BATCH LEAST SQUARES DIFFERENTIAL CORRECTION
OF A GEOCENTRIC ORBIT

PART 2 - MANUAL CORRECTION WORKSHEET

Roger L. Mansfield, May 15, 2025
http://astroger.com/

In this worksheet we differentially correct (DC) the orbit of an artificial Earth satellite or space probe using a test case specified in worksheet Gd1, or in a worksheet derived from Gd1. You should open worksheet Gd1, or your own worksheet derived from Gd1, and click on "Calculate Worksheet" from the Math menu now, if you have not already done so.

The process that we will use in this worksheet is documented in Refs. [1] and [2] for the differential correction of Earth orbits using radar observations. However, we will use optical observations in this worksheet. The batch equation of differential correction (BEDC) is:

$$X_o' = X_o + (A^T WA)^{-1} A^T W [Y - F(X_o)].$$

Here X_o is the initial estimate of the state vector, i.e., position and velocity, at epoch t_o. X_o' is the "improved" estimate of X_o at t_o, obtained by adding $(A^T WA)^{-1} A^T W [Y - F(X_o)]$ to X_o.

If we let n be the number of observations, then Y is a 2n-by-1 column vector of measurements, since for our problem in geocentric motion the measurements are topocentric right ascension (RA, or α) and topocentric declination (DEC, or δ). If we denote the total number of measurements by N, then N = 2n.

$F(X_o)$ is thus an N-by-1column vector of "computed" measurements. What this means is that the RA and DEC for each observation are computed via our UPM model of two-body motion, by propagating the current estimate, X_o to the observation times t_i for i = 1, ... , n, and by then computing the topocentric RA and DEC at each observation time, given the specified location of the observer. We say "current estimate, X_o" because we will find it necessary to iterate on the BEDC, testing for convergence at each iteration by means of a criterion we will define below. If we have convergence on a given iteration, then we stop and convert the solution to conic elements. But if we do not have convergence, then we replace X_o by X_o' and solve the BEDC again, i.e., iterate. (We could also implement an iteration counter and stop the DC if some maximum allowable number of iterations is reached without convergence, but that is not needed here because we iterate the BEDC manually by clicking on "Calculate Worksheet".)

$[Y - F(X_o)]$ is the N-by-1 column vector of residuals, in the sense "observed minus computed". The BEDC is a form of the least squares normal equations, N equations in six unknowns, which result when one answers the question, "what is a necessary condition that the weighted sum of squares of the residuals be a minimum?" The residuals are not actually $\Delta\alpha$ and $\Delta\delta$, but rather $\cos \delta\ \Delta\alpha$ and $\Delta\delta$; they are the projections of $\Delta\mathbf{L}$ on $\mathbf{A}$ and $\mathbf{D}$ in turn. (The $\cos \delta$ factor can become quite important when the object passes near a celestial pole, where large changes in α accompany relatively small changes in arc length in the direction of motion.)

A, the "A-matrix", is the N-by-6 array of partial derivatives of the N measurements with respect to the six components of the state vector X_o. We will compute the A-matrix from the O-matrix and the G-matrix, i.e., A = OG. O is the N-by-6 matrix of partials of the measurements with respect to the state vector at observation times t_i, for i = 1, ... , n. G is Goodyear's 6-by-6 state transition matrix, i.e., the 6-by-6 matrix of partials of the state components at times t_i with respect to the state components at t_o. G is therefore a 6-by-6 Jacobian matrix defined at each observation time t_i for i = 1, ... , n.

W is the weight matrix. Under the assumption that the measurements are Gaussian random variables, and are not correlated (Danby [3] has a good discussion of this), W is a diagonal matrix and each diagonal entry is $1/\sigma_i^2$, where σ_i^2 is the variance of measurement i. (In Part 1, we specified the N-by-N identity matrix with values of σ_i = 1.0 radians. More realistic sigmas for the RA and DEC measurements, e.g., the number of radians in one arc-second, would improve the statistics, and yet not change the solution state vector.

Here now is an outline of the steps we will follow:

1. Retrieve the test case values from disk, as specified by worksheet Gd1, or as specified by your own worksheet that was derived from Gd1 by duplication and modification.

Retrieval includes obtaining the initial or current estimate of state, X, and the RMS history matrix. Each time you click on "Calculate Worksheet," GDC performs another iteration of weighted, batch least squares differential correction. At each iteration the corrected values of X are written to disk along with the RMS for that iteration. The corrected values of X thus become the current state estimate for the next iteration, and the RMS history is accumulated so that you can keep track of how the DC is going.

