HERGET'S METHOD ITERATION WORKSHEET

Roger L. Mansfield, June 21, 1998 http://astroger.com

(Updated to PTC's Mathcad Prime 10.0 on 2024 July 20)

This worksheet implements Herget's method of preliminary orbit determination for comets and minor planets. See Herget [1] for the original journal article, Danby [2] for a more detailed exposition of the original algorithm, Mansfield [3] for the extended Gauss and uniform path propagation improvements incorporated herein into Herget's method, Gray [4] for a Windows application FIND_ORB that includes Herget's method, and Burtz [5] for details of the collection and original analysis of astrometric observations of the asteroid 1035 Amata, which were selected for use in this worksheet, as specified in the Herget's Method Initiation worksheet, HM1.

The author is particularly indebted to Air Force Second Lieutenant Dan C. Burtz for sharing his astrometric data, as taken using the equipment of the U.S. Air Force Academy Observatory during January-March 1998, and to Bill J. Gray, for calling to the author's attention the power of Herget's method for performing preliminary orbit determination (by making FIND_ORB available for downloading from the Project Pluto website, as mentioned by Stuart Goldman in the "Software Showcase,"*Sky & Telescope,* June 1998).

First we define constants and set the Mathcad worksheet ORIGIN to 1 so that subscripts start at unity rather than at zero.

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

$$SecPerDeg := 3600.0 \qquad SecPerRad := DegPerRad \cdot SecPerDeg$$

1. Retrieve values of n, ρ_1 and ρ_n, **JDT**, **L**, **A**, **D**, and **R** as previously specified by calculating the Herget's Method Initiation and Test Case Specification worksheet, HM1.

$$V := \mathrm{READPRN}(\text{“RHOVALS.prn”})$$

$n := V_1$ Number of observations.

$\rho_1 := V_2$ Estimate of geocentric distance at first observation.

$\rho_n := V_3$ Estimate of geocentric distance at n-th observation.

$\rho_1 = 2.67671542$ $\rho_n = 3.43659007$

$n = 5$

$$JDT := \text{READPRN}(\text{“TFILE.prn”}) \qquad JDT = \begin{bmatrix} 2450834.74164 \\ 2450840.7159 \\ 2450841.77069 \\ 2450857.56861 \\ 2450885.59222 \end{bmatrix}$$

$$L := \text{READPRN}(\text{“LFILE.prn”})$$

$$L = \begin{bmatrix} 0.41192221 & 0.41261493 & 0.41242816 & 0.39884919 & 0.33120336 \\ 0.61555733 & 0.62202611 & 0.6233099 & 0.64678145 & 0.69889469 \\ 0.67186998 & 0.66545656 & 0.66437021 & 0.65007159 & 0.63391684 \end{bmatrix}$$

$$A := \text{READPRN}(\text{“AFILE.prn”})$$

$$A = \begin{bmatrix} -0.83108276 & -0.83332748 & -0.83396689 & -0.8511705 & -0.9036638 \\ 0.55614876 & 0.55277963 & 0.55181448 & 0.52488931 & 0.42824261 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D := \text{READPRN}(\text{“DFILE.prn”})$$

$$D = \begin{bmatrix} -0.37365966 & -0.36785083 & -0.3666091 & -0.34121563 & -0.2714702 \\ -0.55837956 & -0.55454324 & -0.55406276 & -0.55332176 & -0.5728477 \\ 0.74066911 & 0.74643658 & 0.74740366 & 0.75987296 & 0.77340122 \end{bmatrix}$$

$$R := \text{READPRN}(\text{“RFILE.prn”})$$

$$R = \begin{bmatrix} 0.5070362 & 0.59386619 & 0.60855178 & 0.80048881 & 0.98495772 \\ -0.77385376 & -0.72072768 & -0.71049291 & -0.5302041 & -0.12172387 \\ -0.33552418 & -0.31249068 & -0.308055 & -0.2298888 & -0.05278695 \end{bmatrix}$$

2. Calculate $\mathbf{r}_1$ and $\mathbf{r}_n$ using the current estimates of ρ_1 and ρ_n.

$$r^{\langle 1 \rangle} := \rho_1 \cdot L^{\langle 1 \rangle} - R^{\langle 1 \rangle} \qquad r^{\langle n \rangle} := \rho_n \cdot L^{\langle n \rangle} - R^{\langle n \rangle}$$

