface, and made readily available to everyone in the rocketry community.

The algorithm designed in this project (called "Hemispherical Iteration Tracking") uses an iterative computational method to locate a three-dimensional region representing the most likely location of the target. The lower boundary of this region is a plane containing the minimal horizontal intersection, and the upper boundary is estimated as the top half of an elliptical sphere. This hemisphere represents the "region of uncertainty" for the target's likely location. Further checks are done to skew this region depending on the target's location relative to the trackers and baseline. To compute the mean altitude, the skewed hemisphere is treated as a mass and the most likely position is computed as the center of mass. To compute the closure error, the height of the hemisphere is compared to the mean altitude as a percentage.

The software was tested with a variety of sweeping sets of angles generated to represent various regions in 3D space relative to the trackers. The divergence of the HIT method was compared to the Geodesic method with favorable results. Additionally, a set of data from a particularly problematic NAR-sanctioned regional was used to compare the new method with the Geodesic equations; the results showed that the closure rate would have improved from only 31% of tracked flights closed to over 88% closed.

NAR competitors would benefit from this new data reduction method by improving the closure rate for altitude events, offering a more even playing field when existing algorithms fail to close reliably under certain conditions. Further statistical analysis would be needed before approving the new method for NAR-sanctioned competition and record attempts.

2.0 ANALYSIS

2.1 Current Methods

Before investigating new approaches to tracking data reduction, the current methods were analyzed. The main concerns were the shape of the uncertainty region, the susceptibility to divergence in certain regions, and real-word applicability to model rocket tracking.

2.1.1 Vertical Midpoint

The Vertical Midpoint method connects a vertical line between the tracking vectors, and locates the average altitude halfway between the lines. The closure error is computed as half the length of this vertical line.

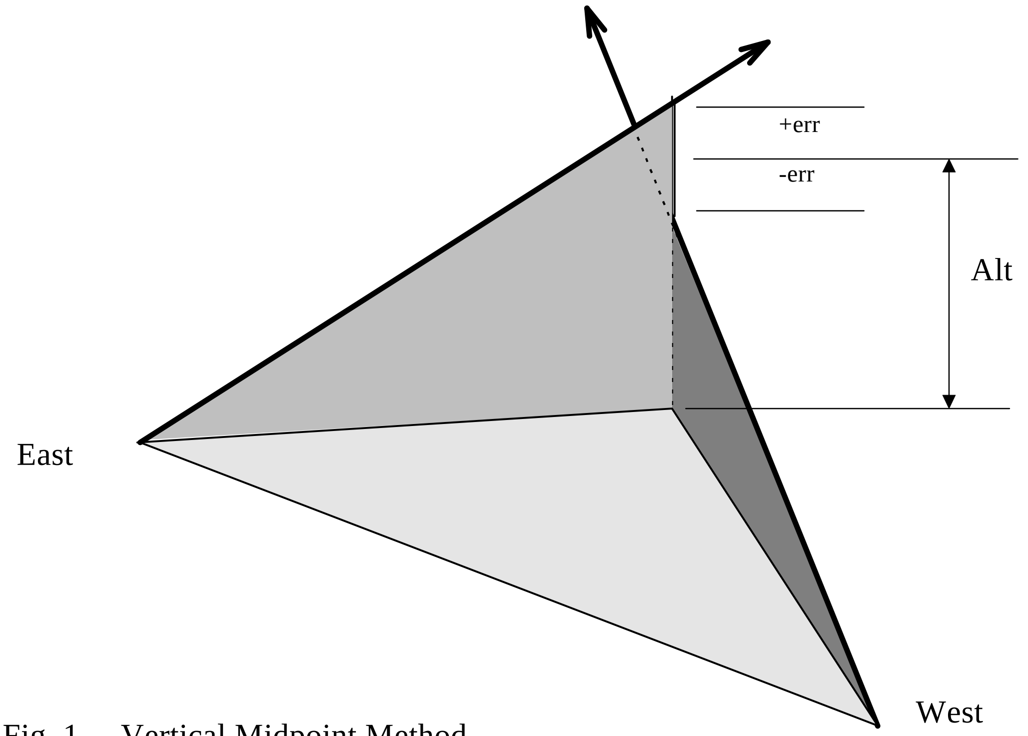

Fig. 1 Vertical Midpoint Method

The uncertainty region is described by a simple vertical line segment in this method. Consequently, errors are concentrated in the vertical direction without concern for sideways movement.

With moderately low elevation angles and azimuth angles significantly away from the baseline, Vertical Midpoint will converge with a similar result as Geodesic (the vertical line is a close approximation to the diameter of the Geodesic sphere). However, when the target is close to baseline, or the elevation angles are large, the vertical connecting line becomes excessively high and the closure percentage diverges.

Obviously, this method limits the useful region for successful tracking, and should not be used as a general method for contest events. Vertical Midpoint is useful only when the simplified calculation must be done relatively quickly by hand.

2.1.2 Geodesic

The Geodesic method describes a shortest line segment between the two tracking vectors. The average position is located at the midpoint of this line and the closure error is proportional to the length of the line. This method differs from Vertical midpoint by not limiting the calculation to a vertical plane; the connecting line will be perpendicular to each vector and may point anywhere in 3D space.

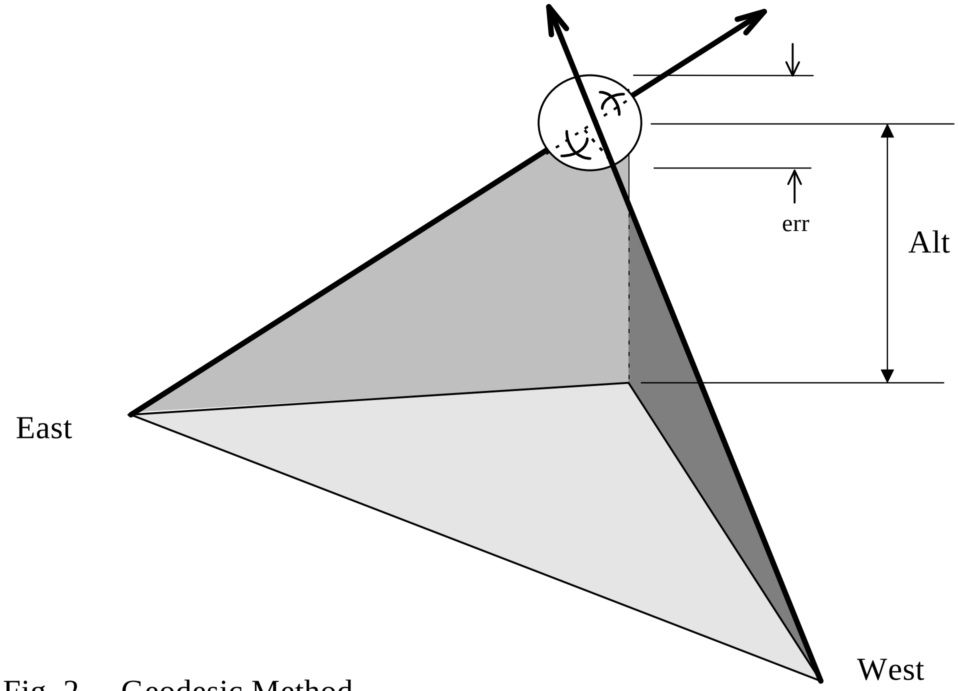

Fig. 2 Geodesic Method

The Geodesic "region of uncertainty" may be visualized as a "ball" being held between two "sticks", where the size of the ball is the just large enough to touch the sticks at only one tangent point each. In this case, errors are assumed to be equal for both trackers and in the direction perpendicular to each other.

When the target is sufficiently away from the baseline and gives relatively low elevation angles, the connecting segment will be mostly vertical, concentrating on errors in the vertical axis. When elevation angles are high or the target is near the baseline, the connecting segment will be mostly horizontal, concentration errors in horizontal rotation. As will be discussed later, this inconsistent treatment of tracking error is not optimized for typical model rocket tracking. However, it is a vast improvement over the more simplistic Vertical Midpoint method.

2.2 Practical Error Modes

A thoughtful attempt was made to visualize and logically consider the error modes of tracking a model rocket to the point of ejection. The author has had significant practical experience manning a tracking station at several Regional and Open meets, and was responsible for data reduction at NARAM-37.

The goal of this analysis was to derive a set of practical rules which would lead to the implementation of an improved method.

2.2.1 Mechanical Resolution

The simplest error mode is the mechanical limitation of the tracker's pointing devices. Most well-built stations will indicate half-degree increments, but many tracking operators report the angles in whole degrees. A +/- 0.5 degree error could cause a track to close (or not close) in some circumstances.

