HERGET'S METHOD ITERATION WORKSHEET

Roger L. Mansfield, October 3, 1999
http://astroger.com

(Updated to PTC's Mathcad Prime 10.0 on 2024 July 30)

This worksheet implements Herget's method of preliminary orbit determination for comets and minor planets, with modifications as necessary to apply the method to artificial Earth satellite orbit determination. 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, and Gray [4] for details of the Cassini flyby test case as implemented in worksheet GH1.

This worksheet is set up to use angles-only (optical) observations only.

Note that the Herget's method initiation worksheet GH1 must be opened and calculated (click on "Calculate Worksheet" from the Mathcad Math menu) before this Herget's method iteration worksheet is opened and calculated. See Step 7, below, for further comments regarding iteration of this worksheet.

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

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

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

1. Retrieve values of n, ρ_1 and ρ_n, **t**, **L**, **A**, **D**, and **R** as previously specified by calculating the Herget's Method Initiation and Test Case Specification worksheet, GH1.

$V := \text{READPRN}(\text{"RHOVALS.prn"})$

$n := V_1$ $\quad$ Number of observations.

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

$\rho_n := V_3$

Estimate of topocentric distance at n-th observation.

$\rho_1 = 81.60966342$

$\rho_n = 353.57432256$

$n = 11$

$JDT := \text{READPRN}(\text{“TFILE.prn”})$

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

$$L^{\mathrm{T}} = \begin{bmatrix} 0.98660872 & -0.13694934 & -0.08858959 \\ 0.98658842 & -0.13709425 & -0.08859152 \\ 0.9865322 & -0.13750037 & -0.08858814 \\ 0.98640839 & -0.13840097 & -0.08856448 \\ 0.98636292 & -0.13873326 & -0.08855095 \\ 0.98632192 & -0.13903399 & -0.08853598 \\ 0.98709883 & -0.13410357 & -0.08747643 \\ 0.98709004 & -0.13417329 & -0.0874687 \\ 0.9870739 & -0.1343005 & -0.08745566 \\ 0.9870467 & -0.13451686 & -0.08743007 \\ 0.98703926 & -0.13457653 & -0.08742234 \end{bmatrix}$$

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

$$A^{\mathrm{T}} = \begin{bmatrix} 0.13748993 & 0.99050317 & 0 \\ 0.13763543 & 0.99048296 & 0 \\ 0.13804311 & 0.99042622 & 0 \\ 0.13894697 & 0.99029982 & 0 \\ 0.1392804 & 0.99025298 & 0 \\ 0.13958213 & 0.9902105 & 0 \\ 0.13461962 & 0.99089735 & 0 \\ 0.13468952 & 0.99088785 & 0 \\ 0.13481706 & 0.99087051 & 0 \\ 0.13503395 & 0.99084097 & 0 \\ 0.13509376 & 0.99083282 & 0 \end{bmatrix}$$

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

$$D^{T} = \begin{bmatrix} 0.08774827 & -0.01218018 & 0.99606821 \\ 0.08774839 & -0.01219333 & 0.99606804 \\ 0.08774001 & -0.01222898 & 0.99606834 \\ 0.08770538 & -0.01230577 & 0.99607045 \\ 0.08768785 & -0.01233341 & 0.99607165 \\ 0.08766926 & -0.01235804 & 0.99607298 \\ 0.08668016 & -0.01177604 & 0.99616659 \\ 0.08667167 & -0.01178112 & 0.99616727 \\ 0.08665724 & -0.01179052 & 0.99616841 \\ 0.08662929 & -0.01180603 & 0.99617066 \\ 0.08662092 & -0.01181021 & 0.99617134 \end{bmatrix}$$

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

$$R^{T} = \begin{bmatrix} -0.47869103 & 0.70847103 & 0.51709 \\ -0.50529837 & 0.68974623 & 0.51709 \\ -0.57126489 & 0.63618608 & 0.51709 \\ -0.68725003 & 0.50868821 & 0.51709 \\ -0.72178086 & 0.45837614 & 0.51709 \\ -0.74980543 & 0.41093567 & 0.51709 \\ -0.80434218 & -0.29001717 & 0.51709 \\ -0.79065068 & -0.32549625 & 0.51709 \\ -0.76283557 & -0.38621004 & 0.51709 \\ -0.69341114 & -0.50025723 & 0.51709 \\ -0.66709087 & -0.53485145 & 0.51709 \end{bmatrix}$$

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

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

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.