Use Extended Gauss method (Ref. 3) to calculate $\mathbf{v}_1$. To do this, we will need function **C** to calculate the first four c-functions and the velocity-calculating function **VELO**, as needed by function **TWOPOE**. **TWOPOE** implements the extended Gauss method.

$$C(x) := \left\| \begin{array}{l} N \leftarrow 0 \\ \text{while } |x| \geq 0.1 \\ \quad \left\| \begin{array}{l} x \leftarrow \frac{x}{4} \\ N \leftarrow N + 1 \end{array} \right. \\ c_3 \leftarrow \frac{\left(1 - \frac{x}{20} \cdot \left(1 - \frac{x}{42} \cdot \left(1 - \frac{x}{72} \cdot \left(1 - \frac{x}{110} \cdot \left(1 - \frac{x}{156} \cdot \left(1 - \frac{x}{210}\right)\right)\right)\right)\right)\right)}{6} \\ c_2 \leftarrow \frac{\left(1 - \frac{x}{12} \cdot \left(1 - \frac{x}{30} \cdot \left(1 - \frac{x}{56} \cdot \left(1 - \frac{x}{090} \cdot \left(1 - \frac{x}{132} \cdot \left(1 - \frac{x}{182}\right)\right)\right)\right)\right)\right)}{2} \\ c_1 \leftarrow 1 - c_3 \cdot x \\ c_0 \leftarrow 1 - c_2 \cdot x \\ \text{while } N > 0 \\ \quad \left\| \begin{array}{l} N \leftarrow N - 1 \\ c_3 \leftarrow \frac{(c_1 \cdot c_2 + c_3)}{4} \\ c_2 \leftarrow \frac{c_1 \cdot c_1}{2} \\ c_1 \leftarrow c_1 \cdot c_0 \\ c_0 \leftarrow 2 \cdot c_0 \cdot c_0 - 1 \end{array} \right. \\ \left[c_0 \;\; c_1 \;\; c_2 \;\; c_3 \right]^{\mathrm{T}} \end{array} \right.$$

$$VELO(K, \Delta t, rmag_1, r_1, r_2, y, Arg) := \left\| \begin{array}{l} c \leftarrow C(Arg) \\ s3 \leftarrow \frac{K \cdot \Delta t \cdot \left(1 - \frac{1}{y}\right)}{c_4} \\ s2 \leftarrow s3^{\frac{2}{3}} \\ f \leftarrow 1 - \frac{s2 \cdot c_3}{rmag_1} \\ g \leftarrow \frac{K \cdot \Delta t}{y} \\ \left(\frac{1}{g}\right) \cdot r_2 - \left(\frac{f}{g}\right) \cdot r_1 \end{array} \right.$$

$$
\begin{aligned}
&TWOPOE\left(K, \Delta t, r_1, r_2\right) := \\
&\quad rmag_1 \leftarrow \sqrt{r_1 \cdot r_1} \\
&\quad rmag_2 \leftarrow \sqrt{r_2 \cdot r_2} \\
&\quad cos\Delta v \leftarrow \sqrt{\frac{1 + \frac{r_2 \cdot r_1}{rmag_2 \cdot rmag_1}}{2}} \\
&\quad \lambda \leftarrow \frac{rmag_2 + rmag_1}{4 \cdot \sqrt{rmag_2 \cdot rmag_1} \cdot cos\Delta v} - \frac{1}{2} \\
&\quad m \leftarrow \frac{K^2 \cdot (\Delta t)^2}{\left(2 \cdot \sqrt{rmag_2 \cdot rmag_1} \cdot cos\Delta v\right)^3} \\
&\quad y \leftarrow 0 \\
&\quad y_{new} \leftarrow 1 \\
&\quad \text{while } \left|y - y_{new}\right| \geq 0.00000001 \\
&\qquad y \leftarrow y_{new} \\
&\qquad x \leftarrow \frac{m}{y^2} - \lambda \\
&\qquad \text{if } x \geq 0 \\
&\qquad\quad z \leftarrow 4 \cdot \operatorname{asin}\left(\sqrt{x}\right) \\
&\qquad\quad Arg \leftarrow z^2 \\
&\qquad \text{else} \\
&\qquad\quad z \leftarrow 4 \cdot \operatorname{asin}\left(\sqrt{-x}\right) \\
&\qquad\quad Arg \leftarrow -z^2 \\
&\qquad c \leftarrow C\left(Arg\right) \\
&\qquad d \leftarrow C\left(\frac{Arg}{4}\right) \\
&\qquad X \leftarrow \frac{8 \cdot c_4}{\left(d_2\right)^3} \\
&\qquad y_{new} \leftarrow 1 + X \cdot (\lambda + x) \\
&\quad v \leftarrow VELO\left(K, \Delta t, rmag_1, r_1, r_2, y_{new}, Arg\right) \\
&\quad \text{augment}\left(r_1, v \cdot K\right)
\end{aligned}
$$