2. Define the functions needed in the DC: **C**, **FG**, **GMAT**, and **FXA**.

3. Obtain the computed measurements, **FX**, and the A-matrix, A, by invoking **FXA**.

4. Compute the residuals, ΔY, the A^TWA matrix ATWA, and the $A^TW\Delta Y$ matrix, ATWΔY.

5. Solve for and apply the corrections to state, ΔX. Compute the current RMS, display the RMS history, and test for convergence.

6. Write the corrected state vector to disk and convert to conic elements.

7. Repeat Steps 1-6, by clicking on "Calculate Worksheet", until convergence is obtained.

As a preliminary, we define some constants that we will need, and set the Mathcad worksheet ORIGIN to 1 so that subscripts start at unity rather than at zero.

$$DegPerRad := \frac{180}{\pi} \qquad \textbf{ORIGIN} \equiv 1$$

$SecPerDeg := 3600.0$

Earth's mean equatorial radius in km:

$SecPerRev := SecPerDeg \cdot 360.0$

$a_e := 6378.137$

1. Retrieve the test case values from disk, as specified by worksheet Gd1, or as specified by your own worksheet that was derived from Gd1 by duplication and/or modification.

$n := \mathrm{READPRN}(\text{"NOBS.prn"})_1$ Number of observations.

$t := \mathrm{READPRN}(\text{"TVALS.prn"})$ Observation times.

$W := \mathrm{READPRN}(\text{"WEIGHTS.prn"})$ Measurement weights matrix.

$R := \mathrm{READPRN}(\text{"RVALS.prn"})$ Values of **R**.

$Y := \mathrm{READPRN}(\text{"YVALS.prn"})$ Values of Y.

$X := \mathrm{READPRN}(\text{"STATE.prn"})$ State vector (corrected by Gdc).

$RMS := \mathrm{READPRN}(\text{"RMS.prn"})$ RMS history for state corrections by Gdc (one entry for each iteration).

$N := 2 \cdot n$ Set number of measurements.

$k := 0.07436684771154$ Set WGS-84 Gaussian constant for geocentric orbital motion. See [10].

$\mu := 1$ Assume that mass of secondary (artificial Earth satellite or space probe) is negligible relative to mass of primary (Earth).

$K := k \cdot \sqrt{\mu}$

$n = 8$ Display number of observations retrieved from disk.

2. Define the functions needed in the DC: **C**, **FG**, **GMAT**, and **FXA**.

For path propagation one needs to calculate only c_0 through c_3, but for the state transition matrix, G, one needs c_0 through c_5. To keep down the length of this worksheet we define one version of **C**, the one that calculates c_0 through c_5. (Remember that since the ORIGIN = 1, the subscripts of the c-functions that we will use outside of the function **C** will range from 1 through 6, rather than from 0 through 5.)

$$
\begin{aligned}
C(x) := \;& N \leftarrow 0 \\
& \text{while } |x| \geq 0.1 \\
& \quad x \leftarrow \frac{x}{4} \\
& \quad N \leftarrow N+1 \\
& c_5 \leftarrow \frac{\left(1-\frac{x}{42}\cdot\left(1-\frac{x}{72}\cdot\left(1-\frac{x}{110}\cdot\left(1-\frac{x}{156}\cdot\left(1-\frac{x}{210}\cdot\left(1-\frac{x}{272}\right)\right)\right)\right)\right)\right)}{120} \\
& c_4 \leftarrow \frac{\left(1-\frac{x}{30}\cdot\left(1-\frac{x}{56}\cdot\left(1-\frac{x}{090}\cdot\left(1-\frac{x}{132}\cdot\left(1-\frac{x}{182}\cdot\left(1-\frac{x}{240}\right)\right)\right)\right)\right)\right)}{24} \\
& c_3 \leftarrow \frac{1}{6} - c_5 \cdot x \\
& c_2 \leftarrow \frac{1}{2} - c_4 \cdot x \\
& c_1 \leftarrow 1 - c_3 \cdot x \\
& c_0 \leftarrow 1 - c_2 \cdot x \\
& \text{while } N > 0 \\
& \quad N \leftarrow N-1 \\
& \quad c_5 \leftarrow \frac{(c_2 \cdot c_3 + c_4 + c_5)}{16} \\
& \quad c_4 \leftarrow \frac{(c_2 \cdot c_2 + c_4 + c_4)}{8} \\
& \quad c_3 \leftarrow \frac{(c_1 \cdot c_2 + c_3)}{4} \\
& \quad c_2 \leftarrow \frac{c_1 \cdot c_1}{2} \\
& \quad c_1 \leftarrow c_1 \cdot c_0 \\
& \quad c_0 \leftarrow 2 \cdot c_0 \cdot c_0 - 1 \\
& \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix}^{\mathrm{T}}
\end{aligned}
$$

Function UKEP solves the uniform Kepler equation for function **FG**. **FG,** in turn, propagates position and velocity for function **FXA**.