This error mode was analyzed in Bobby Gormley's NARAM-37 R&D report. He derived a method for automatically varying the reported angles within half a degree to locate a better closure percentage using the Geodesic equations. The results showed how any standard method could be improved by incorporating an iterative method to compensate for mechanical error and/or rounding error. Even with a +/- 0.25 degree allowance (which would be statistically more reasonable than 0.5 degree), more flights would track within the 10% maximum tracking error. Also, an improved implementation of his software would compute more sets to locate the best closure.

The "region of uncertainty" described by Gormley's method may be visualized as tall, converging, 4-sided pyramids with their apexes at the tracking stations. The most likely location will be along the line described by the Geodesic method, but may not necessarily intersect. However, increasing the error angles further would create a region intersecting at the lower extent of the Geodesic line with more "weight" above this plane (parallel to the ground).

2.2.2 Visual Resolution

Extending the "Gormley paradigm", the tracking vectors could be swept in full rotation around the nominal reported angles. The range of rotation could be chosen to match the error imposed by the tracking station's pointing scope (and the human factor of targeting through the crosshairs). From the experience of the author, this aiming error is of the magnitude of +/-2 degrees; outside of that range, the operator is making a "best guess" (sometime referred to as a "wild guess"!). The vectors now become cones, increasing upward, with their apexes at the tracking stations. The intersection of these cones is a complex shape, similar to a concave pair of elliptical solids. This region could be estimated as a hemisphere with it's base facing downward for most practical target locations.

2.2.3 Relative Acquisition Time

One tracking operator may take more time to lock on to the ejected tracking powder cloud. The significance of this error depends on the relative movement of the tracking cloud due to wind and gravity. In general, one tracker will be pointing down wind and below the other one. To allow for this, the lower tracking vector would be rotated toward the other vector and raised slightly.

More complex movement would have to be taken into consideration if the rocket was tracked to apogee. Since this project is concerned only with tracking to ejection, the relative movement of the rocket will be ignored.

To simplify the problem further, the ejection cloud drift would be handled separately for wind and gravity. In this case, each tracking vector would be moved independently toward the other in all four orthogonal directions, inscribing a complex triangular wedge shape. For most regions of 3D space, this shape could be smoothed into a hemispherical shape, resembling the upper half of an elliptical paraboloid. For extremely large elevation angles, the shape diverges to a tall trapezoid and the hemispherical approximation does not hold true. However, most tracking methods will have increased error at high angles (including the human limitations of bending backwards and seeing the target).

2.2.4 Tracking Plane

When the tracking stations are not properly zeroed, an angular offset may be present in one or both dimensions. This "tilted" base plane will cause closed-form tracking reduction methods to not close for most flights. Allowing for this problem "after the fact" would require significant computation analysis of the data set to determine the error correction. However, an iterative method which allows for other typical error would likely close more tracks, especially if the zeroing error was limited to one dimension.

2.3 Significant Factors

The mechanical error and visual error have similar modes, especially when considered in the vertical direction (where only altitude is concerned, and not absolute position in three dimensions). The visual error, however, is close to an order of magnitude more significant than the mechanical error. Therefore, a method which incorporates the visual error would cover both.

The acquisition timing problem concentrates the error along a base vector between the tracking vectors. This connecting plane would contain the minimal horizontal line (as opposed to the minimal vertical line of Vertical Midpoint or the minimal perpendicular line of Geodesic). Further allowance for gravitation drift would skew the likely position upward from the lower tracking vector.

The tracking plane (zeroing) error shouldn't be significant in an accurately built theodolite pair which has been setup properly. However, allowing for this error mode would require special case software to analyze the tracking data. This is beyond the scope of this report.

A common characteristic of all factors is a concentration of "uncertainty shapes" beginning at the lower extreme of the minimal horizontal connecting line and decreasing in certainty "up and away" from the trackers. A logical analysis of these shapes concluded with an initial iterative algorithm designed around estimating a "conglomerate" shape. For further accuracy, the angles could be tested for extreme cases with allowances made for those special error regions.

3.0 IMPLEMENTATION

3.1 Visualization

Before discussing the software implementation, it may be worthwhile to look at a graphical representation of the error region. The following figure shows an upper hemisphere as an approximation of the "uncertainty region" for the new algorithm. The error is averaged by computing the center of mass of this sphere (NOT by simply taking half of it's height and adding to it's base). By using this "weighted" method, the altitude will represent a more accurate statistical mean. The closure error uses the full range as represented by the height of the sphere.

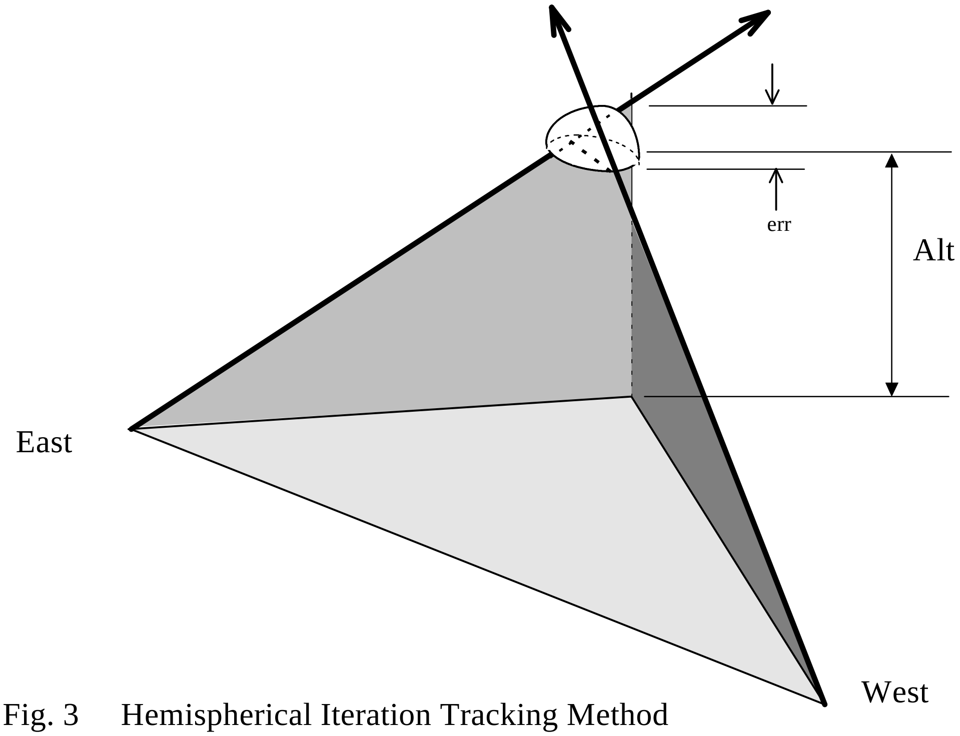

Fig. 3 Hemispherical Iteration Tracking Method

3.2 Flow Description

The following flow diagram describes the software implementation of the Hemispherical Iteration Tracking method.