$k := 0.01720209895$ $\qquad$ $\mu := 1.0$

$K := k \cdot \sqrt{\mu}$ $\qquad$ $\Delta t := JDT_n - JDT_1$

$PV := TWOPOE\left(K, \Delta t, r^{\langle 1 \rangle}, r^{\langle n \rangle}\right)$

Compute and display the conic elements by calling function **PVCO** to transform $\mathbf{r}_1$ and $\mathbf{v}_1$ to conic elements.

First transform $\mathbf{r}_1$ and $\mathbf{v}_1$ from the HCI equatorial J2000.0 reference frame to the HCI ecliptic J2000.0 reference frame.

We will need the obliquity of the ecliptic, ε, at J2000.0, in order to transform the ECI ecliptic J2000.0 coordinates to ECI equatorial J2000.0 coordinates.

$$\varepsilon := \frac{23.4392911}{DegPerRad} \qquad M := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varepsilon) & -\sin(\varepsilon) \\ 0 & \sin(\varepsilon) & \cos(\varepsilon) \end{bmatrix}$$

$ECEQ(r) := M \cdot r$ (Transforms from ecliptic to equatorial.)

$EQEC(r) := M^{-1} \cdot r$ (Transforms from equatorial to ecliptic.)

$$r_1 := EQEC\left(PV^{\langle 1 \rangle}\right) \qquad r_1 = \begin{bmatrix} 0.59556231 \\ 3.07053443 \\ 0.99461396 \end{bmatrix}$$

$$v_1 := EQEC\left(PV^{\langle 2 \rangle}\right) \qquad v_1 = \begin{bmatrix} -0.0086049 \\ 0.00324807 \\ 0.00116843 \end{bmatrix}$$

$v^{\langle 1 \rangle} := PV^{\langle 2 \rangle}$

PVCO invokes function SCAL, which we define now.

(Note that in SCAL, as defined in this document, the subscripts of **c** range from 1 to 4 rather than from 0 to 3.)

$$SCAL(K, \alpha, q, e, v) :=$$
$$s \leftarrow \frac{2}{K} \cdot \sqrt{\frac{q}{1+e}} \cdot \tan\left(\frac{v}{2}\right)$$
$$\Delta s \leftarrow s$$
while $|\Delta s| \geq 0.00000001$
$$c \leftarrow C\left(\alpha \cdot s^2\right)$$
$$r \leftarrow \frac{q \cdot (1+e)}{1 + e \cdot \cos(v)}$$
$$f \leftarrow q + K^2 \cdot e \cdot s^2 \cdot c_3 - r$$
$$Df \leftarrow K^2 \cdot e \cdot s \cdot c_2$$
$$DDf \leftarrow K^2 \cdot e \cdot c_1$$
if $Df \geq 0$
$m \leftarrow 1$
else
$m \leftarrow -1$
$$\Delta s \leftarrow \frac{-5 \cdot f}{\left(Df + m \cdot \sqrt{\left|(4 \cdot Df)^2 - 20 \cdot f \cdot DDf\right|}\right)}$$
$$s \leftarrow s + \Delta s$$
s

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.)

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

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

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

$$CONIC = \begin{bmatrix} 2.5002149 \\ 0.20273768 \\ 18.08743686 \\ 2.20984863 \\ 323.03350335 \\ 518.26174756 \end{bmatrix}$$

We should note that the position vector input to **PVCO** must have units of A.U. and the velocity vector must have units of A.U. per day. We summarize the Herget's method preliminary orbital solution as follows.

$CONIC_1 = 2.5002149$ Perihelion distance in A.U.

$CONIC_2 = 0.20273768$ Path eccentricity.

$CONIC_3 = 18.08743686$ Path inclination, in degrees.

$CONIC_4 = 2.20984863$ Celestial longitude of ascending node, in degrees.

$CONIC_5 = 323.03350335$ Argument of perihelion, in degrees.