$$UKEP\left(\tau, rmag_o, \sigma_o, \alpha\right) :=$$

$$s \leftarrow \frac{\tau}{rmag_o}$$

$$\Delta s \leftarrow s$$

$$\text{while } |\Delta s| \geq 0.00000001$$

$$\quad c \leftarrow C\left(\alpha \cdot s^2\right)$$

$$\quad F \leftarrow rmag_o \cdot s \cdot c_2 + \sigma_o \cdot s^2 \cdot c_3 + s^3 \cdot c_4 - \tau$$

$$\quad DF \leftarrow rmag_o \cdot c_1 + \sigma_o \cdot s \cdot c_2 + s^2 \cdot c_3$$

$$\quad DDF \leftarrow \sigma_o \cdot c_1 + \left(1 - rmag_o \cdot \alpha\right) \cdot s \cdot c_2$$

$$\quad \text{if } DF \geq 0$$

$$\quad\quad m \leftarrow 1$$

$$\quad \text{else}$$

$$\quad\quad m \leftarrow -1$$

$$\quad \Delta s \leftarrow \frac{-5 \cdot F}{\left(DF + m \cdot \sqrt{\left|(4 \cdot DF)^2 - 20 \cdot F \cdot DDF\right|}\right)}$$

$$\quad s \leftarrow s + \Delta s$$

$$s$$

$$FG\left(K, r_o, v_o, \Delta t\right) :=$$

$$\tau \leftarrow K \cdot \Delta t$$

$$rmag_o \leftarrow \sqrt{r_o \cdot r_o}$$

$$\sigma_o \leftarrow r_o \cdot v_o$$

$$\alpha \leftarrow \frac{2}{rmag_o} - v_o \cdot v_o$$

$$s \leftarrow UKEP\left(\tau, rmag_o, \sigma_o, \alpha\right)$$

$$c \leftarrow C\left(\alpha \cdot s^2\right)$$

$$f_r \leftarrow 1 - s^2 \cdot c_3 \cdot rmag_o^{-1}$$

$$g_r \leftarrow \tau - s^3 \cdot c_4$$

$$rmag \leftarrow rmag_o \cdot c_1 + \sigma_o \cdot s \cdot c_2 + s^2 \cdot c_3$$

$$f_v \leftarrow -s \cdot c_2 \cdot \left(rmag \cdot rmag_o\right)^{-1}$$

$$g_v \leftarrow 1 - s^2 \cdot c_3 \cdot rmag^{-1}$$

$$\begin{bmatrix} K & \alpha & rmag_o & f_r & f_v \\ \tau & s & rmag & g_r & g_v \end{bmatrix}$$

Function **GMAT** provides the state transition matrix for function **FXA**.

The state transition matrix formulation that we use below is based upon the seminal works of Goodyear [4], [5]. See also Shepperd [6], Battin [7], and Der [8] for more recent expositions.

Before defining **GMAT**, we define functions $\mathbf{S}_{11}$, $\mathbf{S}_{12}$, $\mathbf{S}_{21}$, and $\mathbf{S}_{22}$ just to make **GMAT** fit horizontally and vertically within the margins of a single Mathcad page.

$$S_{11}\left(rmag_o, rmag, f_r, g_r, f_v, g_v, s\right) := \begin{bmatrix} -\dfrac{f_v \cdot s_2 + \dfrac{f_r - 1}{rmag_o}}{rmag_o} & -f_v \cdot s_3 \\ \dfrac{\left(f_r - 1\right) \cdot s_2}{rmag_o} & \left(f_r - 1\right) \cdot s_3 \end{bmatrix}$$

$$S_{12}\left(rmag_o, rmag, f_r, g_r, f_v, g_v, s\right) := \begin{bmatrix} -f_v \cdot s_3 & -\left(g_v - 1\right) \cdot s_3 \\ \left(f_r - 1\right) \cdot s_3 & g_r \cdot s_3 \end{bmatrix}$$

$$S_{21}\left(rmag_o, rmag, f_r, g_r, f_v, g_v, s\right) := \begin{bmatrix} -f_v \cdot \left(\dfrac{s_1}{rmag_o \cdot rmag} + \dfrac{1}{rmag^2} + \dfrac{1}{rmag_o^{\,2}}\right) & -\dfrac{f_v \cdot s_2 + \dfrac{g_v - 1}{rmag}}{rmag} \\ \dfrac{f_v \cdot s_2 + \dfrac{\left(f_r - 1\right)}{rmag_o}}{rmag_o} & f_v \cdot s_3 \end{bmatrix}$$

$$S_{22}\left(rmag_o, rmag, f_r, g_r, f_v, g_v, s\right) := \begin{bmatrix} -\dfrac{f_v \cdot s_2 + \dfrac{g_v - 1}{rmag}}{rmag} & \dfrac{-\left(g_v - 1\right) \cdot s_2}{rmag} \\ f_v \cdot s_3 & \left(g_v - 1\right) \cdot s_3 \end{bmatrix}$$