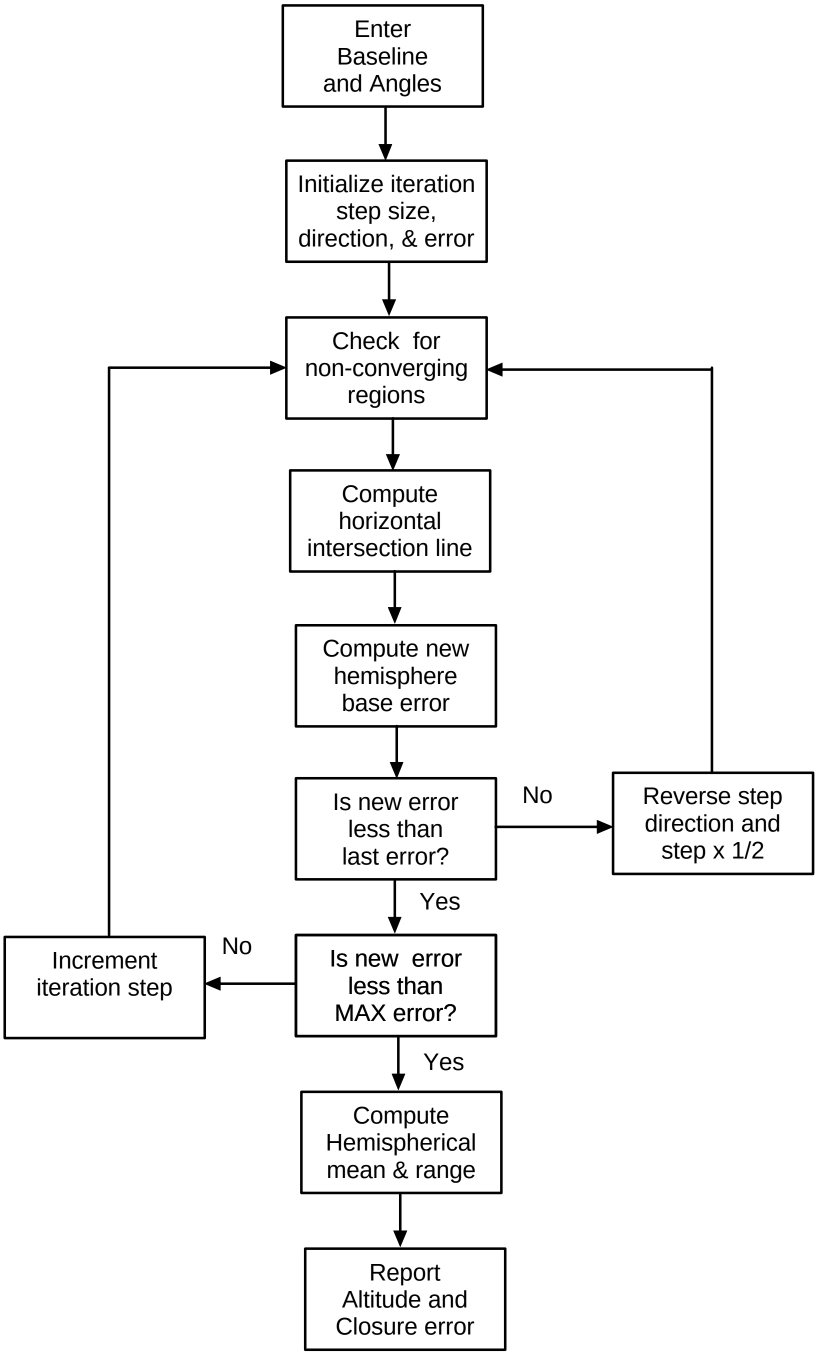

Fig. 4 Hemispherical Iteration Tracking Method Software Algorithm

3.3 Software Operation

The software implementation for the HIT method is written in the ANSI-standard version of the 'C' programming language. The source code is portable to any system which has a standard C-compiler available. The program has been tested using Microsoft C on the PC/DOS platform, and on both the SunOS and System V Unix environments. The program has not been tested on an Apple Macintosh but it should be compatible if compiled with a standard C-compiler.

The program runs from a command-line prompt under DOS, as shown in the following examples (using the "Pink Book" test cases):

```
---------------------------------------------
Two-station Altitude Data Reduction
HIT (Hemispherical Iteration Tracking Method)
(c)1995,1996 John S. DeMar, NAR #52094

Usage: hit [baseline] [AZ1] [EL1] [AZ2] [EL2]
        -v for less verbose report
        +d for debug output
---------------------------------------------
   BASELINE = 300
  AZIMUTH 1 = 90
ELEVATION 1 = 45
  AZIMUTH 2 = 50
ELEVATION 2 = 40
        HIT = 392.4, 4.3%
   Geodesic = 380.0, 6.3%
---------------------------------------------
   BASELINE = 300
  AZIMUTH 1 = 30
ELEVATION 1 = 45
  AZIMUTH 2 = 60
ELEVATION 2 = 45
        HIT = 218.1, 17.8%
   Geodesic = 203.3, 31.2%
---------------------------------------------
   BASELINE = 300
  AZIMUTH 1 = 120
ELEVATION 1 = 75
  AZIMUTH 2 = 25
ELEVATION 2 = 55
        HIT = 596.1, 3.2%
   Geodesic = 596.3, 6.2%
---------------------------------------------
   BASELINE = 300
  AZIMUTH 1 = 30
ELEVATION 1 = 80
  AZIMUTH 2 = 40
ELEVATION 2 = 85
        HIT = 1344.4, 1.6%
   Geodesic = 1337.8, 3.2%
---------------------------------------------
```

Running `hit.exe` with no comamnd-line arguments will display this message about the required input parameters.

Example #1 from the Pink Book.

Altitude is 3.3% higher.
Closure is 2% better.

Example #2 from the Pink Book.

Altitude is 7.3% higher.
Closure is 13.4% better.
Still a NO CLOSE as expected with a gross targeting error.

Example #3 from the Pink Book.

Altitude agrees within 0.1%.
Closure is 3% better.

Example #4 from the Pink Book.

Altitude is 0.5% higher.
Closure is 1.6% better.

4.0 DATA COMPARISON

4.1 Standard Examples

As shown in the Pink Book examples above, the standard test cases give acceptable results using Hemispherical Iteration Tracking. As compared to the Geodesic method, the cases show a significant improvement in closure percentage.

For the medium elevation angles in the first two examples, HIT gives slightly higher altitudes. This is a result of weighting the error at the base of the hemisphere as compared to the center of the somewhat smaller Geodesic sphere.

For higher elevation, as in the last two examples, HIT agrees within 0.5% of the Geodesic altitudes. In these cases, the hemisphere and the Geodesic sphere have very close centroid locations.

4.2 Convergence Tests

To do an extensive test of the new method, eleven data sets were run using twenty increments of one-degree each. The test sets represent unique regions of space relative to the tracking baseline, and the angles are swept to converge or diverge around the target. Each test is limited to one rotational dimension and direction.

The purpose of these tests is to determine if there are any discontinuities or irregularities in the algorithm. The Geodesic results are computed at the same time for comparison.

The results of these tests do not show any problem regions.

See Appendix B for output data from the divergence tests. This information is also included on the enclosed diskette to allow further graphical and statistical analysis.

4.3 Practical Data Set

To further substantiate the usefulness of the method, a complete set of altitude data was recalculated for a NAR Regional Meet (Sanction #1019-96R, ASTRE NYSPACE, May 25, 1996). This set was chosen because of an abnormally high number of non-closed tracks. Tracking operators and the equipment manager could not account for the problem, even though the stations were checked for correct zeroing several times. Later analysis showed a possible 2-degree tilt in one azimuth protractor. However, other factors made it difficult to lock onto the targets, including: very windy conditions, variable skies, and many flights over the west tracker. The events were C Payload and A altitude using a 515m baseline.

NYSPACE '96 Tracking Analysis
May 25-26, 1996
Johnstown, NY
Sanction #1019-96R

Comparison of Geodesic and HIT Methods of tracking data reduction.
John DeMar
June 2, 1996

Baseline = 515m

NAR#	evnt	mtr	Ea	Ee	Wa	We		Geodesic		HIT Method	
52094	C-PL	c6-7	13	55.5	37.5	63		467.6	15.3%	482.2	7.7%
46148	C-PL	c6-5	7	44	50	71.5		405.2	12.9%	419	6.4%
46148	C-PL	c6-5	11	37	73.5	68.5		353.8	12%	368.9	6.1%
46148	C-PL	c6-7	12.5	45	63.5	69		436.7	12.4%	453	6.3%
11077	C-PL	c6-5	12.5	30.3	86	61.5		285.6	15.9%	307.1	8.4%
11077	C-PL	c6-5	12	37	66	64		330.5	16.2%	349.6	8.3%
24516	C-PL	c6-7	10	47	64	73		478.2	11.1%	493.1	5.6%
24516	C-PL	c6-7	-3	39	120	73	?	407.9	35.3%	461.7	16.3%
27910	C-PL	c6-7	21	41	81	61		426.4	12.3%	447.9	6.7%
tm049	C-PL	c4-7	16.5	41.5	81.5	66		433.7	11.3%	452.3	5.9%
tm049	C-PL	c4-7	8	33	138	70		407.9	2.9%	412.4	1.5%
tm049	C-PL	c10-7	14	28	56	53.5		221.9	14.2%	235.1	7.7%
tm049	C-PL	c4-7	14.5	41	79	70		428.9	6.6%	438.9	3.4%
tm136	C-PL	c6-7	12.7	41	55.5	69		380.7	6.1%	387.7	3.1%
tm136	C-PL	c6-5	14	15	85	45		134.2	6.8%	141	4.6%
tm160	C-PL	c10-7	14	55	63	74		624.6	8.5%	638.3	4.3%
tm160	C-PL	c6-5	8	27	85	70		255.3	8.6%	264.2	4.4%
60060	A-alt	a3-4t	12	30	43.5	55		229	11.3%	238	6%
19348	A-alt	a8-3	11	16	43	37	?	110.2	19.4%	121.6	11.5%
24516	A-alt	a3-4t	15	36	34	53.5		263.5	5.4%	267.4	2.9%
11077	A-alt	a3-6t	13.5	29	55	54.5		229.6	14.8%	243.3	8%
11077	A-alt	a3-4t	16	26.5	44.5	44.5		188.5	13.7%	198.6	7.8%
11077	A-alt	a3-4t	15	24	45	45		174.6	10.7%	182.5	6.2%
27910	A-alt	a3-6t	9	42	28.5	55		295.9	15.6%	306.3	7.8%
27910	A-alt	a2-7	3	43	81.5	80.5		460.9	11%	474.1	5.4%
46148	A-alt	a3-6	5	32	41.5	50.5	?	214.7	39.9%	241.1	19.3%
46148	A-alt	a3-6	13.1	46	36.5	61		366.8	10.7%	376.2	5.5%
64217	A-alt	a3-6	7.5	26.5	21.5	46		176.7	8.8%	181	4.5%
64217	A-alt	a3-4t	15.2	25	40.5	36	?	156.6	27.5%	172.7	15.8%
60054	A-alt	a3-4t	8	30	43	60		234.6	14.7%	246	7.5%
tm136	A-alt	a3-6t	12.5	37	38	59.5		289.9	7.2%	295.9	3.7%
tm160	A-alt	a2-7	13.5	45	44.5	64.5		386.8	9.8%	396.8	5%
tm049	A-alt	a3-6t	12.5	34.5	42.5	57		266	11.7%	275.7	6.1%
tm049	A-alt	a3-6t	9	35	26	53.5		245.8	9.8%	251.8	5%
tm049	A-alt	a3-6t	8	36	35	62		282.2	11.1%	290.7	5.6%