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

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

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

$k := 0.07436691613$ $\mu := 1.0$

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

$PV := TWOPOE\left(K, \Delta t \cdot 1440, r^{\langle 1 \rangle}, r^{\langle n \rangle}\right)$

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

$$r_1 := PV^{\langle 1\rangle} \qquad r_1 = \begin{bmatrix} 80.99549642 \\ -11.88486097 \\ -7.74685633 \end{bmatrix}$$

$$v_1 := PV^{\langle 2\rangle} \qquad v_1 = \begin{bmatrix} 0.14837261 \\ -0.01942194 \\ -0.01307913 \end{bmatrix}$$

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

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, α, q, e, υ) :=
    $s \leftarrow \frac{2}{K} \cdot \sqrt{\frac{q}{1+e}} \cdot \tan\left(\frac{\upsilon}{2}\right)$
    $\Delta s \leftarrow s$
    while $|\Delta s| \ge 0.00000001$
        $c \leftarrow C(\alpha \cdot s^2)$
        $r \leftarrow \frac{q \cdot (1+e)}{1 + e \cdot \cos(\upsilon)}$
        $f \leftarrow q + K^2 \cdot e \cdot s^2 \cdot c_3 - r$
        $Df \leftarrow K^2 \cdot e \cdot s \cdot c_2$
        $DDf \leftarrow K^2 \cdot e \cdot c_1$
        if $Df \ge 0$
            $m \leftarrow 1$
        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.)

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

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

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

Specify Earth's mean equatorial radius so that calculations of distance can be converted from E.R. to kilometers as needed:

$$CONIC = \begin{bmatrix} 1.18109631 \\ 5.78982915 \\ 25.36261482 \\ 3.16766493 \\ 248.32697447 \\ 542.81100347 \end{bmatrix}$$

$a_e := 6378.135$

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 Herget's method preliminary orbital solution as follows.

$\left(CONIC_1 - 1\right) \cdot a_e = 1155.05672042$	Perigee height in km, relative to spherical Earth figure.
$CONIC_2 = 5.78982915$	Path eccentricity.
$CONIC_3 = 25.36261482$	Path inclination, in degrees.
$CONIC_4 = 3.16766493$	Right ascension of ascending node, in degrees.
$CONIC_5 = 248.32697447$	Argument of perigee, in degrees.
$CONIC_6 = 542.81100347$	Time of flight from perigee to epoch, in minutes.

(For a discussion of how these results compare with the Cassini Navigation Team's results, see "Analysis of Results with 11 Cassini Observations" at the very end of this worksheet.)

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.

```
$UKEP(\tau, r_o, \sigma_o, \alpha) :=$
    $s \leftarrow \frac{\tau}{r_o}$
    $\Delta s \leftarrow s$
    while $|\Delta s| \geq 0.00000001$
        $c \leftarrow C(\alpha \cdot s^2)$
        $F \leftarrow r_o \cdot s \cdot c_2 + \sigma_o \cdot s^2 \cdot c_3 + s^3 \cdot c_4 - \tau$
        $DF \leftarrow r_o \cdot c_1 + \sigma_o \cdot s \cdot c_2 + s^2 \cdot c_3$
        $DDF \leftarrow \sigma_o \cdot c_1 + (1 - r_o \cdot \alpha) \cdot s \cdot c_2$
        if $DF \geq 0$
            $m \leftarrow 1$
        else
            $m \leftarrow -1$
        $\Delta s \leftarrow \frac{-5 \cdot F}{\left(DF + m \cdot \sqrt{\left|(4 \cdot DF)^2 - 20 \cdot F \cdot DDF\right|}\right)}$
        $s \leftarrow s + \Delta s$
    $s$
```

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, Δt) :=
    $v_o \leftarrow \frac{v_o}{K}$
    $\tau \leftarrow K \cdot \Delta t$
    $rmag_o \leftarrow \sqrt{r_o \cdot r_o}$
    $\sigma_o \leftarrow r_o \cdot v_o$
    $\alpha \leftarrow \frac{2}{rmag_o} - v_o \cdot v_o$
    $s \leftarrow UKEP(\tau, rmag_o, \sigma_o, \alpha)$
    $c \leftarrow C(\alpha \cdot s^2)$
    $f \leftarrow 1 - \frac{s^2 \cdot c_3}{rmag_o}$
    $g \leftarrow \tau - s^3 \cdot c_4$
    $rmag \leftarrow rmag_o \cdot c_1 + \sigma_o \cdot s \cdot c_2 + s^2 \cdot c_3$
    $f_{dot} \leftarrow \frac{-s \cdot c_2}{rmag \cdot rmag_o}$
    $g_{dot} \leftarrow 1 - \frac{s^2 \cdot c_3}{rmag}$
    $\text{augment}\left(\left(f \cdot r_o + g \cdot v_o\right), K \cdot \left(f_{dot} \cdot r_o + g_{dot} \cdot v_o\right)\right)$
```

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

```
UPMF(K, r_o, v_o, t, n) :=
    for $i \in 2 .. n-1$
        $\Delta t \leftarrow \left(JDT_i - JDT_1\right) \cdot 1440$
        $PV \leftarrow UPM(K, r_o, v_o, \Delta t)$
        $M^{\langle i-1 \rangle} \leftarrow PV^{\langle 1 \rangle}$
    $M$
```

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

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

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