(Note that because ORIGIN = 1, the subscripts of the c-functions and Goodyear's s-functions range from 1 to 6 rather than from 0 to 5. It is especially important to note this difference when checking the **GMAT** formulas against Goodyear's original works.)

$$
GMAT(M, r_o, v_o, r, v) := \left\|
\begin{array}{l}
\tau \leftarrow M_{2,1} \\
\alpha \leftarrow M_{1,2} \\
s \leftarrow M_{2,2} \\
rmag_o \leftarrow M_{1,3} \\
rmag \leftarrow M_{2,3} \\
f_r \leftarrow M_{1,4} \\
g_r \leftarrow M_{2,4} \\
f_v \leftarrow M_{1,5} \\
g_v \leftarrow M_{2,5} \\
c \leftarrow C(\alpha \cdot s^2) \\
svec \leftarrow \left[c_1 \quad s \cdot c_2 \quad s^2 \cdot c_3 \quad s^3 \cdot c_4 \quad s^4 \cdot c_5 \quad s^5 \cdot c_6 \right]^{\mathrm{T}} \\
U \leftarrow svec_3 \cdot \tau + s \cdot svec_5 - 3 \cdot svec_6 \\
A \leftarrow \mathrm{augment}(r, v) \\
B \leftarrow \mathrm{augment}(r_o, v_o)^{\mathrm{T}} \\
a_o \leftarrow \dfrac{-r_o}{rmag_o^{\,3}} \\
a \leftarrow \dfrac{-r}{rmag^3} \\
I \leftarrow \mathrm{identity}(3) \\
G_{11} \leftarrow f_r \cdot I + U \cdot v \cdot a_o^{\mathrm{T}} + A \cdot S_{11}(rmag_o, rmag, f_r, g_r, f_v, g_v, svec) \cdot B \\
G_{12} \leftarrow g_r \cdot I - U \cdot v \cdot v_o^{\mathrm{T}} + A \cdot S_{12}(rmag_o, rmag, f_r, g_r, f_v, g_v, svec) \cdot B \\
G_{21} \leftarrow f_v \cdot I + U \cdot a \cdot a_o^{\mathrm{T}} + A \cdot S_{21}(rmag_o, rmag, f_r, g_r, f_v, g_v, svec) \cdot B \\
G_{22} \leftarrow g_v \cdot I - U \cdot a \cdot v_o^{\mathrm{T}} + A \cdot S_{22}(rmag_o, rmag, f_r, g_r, f_v, g_v, svec) \cdot B \\
\mathrm{stack}(\mathrm{augment}(G_{11}, G_{12}), \mathrm{augment}(G_{21}, G_{22}))
\end{array}
\right.
$$

Function **FXA** calculates **FX**, the N-by-1 computed measurements vector, and A, the N-by-6 A-matrix of partials of the measurements at time t_i with respect to the state at time t_o. (Note that in the call to function **FG**, the time since epoch is converted from days to minutes by multiplying by 1440 minutes per day.)

```
FXA(K, r_o, v_o) := | for i ∈ 1 .. n
                    |   M ← FG(K, r_o, v_o, (t_i − t_1)·1440)
                    |   τ ← M_{2,1}
                    |   α ← M_{1,2}
                    |   s ← M_{2,2}
                    |   rmag_o ← M_{1,3}
                    |   rmag ← M_{2,3}
                    |   f_r ← M_{1,4}
                    |   g_r ← M_{2,4}
                    |   r ← f_r·r_o + g_r·v_o
                    |   v ← M_{1,5}·r_o + M_{2,5}·v_o
                    |   ρ ← r + (R)^<i>
                    |   ρmag ← √(ρ·ρ)
                    |   j ← 2·i − 1
                    |   k ← j + 1
                    |   RA ← angle(ρ_1, ρ_2)
                    |   DEC ← asin(ρ_3 / ρmag)
                    |   FX_j ← cos(Y_k)·RA
                    |   FX_k ← DEC
                    |   d_j ← ρmag
                    |   d_k ← ρmag
                    |   O ← [ −sin(RA)/ρmag                cos(RA)/ρmag                 0               0 0 0 ]
                    |       [ −sin(DEC)·cos(RA)/ρmag   −sin(DEC)·sin(RA)/ρmag   cos(DEC)/ρmag   0 0 0 ]
                    |   G ← GMAT(M, r_o, v_o, r, v)
                    |   if i = 1
                    |     | A ← O·G
                    |   else
                    |     | A ← stack(A, O·G)
                    | A ← augment(FX, A)
                    | augment(d, A)
```