SUMMARY:

	GEO		HIT	
Closed:	11/35	31.4%	31/40	88.6%
Not Closed:	24/40		4/40	
Track Lost:	5/40		5/40	

The closure rate improved from 31.4% for the Geodesic Method to 88.6% for the H.I.T. method. This is a significant improvement, and would have had considerable effect on the outcome of the events.

It is important to note that the four flights with obviously large tracking errors remain non-closed using both methods.

5.0 CONCLUSIONS

The tracking data reduction method derived in this projects was based on a practical analysis of the various error modes, and was not limited to a static mathematical treatment of the problem. Using the computing power of an average low-cost portable PC, the software implements a complex rules set that would have been impractical to derive (and hand compute) using standard closed-form trigonometric methods.

The mathematical basis of the new method has a strong similarity to the closed form Geodesic method, but adds weighting for real-world effects of tracking model rockets to ejection. The grouping of errors into a "region of uncertainty" is approximated by a hemispherical shape which is located using an iterative technique.

The resulting "Hemispherical Iteration Tracking" method was shown to improve the rate of closure compared to Geodesic, and was tested without exhibiting any nonlinearities or discontinuities. Further statistical analysis and peer review would be needed in order to approve the new method for NAR-sanctioned contest use.

The software implementation proved to run adequately on any low-cost portable computer, and could be used by any NAR sections with access to any personal computer.

Improvements could be made using the existing software as a framework. Other special cases could be added if warranted by further analysis. Also, the software could be used as a building block for post-flight analysis of tracking data.

BIBLIOGRAPHY

Clarke, Bill, *Aviators Guide to GPS*, 1994, TAB Books.
(Error grouping concepts.)

Gormley, Bobby, "Error Compensatlon in Altitude Data Reduction", NAR R&D Report, 1995, NARTS.
(Iterative technique to compensate for 0.5 degree tracker rounding.)

Leithold, Louis, *The Calculus with Analytical Geometry*, 1976, Harper & Row.
(Refreshing my rusty brain to solve center of mass of a solid hemisphere using a triple integral).