```
RES(POS, R, A, n) :=  for i ∈ 1 .. n − 2
                          P_i ← (POS^⟨i⟩ + R^⟨i+1⟩) · A^⟨i+1⟩
                      P
```

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

Display the **P** and **Q** residuals in kilometers.

$$\text{augment}(P, Q) \cdot a_e = \begin{bmatrix} 0.20854653 & 0.25464657 \\ 1.11938479 & -1.22282167 \\ 0.08881714 & -3.08481388 \\ -0.12167222 & -3.3404625 \\ -0.8817138 & -3.7278967 \\ 3.39791652 & -1.67796347 \\ -7.69884727 & -4.42147931 \\ 0.12644588 & -6.46440586 \\ 1.53123709 & -3.74060366 \end{bmatrix}$$

Compute the RMS error for this iteration in kilometers.

$$RMS := \sqrt{\frac{\sum_{i=1}^{n-2} \left(P_i \cdot P_i + Q_i \cdot Q_i\right)}{2 \cdot n - 4}} \cdot a_e \qquad\qquad RMS = 3.245$$

$$\text{APPENDPRN}\left(\text{“RMS.prn”}, \begin{bmatrix} RMS & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & 0 \\ 261.62347948 & 0 \\ 3.42757947 & 0 \\ 3.24510918 & 0 \\ 3.24510918 & 0 \end{bmatrix}$$

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

RMS History: $$RMS = \begin{bmatrix} 0 \\ 261.623 \\ 3.428 \\ 3.245 \\ 3.245 \end{bmatrix}$$

Number of iterations:

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

$$Iterations = 4$$

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.

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

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

$$v^{\langle 1\rangle} := TWOPOE\left(K, \left(JDT_n - JDT_1\right) \cdot 1440, r^{\langle 1\rangle}, r^{\langle n\rangle}\right)^{\langle 2\rangle}$$

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

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

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

Now compute numerically the partials with respect to ρ_n.

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

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

$$v^{\langle 1\rangle} := TWOPOE\left(K, \left(JDT_n - JDT_1\right) \cdot 1440, r^{\langle 1\rangle}, r^{\langle n\rangle}\right)^{\langle 2\rangle}$$

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

$Pp := RES(POSp, R, A, n)$ $Qp := RES(POSp, R, D, n)$

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

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:

$\mathbf{P} + \mathbf{dP1}\,\Delta\rho_1 + \mathbf{dPn}\,\Delta\rho_n = \mathbf{0}$ (n-2 equations)

$\mathbf{Q} + \mathbf{dQ1}\,\Delta\rho_1 + \mathbf{dQn}\,\Delta\rho_n = \mathbf{0}$ (n-2 equations)

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

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

$\Delta\rho_1 := 0$ $\Delta\rho_n := 0$

$SSR(\Delta\rho_1, \Delta\rho_n) = 0.00000466$

Guess Values
Constraints

$1 = 1$

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

Solver

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

This is what Mathcad Prime 10 gives for its conversion of the Mathcad 15 Minerr setup.

$\begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_n \end{bmatrix} = \begin{bmatrix} 0.00000115 \\ 0.00000608 \end{bmatrix}$

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

$$\text{WRITEPRN}\left(\text{“RHOVALS.prn”}, \begin{bmatrix} n \\ \rho_1 \\ \rho_n \end{bmatrix} \right) = \begin{bmatrix} 11 \\ 81.60966457 \\ 353.57432864 \end{bmatrix}$$

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 GH1 (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 GH1 worksheet window is still active, before returning to this GHC (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), Section 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., "Astrometry for the Cassini Flyby, 18-20 August 1999," Project Pluto website, (http://www.projectpluto.com/cassini.htm).

[5] See "Cassini Completes Inner-Solar System Tour on Way to Saturn," *Aviation Week & Space Technology,* August 23, 1999, p. 46.

ANALYSIS OF RESULTS WITH 11 CASSINI OBSERVATIONS

(This analysis assumes that the GH1 worksheet was opened and "Calculate Worksheet" was clicked once, then the GHC worksheet was opened and "Calculate Worksheet" was clicked four times.)

NASA's JPL (see Ref. 5, above) reported a perigee distance of 1171 km (actually, 727 miles x 5280 feet/mile x 0.3048 meters/foot x 0.001 km/meter). So our GH1/GHC-derived answer, after four iterations (i.e., after four "Calculate Worksheet" clicks of GHC), is only 16 km lower than JPL's best estimate.

NASA's JPL also reported that perigee was on August 17 at 8:28 p.m. PDT. This works out to August 18 at 3:28 UTC. Now the epoch of our GH1/GHC solution is the time of the first observation, 1999 August 18.51899 TT. If we retain only the fractional days, convert **CONIC**$_6$ to fractional days and subtract, then multiply by 24, we get the number of hours. Subtracting three hours and multiplying by 60 gives the minutes (we must not forget to subtract TT-UT):

$$\left(0.51899-\frac{CONIC_6}{1440}\right)\cdot 24=3.40890994 \qquad \text{(hours)}$$

$$\left(\left(0.51899-\frac{CONIC_6}{1440}\right)\cdot 24-3\right)\cdot 60-\frac{64.184}{60}=23.4648632 \qquad \text{(minutes)}$$

We thus find that Cassini's Earth flyby perigee was at 3:23.5 UTC on August 18, about 4.5 minutes earlier than calculated by NASA. Not a bad result, given that (a) we used two-body mechanics with Herget's method, in itself only a two-parameter fit, and (b) we used only eleven observations, all taken when Cassini was between 80 and 350 Earth radii distant from the geocenter!