3. Obtain the computed measurements, **FX**, and the A-matrix, A, by invoking **FXA**.

$$r_o := \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \qquad v_o := \begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \cdot \frac{1}{K}$$

$$M := FXA\left(K, r_o, v_o\right)$$

$$FX := M^{\langle 2 \rangle} \qquad A := \text{submatrix}\left(M, 1, N, 3, 8\right)$$

Extract topocentric distance values for information about computed ranges for each of the observations.

$$d := M^{\langle 1 \rangle}$$

(Click on the **FX** column vector and scroll down to see all N entries.)

(Click on the A matrix and scroll down to see all N rows. Scroll right to see all 6 columns.)

$$FX = \begin{bmatrix} 0.43778172 \\ 0.22941533 \\ 0.43743911 \\ 0.22938841 \\ 0.43726068 \\ 0.22937247 \\ 0.46236678 \\ 0.23881693 \\ 0.46295879 \\ 0.23929609 \\ 0.46332058 \\ 0.23958793 \\ 0.4635863 \\ 0.23980218 \\ 0.46439392 \\ 0.24045154 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.01191692 & 0.02469761 & 0 & \\ -0.00561644 & -0.00271 & 0.02670387 & \\ -0.01234743 & 0.02561302 & 2.59097475 \cdot 10^{-10} & \\ -0.00582394 & -0.00280758 & 0.02768908 & \\ -0.01257441 & 0.02609623 & 6.34769613 \cdot 10^{-10} & \\ -0.0059334 & -0.002859 & 0.02820904 & \\ -0.02171189 & 0.04212544 & -0.00000032 & \\ -0.00996494 & -0.00513569 & 0.04604651 & \\ -0.02271482 & 0.04399883 & -0.0000004 & \\ -0.01042861 & -0.00538344 & 0.04810529 & \\ -0.02333093 & 0.04514687 & -0.00000045 & \\ -0.01071354 & -0.00553604 & 0.0493674 & \\ -0.02378565 & 0.04599281 & -0.00000049 & \\ -0.01092388 & -0.00564886 & 0.05029763 & \\ -0.02517573 & 0.04857167 & -0.00000063 & \\ -0.0115671 & -0.00599479 & 0.05313458 & \ldots \end{bmatrix}$$

4. Compute the residuals, ΔY, the $A^T WA$ matrix ATWA, and the $A^T W\Delta Y$ matrix, ATWΔY.

$\Delta Y := Y - FX$ $\qquad ATWA := A^{T} \cdot W \cdot A$ $\qquad ATW\Delta Y := A^{T} \cdot W \cdot \Delta Y$

$$\Delta Y = \begin{bmatrix} -0.00000129 \\ -0.00000155 \\ 0.00000254 \\ 0.00000235 \\ -0.0000015 \\ 4.80272694 \cdot 10^{-9} \\ -0.00000093 \\ -0.00000372 \\ 0.00000241 \\ 0.0000013 \\ 0.00000046 \\ -0.00000028 \\ -0.00000021 \\ 0.00000063 \\ -0.00000148 \\ 0.00000126 \end{bmatrix}$$

Topocentric distance values.

$$d = \begin{bmatrix} 36.46661089 \\ 36.46661089 \\ 35.16922241 \\ 35.16922241 \\ 34.52099406 \\ 34.52099406 \\ 21.08943369 \\ 21.08943369 \\ 20.18266305 \\ 20.18266305 \\ 19.66414892 \\ 19.66414892 \\ 19.29863011 \\ 19.29863011 \\ 18.2628761 \\ 18.2628761 \end{bmatrix}$$

Display ATWA and ATWΔY matrices.

$$ATWA = \begin{bmatrix} 0.00386 & -0.005877 & -0.003132 & 0.021158 & -0.031922 & -0.016989 \\ -0.005877 & 0.012347 & -0.001602 & -0.03192 & 0.066475 & -0.008775 \\ -0.003132 & -0.001602 & 0.0145 & -0.016989 & -0.008775 & 0.078292 \\ 0.021158 & -0.03192 & -0.016989 & 0.132939 & -0.200411 & -0.10666 \\ -0.031922 & 0.066475 & -0.008775 & -0.200411 & 0.41705 & -0.055135 \\ -0.016989 & -0.008775 & 0.078292 & -0.10666 & -0.055135 & 0.491303 \end{bmatrix}$$

$$ATW\Delta Y = \begin{bmatrix} -6.59916686 \cdot 10^{-17} \\ 8.97359939 \cdot 10^{-18} \\ 2.71849851 \cdot 10^{-16} \\ -1.87218567 \cdot 10^{-16} \\ -1.3165537 \cdot 10^{-15} \\ 3.80787018 \cdot 10^{-15} \end{bmatrix}$$