APPENDIX A: SOFTWARE SOURCE CODE

```
/*------------------------------------------------------------------------

  HIT.C

  Improved Altitude Data Reduction for two tracking stations.

  --------------------------------------------
     Hemispherical Iteration Tracking Method
  --------------------------------------------

  Algorithm and Implementation (c)1995,1996 John S. DeMar.
  All rights reserved.

  John DeMar
  smdemar@mailbox.syr.edu
  4753 Freestone Road
  Liverpool, NY 13090
  NAR 52094

  Revised for submission as a National Association of Rocketry R&D Report.
  August 1996.
  This software may be used for non-commercial purposed by members of the
  National Association of Rocketry. No commercial rights are granted
  without express written consent of the author.

--------------------------------------------------------------------------
  ALGORITHM

  Given:  Two theodolite "tracking stations", zeroed facing each other.
          "Baseline"   distance between two stations.
          "Elevation and Azimuth" angles from each station.

  Assume: Level field (no elevation difference between stations).

 Compute:
  Begin at 'z' = 0, and step 'z' until a minimal distance is found in the
'x,y' plane between the two track vectors. The iteration method decreases
the step size and reverses the step direction when a minima is passed. The
process ends when the either the step size is less then the value MINDIFF
(defaults to 0.01), or two iteration find the same result. If more than
MAXITER loops occur before the criteria is met, the process terminates.

 Report:
  Once the minima is found, a sphere is 'constructed' with a diameter
equal to the minimum x/y-plane distance between the tracking vectors. The
target is assumed to be somewhere in the upper hemisphere of the sphere.
This is based on the nature of a typical track and aiming error. The
z-coordinate at the "center of mass" of the sphere is reported as the mean
altitude. The height of the hemisphere is the probable range of error,
and is used to compute a closure percentage of the mean altitude.

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

  PROGRAM USAGE

  hit |-v| |+d| [baseline] [az1] [el1] [az2] [el2]

   -v  Turns off verbose output.
       Only the two values are reported (altitude and closure%) on one line.
            Intended for batch processing of multiple data sets.

   +d  Turns on debug output.

   baseline      The distance between the two tracking stations.
   az1         The azimuth (sideways angular movement) of station 1.
   el1         The elevation (upward angular movement) of station 1.
   az2         The azimuth (sideways angular movement) of station 2.
   el2         The elevation (upward angular movement) of station 2.

--------------------------------------------------------------------------
 FUTURE STUFF:

  - Add initial elevation offset between tracking stations.

------------------------------------------------------------------------*/

#include <stdlib.h>
```

```
#include <stdio.h>
#include <math.h>


/*-----------------------------------
   use for manual error checking
-----------------------------------*/
#define A1_ADJ   0
#define E1_ADJ   0
#define A2_ADJ   0
#define E2_ADJ   0


/*-----------------------------------
   use for normal end-user compile
-----------------------------------*/
#define DEBUG 0

/*---------------------------------
   use for development debugging
---------------------------------*/
/* #define DEBUG 1 */

/*------------------
   some constants
------------------*/
#define PI 3.1415926
#define TENTHDEG (0.031415926/180)

/*---------------------------------------
   minimal error to end iterative loop
---------------------------------------*/
#define MINDIFF 0.01
/*--------------------------------
   maximum number of iterations
--------------------------------*/
#define MAXITER 100


/*-----------------------------
   prototypes in this module
-----------------------------*/
void geodesic( int pr );


/*----------------------------
  Globals for inputs & modes
----------------------------*/

double bl, a1, e1, a2, e2;   /* baseline, azimuth and elevation tracker 1 & 2 */
double ga, gc;   /* result altitude and closure error */
int quiet = 0;   /* verbose output == 1 */


/*-------------------------------------------------------------------------
   One big function to do the whole thing...   (should be re-structured).
-------------------------------------------------------------------------*/

int main( int argn, char *argv[] )
{
   int debug = DEBUG;
   int i = 0, j = 0;
   double ins[10];
   double x1,y1,x2,y2,c1,c2,dx,dy,dp,dc;
   double step, stop, z;
   double err;
   int stepnum=0;
   int dir = 1;
   int argx = argn;
   double g1,g2,g3,g4,ex,ax,gmin,gmax;

   /* extract parameters from command line */
   if( argn > 10 )
   {
      argn = 10;
   }

   while( argn-- > 0 )
   {
```

```
      if( !strcmp( argv[i], "-v" ) )
      {
         quiet = 1;
      }
      else if( !strcmp( argv[i], "+d" ) )
      {
         debug = 1;
      }
      else
      {
         if( debug ) {
         printf( "%d: %s\r\n", i, argv[i] );
         }
         ins[j] = atof( argv[i] );
         if( debug ) {
         printf( "%d: %g\r\n", j, ins[j] );
         }

         j++;
      }
      i++;
   }

   if( !quiet )
   {
      printf( "Two-station Altitude Data Reduction\r\n" );
      printf( "HIT (Hemispherical Iteration Tracking Method)\r\n" );
      printf( "(c)1995,1996 John S. DeMar, NAR #52094\r\n" );
      printf( "\r\n" );
   }

   /* warn about insufficient parameters */
   if( argx < 6 )
   {
       printf( "Usage: hit [baseline] [AZ1] [EL1] [AZ2] [EL2]\r\r\n", argn );
      printf( "        -v for less verbose report\r\n" );
      printf( "        +d for debug output\r\n" );
      return( 1 );
   }

   if( !quiet )
   {
      printf( "   BASELINE = %g\r\n", ins[1] );
      printf( "  AZIMUTH 1 = %g\r\n", ins[2] );
      printf( "ELEVATION 1 = %g\r\n", ins[3] );
      printf( "  AZIMUTH 2 = %g\r\n", ins[4] );
      printf( "ELEVATION 2 = %g\r\n", ins[5] );
   }

   /* extract parameters into baseline and degrees in radians */
   bl = ins[1];
   a1 = (ins[2]+A1_ADJ) * PI / 180;
   e1 = (ins[3]+E1_ADJ) * PI / 180;
   a2 = (ins[4]+A2_ADJ) * PI / 180;
   e2 = (ins[5]+E2_ADJ) * PI / 180;

   /*********************/
   /* begin algorithm... */
   /*********************/

   step = bl/2;      /* start at a step of 1/2 the baseline */
   stop = MINDIFF;   /* stop when iterative error is less than this */
   z = bl/2;         /* first guess is half the baseline */
   dp = 0;           /* start with a horizontal distance obviously too small! */

   /* loop until the "step" gives subsequent error sample less than "stop" */
   while( step > stop )
   {
      if( debug ) {
      printf( "%d: Step %g, Z %g, DIR %d\r\n", stepnum, step, z, dir );
      }

      /* compute coordinates of vectors at current 'z' guess... */

      /* check special cases for discontinuity */
      if( e1 == 90 )
      {
         /* right over tracker 1 */
         c1 = x1 = y1 = 0;
```

```
      }
      else
      {
         c1 = z / tan(e1);
         x1 = c1 * cos(a1);
         y1 = c1 * sin(a1);
      }

      if( e2 == 90 )
      {
         /* right over tracker 2 */
         c2 = y2 = 0;
         x2 = bl;
      }
      else
      {
         c2 = z / tan(e2);
         x2 = bl - (c2 * cos(a2));
         y2 = c2 * sin(a2);
      }

      if( debug ) {
      printf( "x1=%g, y1=%g, x2=%g, y2=%g\r\n", x1,y1,x2,y2 );
      }

      /* compute distance between coordinates at current 'z' guess... */

      dc = sqrt( ((x2-x1)*(x2-x1)) + ((y2-y1)*(y2-y1)) );

      if( debug )
      {
         printf( "dc %g, dp %g\r\n", dc,dp );
      }

      /* if the new horizontal distance is greater than old, */
      /* then do half steps, and reverse direction */
      if( dc > dp )
      {
         step /= 2;
         dir ^= 1;
      }
      else if( (dc == dp) || (dc == 0) )
      {
         /* if error isn't getting smaller, we're done */

         if( stepnum > 0 ) /* but sure make we've done at least one step */
         {
            break;
         }
      }

      /* otherwise, keep stepping */
      if( dir )
      {
         z += step;
      }
      else
      {
         z -= step;
      }

      /* save previous distance */
      dp = dc;

      /* check for non-convergence limit */
      stepnum++;
      if( stepnum > MAXITER )
      {
         break;
      }

   }


   /* What has been found at this point:
      1) dc: The minimum x,y plane distance between the two vectors.
      2)  z: The altitude at this plane from the baseline plane.

      Assumptions in this version of the method:
```

```
   1) The shape of the region of uncertainty is generalized by a
      hemisphere above the plane containing the minimal x,y distance.
       a) A rotation of an angular error is a cone diverging "up and away"
         from the tracker's point of view. The two cones will overlap
         in a region above the base of our approximating hemisphere.
       b) This overlapping 3D "blob" will be weighted toward each vector.
   2) The true "blob" representing the "uncertainty" will have a complex
      shape that will be difficult to determine in a generalized
      closed form.
       a) From intuitive visualization it seems very likely that
         the "darkest" area of the shape would be contained in our
         estimate of a hemisphere.
       b) We can ignore a "blob" that is elongated in the x-y plane
         because we are only interested in the z-axis (altitude).
       c) This would compress the blob with more weight toward the
         center axis of the hemispherical estimate.
     3) Targets below this region would represent an unlikely gross
       error on the part of both trackers. Other human factors would need
      to be studied to further describe the error modes.

*/


/* Mean altitude is last 'z' point (on minimal x,y line)
   plus the sphere's weighted "uncertainty" distribution center. */
/* Approximate a general case for the center of the uncertainty
   by using center of mass of the hemisphere as the most likely place. */
z = z + ( dc / (2 * PI ));

/* Approximate the error range as the radius (height) of the minimal sphere. */
/* compute as percent of altitude */
dc = ((dc/2)/z)*100;

if( !quiet )
   printf( "        HIT = " );
printf( "%g, ", floor((z+0.05)*10)/10 );
printf( "%g%%", floor((dc+0.05)*10)/10 );
if( quiet )
   printf( "," );

if( debug ) {
   printf( "\r\n" );
   printf( " Iterations = " );
   printf( "%d\r\n", stepnum );
}

/* show geodesic results for comparison... */
geodesic(1);

if( debug )
{
   printf( "\r\n" );
}


/*------------------------------------------------------------------
   The following is an attempt to describe the "uncertainty blob"
with another method. Each of the vectors are swung in each plane
separately toward the other vector. The resulting inscribed shape
is visualized as a distorted wedge. The final step is a computing of
the center of mass of this wedge (with a lot of assumptions!).
This seems to be a wild goose chase, but I left in here anyhow.
------------------------------------------------------------------*/

/* try reducing e1 to see if converging */
  dp = dc = gc;
ex = e1;
e1 -= TENTHDEG;
geodesic(0);
/* if better, this is the upper vector */
if( gc < dp )
{
   if(debug)
      printf( "v1 is higher\r\n" );
   while( gc < dp )
   {
        dp = gc;
      e1 -= TENTHDEG;
      g1 = ga;
```

```
            geodesic(0);
         }
         /* now move azimuth to intersect other vector */
         e1 = ex;
         geodesic(0);
         dp = dc = gc;
         ax = a1;
         /* move positive... */
         while( gc <= dp )
         {
            dp = gc;
            a1 += TENTHDEG;
            g2 = ga;
            geodesic(0);
         }
         /* move negative in case wrong direction */
         dp = dc = gc;
         while( gc <= dp )
         {
            dp = gc;
            a1 -= TENTHDEG;
            g2 = ga;
            geodesic(0);
         }
         a1 = ax;

         /* now move other azimuth to intersect this vector */
         geodesic(0);
         dp = dc = gc;
         ax = a2;
         /* move negative... */
         while( gc <= dp )
         {
            dp = gc;
            a2 -= TENTHDEG;
            g3 = ga;
            geodesic(0);
         }
         a2 = ax;

         /* now move other elevation to intersect this vector */
         geodesic(0);
         dp = dc = gc;
         ex = e2;
         /* move positive... */
         while( gc <= dp )
         {
            dp = gc;
            e2 += TENTHDEG;
            g4 = ga;
            geodesic(0);
         }
         e2 = ex;
      }
      else
      {
         /* try other vector */
           dp = dc;
         e1 = ex;
         ex = e2;
         e1 += TENTHDEG;
         e2 -= TENTHDEG;
         geodesic(0);
         /* if better, this is the upper vector */
         if( gc < dp )
         {
            if(debug)
               printf( "v2 is higher\r\n" );
            while( gc < dp )
            {
                 dp = gc;
               e2 -= TENTHDEG;
               g1 = ga;
               geodesic(0);
            }

            /* now move azimuth to intersect other vector */
            e2 = ex;
            geodesic(0);
```

```
         dp = dc = gc;
         ax = a2;
         /* move positive... */
         while( gc <= dp )
         {
            dp = gc;
            a2 += TENTHDEG;
            g2 = ga;
            geodesic(0);
         }
         /* move negative in case wrong direction */
         dp = dc = gc;
         while( gc <= dp )
         {
            dp = gc;
            a2 -= TENTHDEG;
            g2 = ga;
            geodesic(0);
         }
         a2 = ax;

         /* now move other azimuth to intersect this vector */
         geodesic(0);
         dp = dc = gc;
         ax = a1;
         /* move negative... */
         while( gc <= dp )
         {
            dp = gc;
            a1 -= TENTHDEG;
            g3 = ga;
            geodesic(0);
         }
         a1 = ax;

         /* now move other elevation to intersect this vector */
         geodesic(0);
         dp = dc = gc;
         ex = e1;
         /* move positive... */
         while( gc <= dp )
         {
            dp = gc;
            e1 += TENTHDEG;
            g4 = ga;
            geodesic(0);
         }
         e1 = ex;
      }
      else
      {
         if( debug )
            printf( "v1/v2 are close\r\n" );
         /* can't get better than this */
         g1 = ga;
         g2 = ga;
         g3 = ga;
         g4 = ga;
      }
   }

   gmin = g1;
   if( g2 < gmin )
      gmin = g2;
   if( g3 < gmin )
      gmin = g3;
   if( g4 < gmin )
      gmin = g4;

   gmax = g1;
   if( g2 > gmax )
      gmax = g2;
   if( g3 > gmax )
      gmax = g3;
   if( g4 > gmax )
      gmax = g4;


   if( (g1 > gmin) && (g1 < gmax) )
```

```
   {
      ga = g1;
   }
   else if( (g2 > gmin) && (g2 < gmax) )
   {
      ga = g2;
   }
   else if( (g3 > gmin) && (g3 < gmax) )
   {
      ga = g3;
   }
   else if( (g4 > gmin) && (g4 < gmax) )
   {
      ga = g4;
   }

   if( (g1 != ga) && (g1 > gmin) && (g1 < gmax) )
   {
      gmax = g1;
   }
   else if( (g2 != ga) && (g2 > gmin) && (g2 < gmax) )
   {
      gmax = g2;
   }
   else if( (g3 != ga) && (g3 > gmin) && (g3 < gmax) )
   {
      gmax = g3;
   }
   else if( (g4 != ga) && (g4 > gmin) && (g4 < gmax) )
   {
      gmax = g4;
   }

   gmin = ga;

   if( gmin > gmax )
   {
      ga = gmin;
      gmin = gmax;
      gmax = ga;
   }

   gc = fabs(gmax-gmin)/2;
   ga = gmin + gc;
   gc /= ga;

   if( !quiet )
   {
      printf( "\r\n" );
   }

   if( debug )
      printf("Blob coords: %g, %g, %g, %g\r\n", g1, g2, g3, g4 );
   if( !quiet )
   {
      printf("Blob method = " );
   }
   else
   {
      printf(", ");
   }

   printf( "%g, %g%%\r\n", floor((ga+0.05)*10)/10, floor((gc*100+0.05)*10)/10 );

   return( 0 );
}


/*-------------------------------------------------------------------------
  A standard Geodesic computation, used for comparison.
-------------------------------------------------------------------------*/

void geodesic( int pr )
{
   double f,d1,d2;

   f = sin(e1)*sin(e2) - cos(e1)*cos(e2)*(cos(a1)*cos(a2)-sin(a1)*sin(a2));
   d1 = bl*(cos(e1)*cos(a1)+f*cos(e2)*cos(a2))/(1-f*f);
   d2 = bl*(cos(e2)*cos(a2)+f*cos(e1)*cos(a1))/(1-f*f);
```

```
   ga = d1*d2*(sin(e1)+sin(e2))/(d1+d2);
   gc = bl*fabs( (cos(e2)*sin(e1)*sin(a2)-cos(e1)*sin(e2)*sin(a1))/(ga*sqrt(1-f*f)) );

   if( pr )
   {
      if(!quiet)
      {
         printf( "\r\n" );
         printf( "   Geodesic =" );
      }
      printf( " %g, %g%%", floor((ga+0.05)*10)/10, floor(((gc*100)+0.05)*10)/10 );
   }

}
```