$CONIC_6 = 518.26174756$ Time of flight from perihelion to epoch, in days.

$$a := \frac{CONIC_1}{1 - CONIC_2} \qquad a = 3.13600033 \quad \text{A.U.}$$

$$n_c := K \cdot a^{\frac{-3}{2}} \cdot DegPerRad \qquad n_c = 0.177476119 \quad \text{deg/day}$$

$$M := n_c \cdot CONIC_6 \qquad M = 91.97908382 \quad \text{degrees}$$

3. Use f and g functions of Stumpff's c-functions (Ref. 3) to calculate positions $\mathbf{r}_2$ though $\mathbf{r}_{n-1}$.

To do this, we first define function UKEP to solve the uniform Kepler equation. This function will be invoked by function **UPM**, defined next.

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

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

$\Delta s \leftarrow s$

while $|\Delta s| \geq 0.00000001$

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

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

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

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

if $DF \geq 0$

$m \leftarrow 1$

else

$m \leftarrow -1$

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

$s \leftarrow s + \Delta s$

s

Function **UPM** implements uniform path mechanics. Given time Δt since epoch, it calculates position and velocity using f and g functions of Stumpff's c-functions. (Note that here, and in function UKEP, above, the subscripts of **c** range from 1 to 4 rather than from 0 to 3, since the Mathcad ORIGIN = 1.)

$$UPM(K, r_o, v_o, \Delta t) :=$$

$v_o \leftarrow \dfrac{v_o}{K}$

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

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

$\sigma_o \leftarrow r_o \cdot v_o$

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

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

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

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

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

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

$f_{dot} \leftarrow \dfrac{-s \cdot c_2}{rmag \cdot rmag_o}$

$g_{dot} \leftarrow 1 - \dfrac{s^2 \cdot c_3}{rmag}$

augment $\left(\left(f \cdot r_o + g \cdot v_o\right), K \cdot \left(f_{dot} \cdot r_o + g_{dot} \cdot v_o\right)\right)$

We still need a driver function for **UPM**. That function is performed by **UPMF**, defined as follows.

$$UPMF\left(K, r_o, v_o, JDT, n\right) := \left\| \begin{array}{l} \text{for } i \in 2 .. n-1 \\ \quad \left\| \begin{array}{l} \Delta t \leftarrow JDT_i - JDT_1 \\ PV \leftarrow UPM\left(K, r_o, v_o, \Delta t\right) \\ M^{\langle i-1 \rangle} \leftarrow PV^{\langle 1 \rangle} \end{array} \right. \\ M \end{array} \right\|$$

We invoke **UPMF** now to obtain **POS**, a 3-by-(n-2) matrix of position vectors $\mathbf{r}_2$, ..., $\mathbf{r}_{n-1}$.

$$POS := UPMF\left(K, r^{\langle 1 \rangle}, v^{\langle 1 \rangle}, JDT, n\right)$$

4. Calculate the residual functions $\mathbf{P}_1$ through $\mathbf{P}_{n-2}$ and $\mathbf{Q}_1$ through $\mathbf{Q}_{n-2}$.

$$RES\left(POS, R, A, n\right) := \left\| \begin{array}{l} \text{for } i \in 1 .. n-2 \\ \quad \left\| P_i \leftarrow \left\langle POS^{\langle i \rangle} + R^{\langle i+1 \rangle} \right\rangle \cdot A^{\langle i+1 \rangle} \right. \\ P \end{array} \right\|$$

$$P := RES\left(POS, R, A, n\right) \qquad Q := RES\left(POS, R, D, n\right)$$

Display the **P** and **Q** residuals in A.U.

$$\text{augment}\left(P, Q\right) = \begin{bmatrix} 0.00000122 & 0.00000374 \\ 0.00000222 & -0.00000499 \\ -0.00000147 & 0.00000023 \end{bmatrix}$$

Compute the RMS for this iteration in arc-seconds. To do this, we compute the geocentric distances, ρ_i, and divide the residuals P_i and Q_i by ρ_i for i = 1, ... , n-2, in order to convert from A.U. to radians. Finally, we multiply by the number of arc-seconds in one radian.