5. Solve for and apply the corrections to state, ΔX. Compute the current RMS error, display the RMS error history, and test for convergence. See [9], Chapter 15 for documentation of the WRMS and PWRMS convergence criteria below.

$$\Delta X := ATWA^{-1} \cdot ATW\Delta Y \qquad \Delta X = \begin{bmatrix} 6.1724988 \cdot 10^{-10} \\ 2.92281652 \cdot 10^{-10} \\ 1.57147659 \cdot 10^{-10} \\ 2.46884139 \cdot 10^{-10} \\ 1.31843432 \cdot 10^{-10} \\ 6.99234614 \cdot 10^{-11} \end{bmatrix}$$

$$X := \mathrm{stack}\left(r_o, v_o\right) + \Delta X$$

$$WSS := \sum_{i=1}^{N} \left(W_{i,i} \cdot \Delta Y_i\right)^2$$

Weighted sum of squares of residuals.

$$WSS = 4.49952222 \cdot 10^{-11}$$

$$WRMS := \sqrt{\frac{WSS}{N}} \cdot a_e$$

Weighted RMS in km.

$$WRMS = 0.01069589$$

$$PWSS := \sum_{i=1}^{6} \left(ATW\Delta Y_i \cdot \Delta X_i\right)$$

Predicted weighted sum of squares of residuals for next iteration, in km.

$$PWSS = 5.10692472 \cdot 10^{-26}$$

$$PWRMS := \sqrt{\frac{\left|WSS - PWSS\right|}{N}} \cdot a_e$$

Predicted weighted RMS for next iteration, in km.

$$PWRMS = 0.01069589$$

$$Converged := \begin{Vmatrix} \text{if } \left|WRMS - PWRMS\right| < 0.001 \cdot WRMS \\ \quad \Vert\ 1 \\ \text{else} \\ \quad \Vert\ 0 \end{Vmatrix}$$

$$Converged = 1$$

$$\text{APPENDPRN}\left(\text{“RMS.prn”}, \left[WRMS \;\; Converged\right]\right) = \begin{bmatrix} 0 & 0 \\ 0.01969325 & 0 \\ 0.0106959 & 1 \\ 0.01069589 & 1 \end{bmatrix}$$

$$RMS := \text{READPRN}\left(\text{“RMS.prn”}\right)$$

RMS History:

Number of iterations:

$$Iterations := \text{rows}\left(RMS\right) - 1 \qquad RMS = \begin{bmatrix} 0 & 0 \\ 0.02 & 0 \\ 0.011 & 1 \\ 0.011 & 1 \end{bmatrix}$$

$$Iterations = 3$$

6. Write the corrected state vector to disk and convert to conic elements.

$$\text{WRITEPRN}\left(\text{“STATE.prn”}, \text{stack}\left(\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}, \begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \cdot K\right)\right) = \begin{bmatrix} 32.64663276 \\ 15.95530778 \\ 8.82601938 \\ -0.17401969 \\ -0.08201306 \\ -0.04487175 \end{bmatrix}$$

Compute and display the conic elements by calling function **PVCO** to transform position and velocity to conic elements.

$$r_1 := \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \qquad v_1 := \begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \cdot K \qquad \frac{v_1}{K} = \begin{bmatrix} -2.34001702 \\ -1.10281747 \\ -0.60338379 \end{bmatrix}$$

PVCO needs position in E.R. But here is position in units of km:

$$r_1 \cdot a_e = \begin{bmatrix} 208224.69631 \\ 101765.138913 \\ 56293.560761 \end{bmatrix}$$

PVCO needs velocity in E.R./min. But here is velocity in units of km/sec:

$$v_1 \cdot \frac{a_e}{60} = \begin{bmatrix} -18.49869036 \\ -8.71817543 \\ -4.76996949 \end{bmatrix}$$

$$rmag1 := \sqrt{r_1 \cdot r_1} \qquad rmag1 = 37.39349001$$ E.R.

PVCO also invokes function SCAL1, which we define now.

$$SCAL1(K, \alpha, q, e, v) := \begin{array}{l} \text{if } \alpha > 0 \\ \quad E \leftarrow v - 2 \cdot \operatorname{atan}\left(\dfrac{e \cdot \sin(v)}{1 + \sqrt{1 - e^2} + e \cdot \cos(v)}\right) \\ \quad s \leftarrow \dfrac{E}{\sqrt{\alpha}} \\ \text{else} \\ \quad w \leftarrow \dfrac{1}{K} \cdot \sqrt{\dfrac{q}{1+e}} \cdot \tan\left(\dfrac{v}{2}\right) \\ \quad \text{if } \alpha = 0 \\ \quad\quad s \leftarrow 2 \cdot w \\ \quad \text{else} \\ \quad\quad E \leftarrow 2 \cdot \operatorname{atanh}\left(\sqrt{-\alpha} \cdot w\right) \\ \quad\quad s \leftarrow \dfrac{E}{\sqrt{-\alpha}} \\ s \end{array}$$

Finally, now, we define function **PVCO**.