APPENDIX B: CONVERGENCE TEST DATA

1) Low Elevation
Both near baseline
One moving up in elevation

2) Low Elevation
One near baseline
Other moving away from baseline

3) Low Elevation
Both start 1/3 from baseline
One moving away from baseline

4) Low Elevation
Both start 1/3 from baseline
One moving up in elevation

5) Low Elevation
Both start 2/3 from baseline
One moving up in elevation

6) Low Elevation
Both start 2/3 from baseline
One moving away from baseline

7) Medium Elevation
Both start near from baseline
One moving away from baseline

8) High Elevation
Both start near from baseline
One moving away from baseline

9) High Elevation
Both start 1/3 from baseline
One moving away from baseline

10) High Elevation
Both start 2/3 from baseline
One moving up in eleavation

11) Low Elevation
One 120deg from baseline
One moving up in eleavation

```
Hemispherical Iteration Tracking Method
Convergence Test 1

Low Elevation
Both near baseline
One moving up in elevation

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 10 20 10 20    18.5,    0%,    18.5,  0.2%,
100 10 20 10 21    19.0,  2.2%,    19.1,  1.2%,
100 10 20 10 22    19.5,  4.3%,    19.6,  2.4%,
100 10 20 10 23    19.9,  6.2%,    20.1,  3.4%,
100 10 20 10 24    20.4,  8.0%,    20.6,  4.3%,
100 10 20 10 25    20.8,  9.6%,    21.1,  5.1%,
100 10 20 10 26    21.3, 11.2%,    21.6,  5.9%,
100 10 20 10 27    21.7, 12.6%,    22.1,  6.7%,
100 10 20 10 28    22.1, 13.9%,    22.5,  7.3%,
100 10 20 10 29    22.5, 15.2%,    22.9,  8.0%,
100 10 20 10 30    22.9, 16.4%,    23.3,  8.6%,
100 10 20 10 31    23.3, 17.5%,    23.7,  9.1%,
100 10 20 10 32    23.6, 18.6%,    24.1,  9.6%,
100 10 20 10 33    24.0, 19.6%,    24.5, 10.1%,
100 10 20 10 34    24.3, 20.5%,    24.8, 10.6%,
100 10 20 10 35    24.7, 21.4%,    25.2, 11.0%,
100 10 20 10 36    25.0, 22.3%,    25.5, 11.5%,
100 10 20 10 37    25.3, 23.1%,    25.9, 11.9%,
100 10 20 10 38    25.7, 23.9%,    26.2, 12.2%,
100 10 20 10 39    26.0, 24.7%,    26.5, 12.6%,
100 10 20 10 40    26.3, 25.4%,    26.8, 12.9%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 2

Low Elevation
One near baseline
Other moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 10 20 10 20    18.5,    0%,    18.5,  0.2%,
100 10 20 11 20    18.5,  4.2%,    18.7,  2.3%,
100 10 20 12 20    18.5,  8.3%,    18.8,  4.6%,
100 10 20 13 20    18.5, 12.4%,    19.0,  6.9%,
100 10 20 14 20    18.6, 16.3%,    19.2,  9.1%,
100 10 20 15 20    18.6, 20.2%,    19.4, 11.3%,
100 10 20 16 20    18.6, 24.0%,    19.5, 13.4%,
100 10 20 17 20    18.6, 27.7%,    19.7, 15.5%,
100 10 20 18 20    18.6, 31.3%,    19.9, 17.5%,
100 10 20 19 20    18.6, 34.8%,    20.1, 19.5%,
100 10 20 20 20    18.5, 38.3%,    20.3, 21.5%,
100 10 20 21 20    18.5, 41.7%,    20.4, 23.5%,
100 10 20 22 20    18.5, 45.1%,    20.6, 25.4%,
100 10 20 23 20    18.5, 48.4%,    20.8, 27.2%,
100 10 20 24 20    18.4, 51.6%,    21.0, 29.1%,
100 10 20 25 20    18.4, 54.8%,    21.1, 30.9%,
100 10 20 26 20    18.3, 57.9%,    21.3, 32.6%,
100 10 20 27 20    18.3, 61.0%,    21.5, 34.4%,
100 10 20 28 20    18.2, 64.0%,    21.7, 36.1%,
100 10 20 29 20    18.2, 67.0%,    21.9, 37.8%,
100 10 20 30 20    18.1, 70.0%,    22.0, 39.4%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 3

Low Elevation
Both start 1/3 from baseline
One moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 30 20 30 20    21.0,    0%,    21.0,  0.1%,
100 30 20 31 20    21.1,  2.4%,    21.3,  2.1%,
100 30 20 32 20    21.2,  4.7%,    21.5,  4.1%,
100 30 20 33 20    21.3,  7.0%,    21.8,  6.0%,
100 30 20 34 20    21.4,  9.2%,    22.0,  7.9%,
100 30 20 35 20    21.5, 11.4%,    22.3,  9.8%,
100 30 20 36 20    21.6, 13.5%,    22.6, 11.6%,
100 30 20 37 20    21.7, 15.5%,    22.8, 13.4%,
100 30 20 38 20    21.8, 17.5%,    23.1, 15.1%,
100 30 20 39 20    21.9, 19.4%,    23.3, 16.8%,
100 30 20 40 20    22.0, 21.3%,    23.6, 18.5%,
100 30 20 41 20    22.0, 23.1%,    23.9, 20.1%,
100 30 20 42 20    22.1, 24.9%,    24.2, 21.6%,
100 30 20 43 20    22.2, 26.6%,    24.4, 23.2%,
100 30 20 44 20    22.3, 28.3%,    24.7, 24.7%,
100 30 20 45 20    22.4, 29.9%,    25.0, 26.1%,
100 30 20 46 20    22.5, 31.5%,    25.2, 27.6%,
100 30 20 47 20    22.5, 33.1%,    25.5, 29.0%,
100 30 20 48 20    22.6, 34.6%,    25.8, 30.3%,
100 30 20 49 20    22.7, 36.1%,    26.1, 31.6%,
100 30 20 50 20    22.8, 37.5%,    26.4, 32.9%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 4

Low Elevation
Both start 1/3 from baseline
One moving up in elevation

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 30 20 30 20    21.0,     0%,    21.0,  0.1%,
100 30 20 30 21    21.6,  4.3%,    21.8,  3.5%,
100 30 20 30 22    22.1,  8.3%,    22.6,  6.7%,
100 30 20 30 23    22.7, 12.1%,    23.3,  9.5%,
100 30 20 30 24    23.3, 15.7%,    24.0, 12.0%,
100 30 20 30 25    23.8, 19.1%,    24.7, 14.4%,
100 30 20 30 26    24.4, 22.3%,    25.3, 16.5%,
100 30 20 30 27    24.9, 25.3%,    25.9, 18.5%,
100 30 20 30 28    25.5, 28.3%,    26.5, 20.3%,
100 30 20 30 29    26.0, 31.0%,    27.0, 22.0%,
100 30 20 30 30    26.5, 33.7%,    27.6, 23.6%,
100 30 20 30 31    27.0, 36.2%,    28.0, 25.0%,
100 30 20 30 32    27.6, 38.7%,    28.5, 26.4%,
100 30 20 30 33    28.1, 41.0%,    29.0, 27.7%,
100 30 20 30 34    28.5, 43.2%,    29.4, 28.9%,
100 30 20 30 35    29.0, 45.4%,    29.8, 30.1%,
100 30 20 30 36    29.5, 47.5%,    30.2, 31.2%,
100 30 20 30 37    30.0, 49.5%,    30.6, 32.3%,
100 30 20 30 38    30.4, 51.4%,    31.0, 33.3%,
100 30 20 30 39    30.9, 53.3%,    31.3, 34.2%,
100 30 20 30 40    31.3, 55.1%,    31.6, 35.1%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 5

Low Elevation
Both start 2/3 from baseline
One moving up in elevation

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 60 20 60 20    36.4,    0%,    36.4,  0.1%,
100 60 20 60 21    37.4,  4.9%,    38.1,  6.0%,
100 60 20 60 22    38.3,  9.5%,    39.5, 11.4%,
100 60 20 60 23    39.3, 14.0%,    40.8, 16.2%,
100 60 20 60 24    40.2, 18.3%,    41.8, 20.5%,
100 60 20 60 25    41.1, 22.4%,    42.6, 24.5%,
100 60 20 60 26    41.9, 26.3%,    43.3, 28.2%,
100 60 20 60 27    42.7, 30.1%,    43.9, 31.6%,
100 60 20 60 28    43.5, 33.8%,    44.3, 34.8%,
100 60 20 60 29    44.3, 37.4%,    44.7, 37.8%,
100 60 20 60 30    45.0, 40.9%,    44.9, 40.7%,
100 60 20 60 31    45.7, 44.3%,    45.0, 43.4%,
100 60 20 60 32    46.4, 47.7%,    45.1, 46.1%,
100 60 20 60 33    47.0, 50.9%,    45.1, 48.6%,
100 60 20 60 34    47.5, 54.2%,    45.1, 51.0%,
100 60 20 60 35    48.0, 57.3%,    45.0, 53.3%,
100 60 20 60 36    48.5, 60.5%,    44.9, 55.6%,
100 60 20 60 37    48.9, 63.6%,    44.8, 57.7%,
100 60 20 60 38    49.3, 66.7%,    44.6, 59.9%,
100 60 20 60 39    49.6, 69.7%,    44.4, 61.9%,
100 60 20 60 40    49.9, 72.8%,    44.2, 63.9%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 6

Low Elevation
Both start 2/3 from baseline
One moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 60 20 60 20    36.4,    0%,    36.4,  0.1%,
100 60 20 61 20    37.0,  0.9%,    37.1,  1.2%,
100 60 20 62 20    37.5,  1.8%,    37.8,  2.3%,
100 60 20 63 20    38.1,  2.6%,    38.6,  3.4%,
100 60 20 64 20    38.7,  3.4%,    39.3,  4.4%,