$$\rho := \left\| \begin{array}{l} \text{for } i \in 1 .. n-2 \\ \quad \rho_i \leftarrow \sqrt{\left(POS^{\langle i \rangle} + R^{\langle i+1 \rangle}\right) \cdot \left(POS^{\langle i \rangle} + R^{\langle i+1 \rangle}\right)} \\ \rho \end{array} \right\|$$

$$RMS := \sqrt{\frac{\sum_{i=1}^{n-2} \frac{\left(P_i \cdot P_i + Q_i \cdot Q_i\right)}{\left(\rho_i\right)^2}}{2 \cdot n - 4}} \cdot SecPerRad \qquad RMS = 0.209$$

$$\text{APPENDPRN}\left(\text{“RMS.prn”}, \left[RMS \;\; 0\right]\right) = \begin{bmatrix} 0 & 0 \\ 4240.04615587 & 0 \\ 6974.44972535 & 0 \\ 2751.9023691 & 0 \\ 1319.59863123 & 0 \\ 41.27370122 & 0 \\ 0.22285787 & 0 \\ 0.20908032 & 0 \\ 0.2090803 & 0 \end{bmatrix}$$

$$RMS := \text{READPRN}\left(\text{“RMS.prn”}\right)^{\langle 1 \rangle}$$

RMS History:

$$RMS = \begin{bmatrix} 0 \\ 4240.046 \\ 6974.45 \\ 2751.902 \\ 1319.599 \\ 41.274 \\ 0.223 \\ 0.209 \\ 0.209 \end{bmatrix}$$

Number of iterations:

$$Iterations := \text{rows}\left(RMS\right) - 1$$

$$Iterations = 8$$

5. Compute numerically the partial derivatives **dP1**$_i$, **dQ1**$_i$, **dPn**$_i$, **dQn**$_i$ for i=1, ..., n-2.

$\delta := 0.001$ (Set variation parameter for numerical partials.)

First compute numerically the partials with respect to ρ_1.

$$\rho_{1p} := \rho_1 + \delta$$

$$r^{\langle 1 \rangle} := \rho_{1p} \cdot L^{\langle 1 \rangle} - R^{\langle 1 \rangle} \qquad r^{\langle n \rangle} := \rho_n \cdot L^{\langle n \rangle} - R^{\langle n \rangle}$$

$$v^{\langle 1\rangle} := TWOPOE\left(K, JDT_n - JDT_1, r^{\langle 1\rangle}, r^{\langle n\rangle}\right)^{\langle 2\rangle}$$

$$POSp := UPMF\left(K, r^{\langle 1\rangle}, v^{\langle 1\rangle}, JDT, n\right)$$

$$Pp := RES\left(POSp, R, A, n\right) \qquad Qp := RES\left(POSp, R, D, n\right)$$

$$dP1 := \frac{Pp - P}{\delta} \qquad dQ1 := \frac{Qp - Q}{\delta}$$

Now compute numerically the partials with respect to ρ_n.

$$\rho_{np} := \rho_n + \delta$$

$$r^{\langle 1\rangle} := \rho_1 \cdot L^{\langle 1\rangle} - R^{\langle 1\rangle} \qquad r^{\langle n\rangle} := \rho_{np} \cdot L^{\langle n\rangle} - R^{\langle n\rangle}$$

$$v^{\langle 1\rangle} := TWOPOE\left(K, JDT_n - JDT_1, r^{\langle 1\rangle}, r^{\langle n\rangle}\right)^{\langle 2\rangle}$$

$$POSp := UPMF\left(K, r^{\langle 1\rangle}, v^{\langle 1\rangle}, JDT, n\right)$$

$$Pp := RES\left(POSp, R, A, n\right) \qquad Qp := RES\left(POSp, R, D, n\right)$$

$$dPn := \frac{Pp - P}{\delta} \qquad dQn := \frac{Qp - Q}{\delta}$$

6. Compute the corrections to ρ_1 and ρ_n by a least squares fit. There are 2n-4 equations (n-2 equations in **P** and n-2 equations in **Q**) in two unknowns, $\Delta\rho_1$ and $\Delta\rho_n$, as follows:

P + **dP1** $\Delta\rho_1$ + **dPn** $\Delta\rho_n$ = **0** (n-2 equations)

Q + **dQ1** $\Delta\rho_1$ + **dQn** $\Delta\rho_n$ = **0** (n-2 equations)