(Note that in **PVCO**, as defined in this document, the subscripts of the **P**, **Q**, and **W** vectors range from 1 through 3 rather than from 0 through 2. Also, the subscripts of **c** range from 1 through 4 rather than from 0 through 3.)

$$
PVCO(K, r, v) := \left|
\begin{array}{l}
rmag \leftarrow \sqrt{r \cdot r} \\
h \leftarrow r \times v \\
hmag \leftarrow \sqrt{h \cdot h} \\
W \leftarrow \dfrac{h}{hmag} \\
E \leftarrow \dfrac{v \cdot v}{2} - \dfrac{K^2}{rmag} \\
\alpha \leftarrow -2 \cdot E \\
p \leftarrow \dfrac{hmag^2}{K^2} \\
e \leftarrow \sqrt{1.0 - \alpha \cdot p \cdot K^{-2}} \\
q \leftarrow \dfrac{p}{1+e} \\
U \leftarrow \dfrac{r}{rmag} \\
V \leftarrow W \times U \\
\upsilon \leftarrow \text{angle}\left(\dfrac{hmag}{K^2} \cdot v \cdot V - 1.0, \dfrac{hmag}{K^2} \cdot v \cdot U\right) \\
P \leftarrow \cos(\upsilon) \cdot U - \sin(\upsilon) \cdot V \\
Q \leftarrow \sin(\upsilon) \cdot U + \cos(\upsilon) \cdot V \\
i \leftarrow \text{acos}\left(W_3\right) \\
\Omega \leftarrow \text{angle}\left(-W_2, W_1\right) \\
\omega \leftarrow \text{angle}\left(Q_3, P_3\right) \\
s \leftarrow SCAL1(K, \alpha, q, e, \upsilon) \\
c \leftarrow C\left(\alpha \cdot s^2\right) \\
\Delta t \leftarrow q \cdot s + K^2 \cdot e \cdot s^3 \cdot c_4 \\
\begin{bmatrix} q \\ e \\ i \cdot DegPerRad \\ \Omega \cdot DegPerRad \\ \omega \cdot DegPerRad \\ \Delta t \end{bmatrix}
\end{array}
\right.
$$

We now invoke **PVCO** and place its output into array **CONIC**.

$$CONIC := PVCO\left(K, r_1, v_1\right)$$

$$CONIC = \begin{bmatrix} 0.49411352 \\ 4.46001649 \\ 35.78644264 \\ 6.35496413 \\ 125.81289484 \\ -187.25342707 \end{bmatrix}$$

We should note that the position vector input to **PVCO** must have units of E.R. and the velocity vector must have units of E.R. per minute. We summarize the weighted batch least squares orbital solution as follows.

$\left(CONIC_1 - 1\right) \cdot a_e = -3226.61327$	Perigee height in km, relative to spherical Earth figure.
$CONIC_2 = 4.46001649$	Path eccentricity.
$CONIC_3 = 35.78644$	Path inclination, in degrees.
$CONIC_4 = 6.35496$	Right ascension of ascending node, in degrees.
$CONIC_5 = 125.81289$	Argument of perigee, in degrees.
$CONIC_6 = -187.25343$	Time of flight from perigee to epoch, in minutes.

First five quantities agree well with Visual Studio C++ batch DC program bd4c_mpc.exe, except for the last quantity. But bd4c_mpc.exe places epoch at last observation, whereas this GDC worksheet places epoch at first observation.

Time difference between first and last observation is 0.063751 days, i.e., 91.80144 minutes*. Adding this quantity to -187.25343, we get -95.45199 minutes, so "minutes to perigee," i.e., "time of flight from epoch to perigee" is 95.45199 minutes, which agrees well with the bd4c_mpc.exe batch DC result in output file bdc_sum.txt.

*See bd4c_mpc_2010.exe input file mpc_obs.txt, or Gd1 worksheet input file 2024 UQ Obs.txt.

We have the height of perigee above a spherical Earth figure, but for a closest approach determination, it would be more accurate to have the actual height of the object above its subpoint on a oblate spheroidal Earth at the instant of perigee. We calculate this now.

$$f := \frac{1}{298.26}$$ Earth's polar vs. equatorial flattening factor.

$$e_e := \sqrt{2 \cdot f - f^2}$$ Eccentricity of Earth's meridional reference ellipse.