100 60 20 65 20    39.4,  4.2%,    40.1,  5.4%,
100 60 20 66 20    40.0,  4.9%,    40.9,  6.4%,
100 60 20 67 20    40.7,  5.7%,    41.8,  7.3%,
100 60 20 68 20    41.4,  6.3%,    42.6,  8.2%,
100 60 20 69 20    42.1,  7.0%,    43.5,  9.0%,
100 60 20 70 20    42.8,  7.6%,    44.4,  9.8%,
100 60 20 71 20    43.6,  8.2%,    45.3, 10.6%,
100 60 20 72 20    44.4,  8.7%,    46.3, 11.3%,
100 60 20 73 20    45.3,  9.2%,    47.3, 12.0%,
100 60 20 74 20    46.1,  9.7%,    48.3, 12.6%,
100 60 20 75 20    47.0, 10.2%,    49.4, 13.2%,
100 60 20 76 20    48.0, 10.6%,    50.5, 13.8%,
100 60 20 77 20    49.0, 11.0%,    51.6, 14.3%,
100 60 20 78 20    50.0, 11.4%,    52.8, 14.8%,
100 60 20 79 20    51.1, 11.7%,    54.1, 15.3%,
100 60 20 80 20    52.2, 12.0%,    55.4, 15.7%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 7

Medium Elevation
Both start near from baseline
One moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 20 45 20 45    53.2,    0%,    53.2,    0%,
100 20 45 21 45    53.4,  1.5%,    53.5,  0.8%,
100 20 45 22 45    53.5,  3.1%,    53.9,  1.6%,
100 20 45 23 45    53.7,  4.6%,    54.2,  2.4%,
100 20 45 24 45    53.9,  6.1%,    54.5,  3.2%,
100 20 45 25 45    54.1,  7.5%,    54.8,  4.0%,
100 20 45 26 45    54.2,  9.0%,    55.1,  4.7%,
100 20 45 27 45    54.4, 10.4%,    55.5,  5.5%,
100 20 45 28 45    54.6, 11.8%,    55.8,  6.3%,
100 20 45 29 45    54.7, 13.2%,    56.1,  7.0%,
100 20 45 30 45    54.9, 14.6%,    56.4,  7.7%,
100 20 45 31 45    55.1, 16.0%,    56.8,  8.4%,
100 20 45 32 45    55.3, 17.3%,    57.1,  9.2%,
100 20 45 33 45    55.4, 18.7%,    57.4,  9.9%,
100 20 45 34 45    55.6, 20.0%,    57.8, 10.5%,
100 20 45 35 45    55.8, 21.2%,    58.1, 11.2%,
100 20 45 36 45    55.9, 22.5%,    58.4, 11.9%,
100 20 45 37 45    56.1, 23.8%,    58.8, 12.6%,
100 20 45 38 45    56.3, 25.0%,    59.1, 13.2%,
100 20 45 39 45    56.5, 26.2%,    59.5, 13.9%,
100 20 45 40 45    56.6, 27.4%,    59.8, 14.5%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 8

High Elevation
Both start near from baseline
One moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 20 70 20 70    146.2,    0%,   146.2,    0%,
100 20 70 21 70    146.7,  0.6%,   146.8,  0.3%,
100 20 70 22 70    147.1,  1.2%,   147.4,  0.6%,
100 20 70 23 70    147.6,  1.8%,   148.0,  0.9%,
100 20 70 24 70    148.1,  2.3%,   148.7,  1.2%,
100 20 70 25 70    148.6,  2.9%,   149.3,  1.5%,
100 20 70 26 70    149.0,  3.5%,   149.9,  1.7%,
100 20 70 27 70    149.5,  4.0%,   150.6,  2.0%,
100 20 70 28 70    150.0,  4.6%,   151.2,  2.3%,
100 20 70 29 70    150.5,  5.2%,   151.8,  2.6%,
100 20 70 30 70    151.0,  5.7%,   152.5,  2.9%,
100 20 70 31 70    151.5,  6.3%,   153.1,  3.1%,
100 20 70 32 70    152.0,  6.8%,   153.8,  3.4%,
100 20 70 33 70    152.5,  7.3%,   154.4,  3.7%,
100 20 70 34 70    153.0,  7.9%,   155.1,  3.9%,
100 20 70 35 70    153.5,  8.4%,   155.8,  4.2%,
100 20 70 36 70    154.1,  8.9%,   156.4,  4.4%,
100 20 70 37 70    154.6,  9.4%,   157.1,  4.7%,
100 20 70 38 70    155.1,  9.9%,   157.8,  5.0%,
100 20 70 39 70    155.7, 10.4%,   158.5,  5.2%,
100 20 70 40 70    156.2, 10.9%,   159.2,  5.5%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 9

High Elevation
Both start 1/3 from baseline
One moving away from baseline

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 30 70 30 70    158.6,   0%,    158.6,    0%,
100 30 70 31 70    159.4, 0.5%,    159.6,  0.3%,
100 30 70 32 70    160.2, 1.1%,    160.5,  0.5%,
100 30 70 33 70    161.1, 1.6%,    161.5,  0.8%,
100 30 70 34 70    161.9, 2.1%,    162.5,  1.1%,
100 30 70 35 70    162.7, 2.6%,    163.5,  1.3%,
100 30 70 36 70    163.6, 3.1%,    164.5,  1.6%,
100 30 70 37 70    164.4, 3.6%,    165.5,  1.8%,
100 30 70 38 70    165.3, 4.1%,    166.5,  2.1%,
100 30 70 39 70    166.2, 4.6%,    167.5,  2.3%,
100 30 70 40 70    167.1, 5.1%,    168.6,  2.6%,
100 30 70 41 70    168.0, 5.6%,    169.6,  2.8%,
100 30 70 42 70    168.9, 6.1%,    170.7,  3.1%,
100 30 70 43 70    169.8, 6.5%,    171.7,  3.3%,
100 30 70 44 70    170.7, 7.0%,    172.8,  3.5%,
100 30 70 45 70    171.7, 7.4%,    173.9,  3.8%,
100 30 70 46 70    172.6, 7.9%,    175.0,  4.0%,
100 30 70 47 70    173.6, 8.3%,    176.1,  4.2%,
100 30 70 48 70    174.6, 8.7%,    177.2,  4.4%,
100 30 70 49 70    175.6, 9.2%,    178.4,  4.6%,
100 30 70 50 70    176.6, 9.6%,    179.6,  4.8%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 10

High Elevation
Both start 2/3 from baseline
One moving up in eleavation

 BL A1 E1 A2 E2    GEO             HIT
---------------------------------------------------
100 60 70 60 70    274.7,    0%,   274.8,    0%,
100 60 70 60 71    281.8,  1.6%,   282.5,  0.8%,
100 60 70 60 72    287.9,  3.3%,   289.2,  1.7%,
100 60 70 60 73    292.7,  4.9%,   294.6,  2.5%,
100 60 70 60 74    296.1,  6.6%,   298.3,  3.4%,
100 60 70 60 75    297.8,  8.3%,   300.3,  4.2%,
100 60 70 60 76    297.6, 10.1%,   300.4,  5.1%,
100 60 70 60 77    295.4, 11.9%,   298.5,  6.1%,
100 60 70 60 78    291.1, 13.9%,   294.5,  7.0%,
100 60 70 60 79    284.7, 16.1%,   288.5,  8.1%,
100 60 70 60 80    276.3, 18.4%,   280.6,  9.2%,
100 60 70 60 81    266.1, 20.9%,   271.0, 10.4%,
100 60 70 60 82    254.3, 23.7%,   259.9, 11.7%,
100 60 70 60 83    241.1, 26.8%,   247.7, 13.1%,
100 60 70 60 84    226.9, 30.3%,   234.5, 14.7%,
100 60 70 60 85    212.0, 34.2%,   220.8, 16.5%,
100 60 70 60 86    196.7, 38.6%,   206.8, 18.4%,
100 60 70 60 87    181.4, 43.6%,   192.7, 20.5%,
100 60 70 60 88    166.3, 49.2%,   178.8, 22.9%,
100 60 70 60 89    151.5, 55.7%,   165.2, 25.5%,
100 60 70 60 90    137.4, 63.0%,   152.1, 28.5%,
```

```
Hemispherical Iteration Tracking Method
Convergence Test 11

Low Elevation
One 120deg from baseline
One moving up in eleavation

 BL A1 E1 A2 E2    GEO             HIT             BLOB
---------------------------------------------------------------
100 120 30 30 10   37.0, 55.5%,    27.6, 56.8%,    28.4,  7.4%
100 120 30 30 11   39.4, 46.8%,    30.4, 47.0%,    30.6, 10.1%
100 120 30 30 12   41.7, 38.8%,    33.5, 38.3%,    32.7, 12.5%
100 120 30 30 13   44.2, 31.5%,    36.7, 30.6%,    35.6, 12.2%
100 120 30 30 14   46.7, 24.8%,    40.2, 23.7%,    39.1, 10.4%
100 120 30 30 15   49.2, 18.5%,    43.9, 17.4%,    42.8,  8.4%
100 120 30 30 16   51.7, 12.7%,    47.8, 11.8%,    46.7,  6.2%
100 120 30 30 17   54.2,  7.2%,    51.8,  6.6%,    51.0,  3.9%
100 120 30 30 18   56.7,  2.1%,    55.9,  1.9%,    55.6,  1.2%
100 120 30 30 19   59.1,  2.7%,    61.1,  2.4%,    59.4,  0.4%
100 120 30 30 20   61.5,  7.3%,    67.0,  6.2%,    62.3,  1.2%
100 120 30 30 21   63.8, 11.6%,    72.9,  9.6%,    65.1,  2.1%
100 120 30 30 22   66.0, 15.9%,    78.5, 12.8%,    68.0,  2.9%
100 120 30 30 23   68.0, 19.9%,    83.7, 15.8%,    70.8,  3.9%
100 120 30 30 24   69.9, 23.9%,    88.2, 18.7%,    73.5,  4.9%
100 120 30 30 25   71.5, 27.8%,    91.8, 21.6%,    76.2,  6.1%
100 120 30 30 26   73.0, 31.7%,    94.3, 24.6%,    78.7,  7.3%
100 120 30 30 27   74.1, 35.7%,    95.4, 27.7%,    81.2,  8.7%
100 120 30 30 28   75.0, 39.6%,    95.3, 31.0%,    83.6, 10.2%
100 120 30 30 29   75.6, 43.7%,    93.7, 34.8%,    85.8, 11.9%
100 120 30 30 30   75.8, 47.9%,    90.9, 38.9%,    87.9, 13.7%
```