We use Mathcad's Minerr function (see Mathcad PLUS 6 Manual, p. 355) to find the solution:

$$SSR\left(\Delta\rho_1, \Delta\rho_n\right) := \sum_{i=1}^{n-2}\left(\left(P_i + dP1_i \cdot \Delta\rho_1 + dPn_i \cdot \Delta\rho_n\right)^2 + \left(Q_i + dQ1_i \cdot \Delta\rho_1 + dQn_i \cdot \Delta\rho_n\right)^2\right)$$

$\Delta\rho_1 := \rho_1$ $\Delta\rho_n := \rho_n$

$SSR(\Delta\rho_1, \Delta\rho_n) = 0.01016244$

Given

Constraints

$1 = 1$

$SSR(\Delta\rho_1, \Delta\rho_n) = 0$

Solver

$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_n \end{bmatrix} := \mathbf{minerr}(\Delta\rho_1, \Delta\rho_n)$

< This is the result after the Mathcad Prime 10.0 worksheet converter converts Minerr function setup in Mathcad 15 worksheet HMC.xmcd.

In Mathcad Prime 10.0, see Functions > Solving > minerr.

$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_n \end{bmatrix} = \begin{bmatrix} 2.79540536 \cdot 10^{-9} \\ 8.81318839 \cdot 10^{-9} \end{bmatrix}$

$\rho_1 := \rho_1 + \Delta\rho_1$

$\rho_n := \rho_n + \Delta\rho_n$

$\text{WRITEPRN}\left(\text{“RHOVALS.prn”}, \begin{bmatrix} n \\ \rho_1 \\ \rho_n \end{bmatrix}\right) = \begin{bmatrix} 5 \\ 2.67671542 \\ 3.43659008 \end{bmatrix}$

7. Repeat steps 1-6 by clicking on the Mathcad "Calculate Worksheet" command (from the Math menu), as many times as necessary to get Herget's method to converge. Usually convergence will be noted as having occurred when the last two RMS values in the **RMS** matrix of Step 4 are the same, to three significant figures, and are smaller than any other RMS value above them in the **RMS** matrix.

If the RMS does not trend downward after five or so iterations, click on the open HM1 (Herget's Method Initiation) worksheet, define new starting values of ρ_1 and ρ_n, and then click on the "Calculate Worksheet" command while the HM1 worksheet window is still active, before returning to this HMC (Herget's Method Iteration) worksheet window.

REFERENCES

[1] Herget, Paul, "Computation of Preliminary Orbits," *Astronomical Journal* , Vol. 70 No. 1 (February 1965), pp. 1-3.

Paul Herget (1908-1981), a native of Cincinnati, Ohio, was Director of the Cincinnati Observatory from 1943 to 1978. In 1947 the International Astronomical Union invited Herget to organize the Minor Planet Center (MPC) and become its first director. From 1947 to 1978 the MPC collected more than 170,000 precise positions of asteroids and published 4,358 Minor Planet Circulars while under Herget's directorship. More information about Herget's career as an astronomer can be found in *Physics Today,* January 1982, pp. 86, 87.

Herget is probably most well known among dynamical astronomers for his privately published and widely cited booklet,*The Computation of Orbits,* originally published in 1948 and reprinted in May 1962 (ix + 177 pages).

[2] Danby, J.M.A., *Fundamentals of Celestial Mechanics,* Willmann-Bell (2nd Ed. 1988), Sect. 7.4.

[3] 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.

[4] Gray, Bill J., "FIND_ORB," *Project Pluto,* (March 1997).

FIND_ORB is a program for computing orbits from observations of comets and minor planets; the user interface is Microsoft Windows-based. The program uses Herget's method for computing a preliminary orbit, and batch least squares for fitting the definitive orbit to the observations. The program can optionally account for perturbations due to the major planets and Earth's moon (the user specifies which planets are "perturbers" by checking boxes). FIND_ORB can solve for a forced parabolic fit in the case of a comet, and permits a user-specified epoch for easy comparison of FIND_ORB's solution with a corresponding Minor Planet Center solution. Browse http://www.projectpluto.com for further information.

Note: The Herget's method worksheets HMC and HM1 together yield results slightly different from those of FIND_ORB's Herget Step iteration, for the same test data. The reason is that HM1 uses a simplified model for Earth's orbital motion, i.e., procedural function **HGEO**, while FIND_ORB uses the more accurate VSOP model of Bretagnon and Francou.

[5] Burtz, Daniel C., "Asteroid Orbit Determination and Rotational Period Calculation with CCD Astronomy," *Master of Engineering Program Office,* CU-Colorado Springs (May 1998).