We define function **GRT**, which inputs space object's position vector and outputs the geodetic latitude of the subsatellite point (subpoint), and the object's height above the subpoint.

$$\begin{aligned}
GRT(r) := \;& rmag \leftarrow \sqrt{r \cdot r} \\
& \delta \leftarrow \operatorname{asin}\left(\frac{r_3}{rmag}\right) \\
& \phi_c \leftarrow \delta \\
& \text{for } j \in 1 \,..\, 4 \\
& \quad r_s \leftarrow \frac{\sqrt{1 - {e_e}^2}}{\sqrt{1 - \left(e_e \cdot \cos\left(\phi_c\right)\right)^2}} \\
& \quad \phi_s \leftarrow \operatorname{atan}\left(\frac{\tan\left(\phi_c\right)}{1 - {e_e}^2}\right) \\
& \quad H_s \leftarrow \sqrt{rmag^2 - \left(r_s \cdot \sin\left(\phi_s - \phi_c\right)\right)^2} - r_s \cdot \cos\left(\phi_s - \phi_c\right) \\
& \quad \phi_c \leftarrow \delta - \operatorname{asin}\left(\frac{H_s \cdot \sin\left(\phi_s - \phi_c\right)}{rmag}\right) \\
& \begin{bmatrix} \phi_s \\ H_s \end{bmatrix}
\end{aligned}$$

$$\Delta t := -CONIC_6 \qquad M := FG\left(K, r_1, \frac{v_1}{K}, \Delta t\right)$$

$$f_r := M_{1,4} \qquad g_r := M_{2,4}$$

$$r := f_r \cdot r_1 + g_r \cdot \frac{v_1}{K} \qquad LatHt := GRT(r)$$

Geodetic latitude, ϕ_s, and height above spheroid, H_s, at time of perigee passage:

$$LatHt_1 \cdot DegPerRad = 28.63437$$ (degrees)

$$LatHt_2 \cdot a_e = -3221.75991$$ (km)

7. Repeat Steps 1-6, by clicking on "Calculate Worksheet", until convergence is obtained.

REFERENCES

[1] Mansfield, Roger L., "Algorithms for Reducing Radar Observations of a Hyperbolic Near Earth Flyby," *Journal of the Astronautical Sciences* (April-June 1993), pp. 249-259.

[2] Mansfield, Roger L., "Tracking Data Reduction for the Geotail, Mars Observer, and Galileo Missions," *Eleventh Space Surveillance Workshop,* MIT Lincoln Laboratory (March 31, 1993).

[Relevant material was distributed in a 19-page presentation handout, separately from the published conference proceedings, and was distributed only to attendees of the author's actual presentation. The published conference proceedings contain a six-page extended abstract of the presentation only (no equations).]

[3] Danby, J.M.A., *Fundamentals of Celestial Mechanics,* Willmann-Bell (2nd Ed. 1988), Appendix I.

[4] Goodyear, William H., "Completely General Closed-Form Solution for the Coordinates and Partial Derivatives of the Two-Body Problem," *Astronomical Journal,* Vol. 70, No. 3 (1965), pp. 189-192.

[5] Goodyear, William H., "A General Method for the Computation of Cartesian Coordinates and Partial Derivatives of the Two-Body Problem," NASA Report CR-522 (1966).

[This report, 97 pages in length, was prepared under Contract No. NAS 5-9762 by IBM for NASA's Goddard Space Flight Center, Greenbelt, Maryland.]

[6] Shepperd, Stanley W., "Universal Keplerian State Transition Matrix," *Celestial Mechanics,* Vol. 35, No. 2 (February 1985), pp. 129-144.

[7] Battin, Richard H., *An Introduction to the Mathematics and Methods of Astrodynamics,* AIAA Education Series (1987), Section 9.5.

[8] Der, Gim J., "An Elegant State Transition Matrix," *Journal of the Astronautical Sciences* (October - December 1997), pp. 371-390.

[9] Mansfield, Roger L. *Topics in Astrodynamics,* Astronomical Data Service, Colorado Springs, Colorado, USA (2003). See http://astrotopics.astroger.com. Chapter 14 addresses universal variables (uniform path mechanics, or UPM). Chapter 15 addresses batch least squares differential correction. Chapters 1-13 build a foundation for understanding the next two chapters, Chapters 14 and 15.

[10] Gaussian constant for Earth with units of E.R.$^{3/2}$/min. Based here on updated WGS-84 value of GM = $3.986004415 * 10^8$ m^3 / sec^2 . See David A . Vallado, *Fundamentals of Astrodynamics and Applications,* 5th Edition (2022), Microcosm Press, Torrance, CA USA, p. 132.