HERGET'S METHOD ITERATION WORKSHEET

Roger L. Mansfield, April 26, 2025
http://astroger.com/

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, and Mansfield [3] for the extended Gauss and uniform path propagation improvements incorporated herein into Herget's method.

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 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, **t**, **L**, **A**, **D**, and **R** as previously specified by calculating the Herget's Method Initiation and Test Case Specification worksheet, Gh1.

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

$n := V_1$ Number of observations.

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

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

$\rho_1 = 36.4977965 \qquad \rho_n = 18.27272293$

$n = 9$

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

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

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

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

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

See Gh1 worksheet for values of

Julian date array **JDT**,

L, **A**, **D**, triad,

and position vector **R**.

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

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

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

Use Extended Gauss method (Ref. 3) to calculate $\mathbf{v}_1$. To do this, 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.

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

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

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

$k := 0.07436684771154$ See Note below*. $\mu := 1.0$

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

$\Delta t := JDT_n - JDT_1$

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

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

$r_1 := PV^{\langle 1 \rangle}$

$$r_1 = \begin{bmatrix} 32.67401578 \\ 15.96846826 \\ 8.83305626 \end{bmatrix}$$

$v_1 := PV^{\langle 2 \rangle}$

$$v_1 = \begin{bmatrix} -0.17422571 \\ -0.08210892 \\ -0.0449227 \end{bmatrix}$$

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

$a_e := 6378.137$

Specify Earth's mean equatorial radius (E.R.) in km.

*Note: Gaussian Earth gravity constant k above has units of E.R.$^{3/2}$/min.

$$r_1 \cdot a_e = \begin{bmatrix} 208399.34897676 \\ 101849.07822108 \\ 56338.44293589 \end{bmatrix}$$

Display conversion of position from ER to km. This has not changed position *r* to km, because "=" here means "show me value of what is left of equal sign," whereas colon-equals ":=" means "assign number on right to variable on left."

$$v_1 \cdot \frac{a_e}{60} = \begin{bmatrix} -18.5205911 \\ -8.72836619 \\ -4.77538602 \end{bmatrix}$$

Display conversion of velocity from ER/min to km/sec. This has not changed velocity *v* to km/sec, because here again, "=" here means "show me value of what is on left of equal sign," whereas colon-equals ":=" means "assign number on right to variable on left."

PVCO invokes function SCAL1, which we define now.

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

Finally, now, 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.)

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

Now invoke **PVCO** and place its output into array **CONIC**.

$$CONIC := PVCO(K, r_1, v_1)$$

$$CONIC = \begin{bmatrix} 0.49490764 \\ 4.47383725 \\ 35.78888823 \\ 6.35728904 \\ 125.76970685 \\ -187.19534482 \end{bmatrix}$$

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 Herget's method preliminary orbital solution as follows.

$\left(CONIC_1 - 1\right) \cdot a_e = -3221.54825644$ Perigee height in km, relative to spherical Earth figure.

$CONIC_2 = 4.47383725$ Path eccentricity.

$CONIC_3 = 35.78888823$ Path inclination, in degrees.

$CONIC_4 = 6.35728904$ Right ascension of ascending node, in degrees.

$CONIC_5 = 125.76970685$ Argument of perigee, in degrees.

$CONIC_6 = -187.19534482$ Time of flight from perigee to epoch, in minutes.

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.

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

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.)

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

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

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

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

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

$$POS^{T} = \begin{bmatrix} 31.52488561 & 15.42690595 & 8.53676128 \\ 30.95051182 & 15.15621388 & 8.38866235 \\ 19.14082802 & 9.59025381 & 5.34341236 \\ 18.68576526 & 9.37576348 & 5.22605745 \\ 18.34461117 & 9.21496174 & 5.13807714 \\ 17.88917181 & 9.00029007 & 5.02062236 \\ 17.56805402 & 8.84892951 & 4.93780717 \end{bmatrix}$$

Note that **POS**-transpose is displayed instead of **POS**.

Displaying **POS** would force worksheet to go from page to draft format

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

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

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

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

$$\text{augment}(P, Q) \cdot a_e = \begin{bmatrix} -0.67208434 & -0.79647346 \\ 0.32713281 & -0.22388447 \\ -0.73869445 & 0.36888137 \\ 1.39788101 & 1.97215119 \\ -0.95768878 & -0.21425451 \\ -0.57968384 & 0.03948942 \\ -0.40639301 & -0.03694655 \end{bmatrix}$$

Compute the RMS error for this iteration in kilometers.

$$RMS := \sqrt{\frac{\sum_{i=1}^{n-2} \left(P_i \cdot P_i + Q_i \cdot Q_i\right)}{2 \cdot n - 4}} \cdot a_e \qquad RMS = 0.812$$

$$\text{APPENDPRN}(\text{“RMS.prn”}, [RMS \;\; 0]) = \begin{bmatrix} 0 & 0 \\ 266.122622 & 0 \\ 4.76877572 & 0 \\ 0.81220994 & 0 \\ 0.81220986 & 0 \end{bmatrix}$$

$RMS := \text{READPRN}(\text{"RMS.prn"})^{\langle 1 \rangle}$

RMS History:

Number of iterations:

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

$Iterations = 4$

$$RMS = \begin{bmatrix} 0 \\ 266.123 \\ 4.769 \\ 0.812 \\ 0.812 \end{bmatrix}$$

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.

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

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

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

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

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

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

$$dPn := \frac{Pp - P}{\delta}$$

$$dQn := \frac{Qp - Q}{\delta}$$

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:

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

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

These equations can be cast into the familiar matrix notation **Y = A X**, for which the least squares normal equations are $(\mathbf{A}^T\mathbf{A})\,\mathbf{X} = \mathbf{A}^T\mathbf{Y}$, and for which the solution is $\mathbf{X} = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\,\mathbf{Y}$. That is,

$Y := -\text{stack}(P, Q)$

$A := \text{augment}(\text{stack}(dP1, dQ1), \text{stack}(dPn, dQn))$

$$A^T \cdot A = \begin{bmatrix} 0.0000465 & -0.00002353 \\ -0.00002353 & 0.00003092 \end{bmatrix}$$

$X := (A^T \cdot A)^{-1} \cdot A^T \cdot Y$

$$X = \begin{bmatrix} -7.81304848 \cdot 10^{-11} \\ -1.31420472 \cdot 10^{-10} \end{bmatrix}$$

$\Delta\rho_1 := X_1$

$\Delta\rho_n := X_2$

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

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

$$\text{WRITEPRN}\left(\text{“RHOVALS.prn”}, \begin{bmatrix} n \\ \rho_1 \\ \rho_n \end{bmatrix}\right) = \begin{bmatrix} 9 \\ 36.4977965 \\ 18.27272293 \end{bmatrix}$$

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 book, *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., "Preliminary Determination of the Geocentric Earth Flyby Path of Asteroid 2012 DA14," AAS Paper 14-288, 24th AAS/AIAA Space Flight Mechanics Meeting, Santa Fe, New Mexico, 28 January 2014.