SUMMARY

Historically, the choice of formulae for model rocketry altitude data reduction has been limited to linear closed-form equations. Using simple trigonometric functions, these equations require only a pocket calculator or, if necessary, could be derived by hand using sine and cosine tables. In recent years the widespread availability of programmable calculators and portable computers has automated these calculations. In fact, the typical portable computer has more than enough computing power to perform vastly more complex methods of data reduction beyond the static closed-form algorithms.

The main objective of this project was to implement a computer software program for a more accurate method of determining a model rocket's likely altitude given two-station theodolite angles. The mathematical basis of the new method has a strong similarity to the closed form Geodesic method, but adds weighting for real-world effects of optically tracking a model rocket to ejection. The grouping of errors into a "region of uncertainty" is approximated by a hemispherical shape which is located using an iterative software technique, hence the name "Hemispherical Iteration Tracking".

The software was tested for various regions in 3D space relative to the trackers and the convergence of the HIT method was compared to the Geodesic method with favorable results. Additionally, a set of data from a particularly problematic NAR-sanctioned regional was used to compare the new method with the Geodesic equations; the results showed that the closure rate would have improved from only 31% of tracked flights closed to over 88% closed.

NAR competitors would benefit from this new data reduction method by improving the closure rate for altitude events, offering a more even playing field when existing algorithms fail to close reliably under certain conditions. Further statistical analysis would be needed before approving the new method for NAR-sanctioned competition and record attempts.

(297 words)