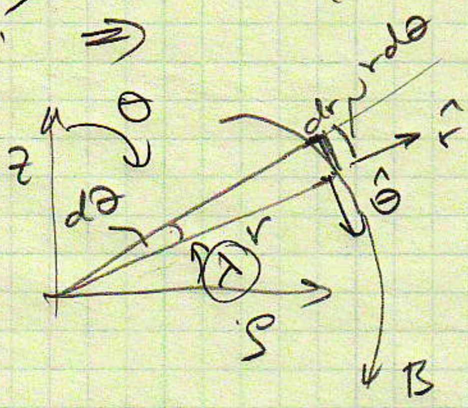
① $\bar{B} = B_0 \left(\frac{R_p}{r}\right)^3 (2\cos\theta\, \hat{r} + \sin\theta\, \hat{\theta})$

a) ⟹



$$\frac{dr}{r d\theta} = \frac{B_r}{B_\theta} = \frac{2\cos\theta}{\sin\theta}$$

$$\frac{dr}{r} = \frac{2\cos\theta}{\sin\theta} d\theta$$

$$\ln r = 2 \ln \sin\theta + \text{const}$$

<u>B.C.</u> $r = R$ @ $\theta = \pi/2$

$$r = R \sin^2\theta$$

or $\boxed{r = R\cos^2\lambda}$

$$R_c := \frac{(r^2 + r_\lambda^2)^{3/2}}{|r^2 + 2r_\lambda^2 - r r_{\lambda\lambda}|} \Longrightarrow \frac{R^3}{R^2 + 2R^2} = \underline{\underline{\frac{R}{3}}}$$

$r_\lambda = \frac{dr}{d\lambda} = -2R\cos\lambda\sin\lambda \Rightarrow$ $\lambda = 0$: $r_\lambda = 0$

$r_{\lambda\lambda} = \frac{d^2r}{d\lambda^2} = -2R(\cos^2\lambda - \sin^2\lambda) \Rightarrow -2R$

b) curvature and gradient drifts

energy $= w_\parallel + w_\perp$ @ $\lambda = 0$

$$\bar{V}_c = \frac{2w_\parallel}{qB}\left[\frac{\hat{B} \times \hat{R}_c}{R_c}\right]$$

$$\boxed{\bar{V}_c = \frac{6W_\parallel}{qBR}\hat{\phi}}$$

$$\bar{V}_G = \frac{W_\perp}{qB}\left[\frac{\hat{B}\times\bar{\nabla}B}{B}\right]$$

$$\bar{\nabla}B = \bar{\nabla}\left[B_o\left(\frac{R_p}{r}\right)^3\right] = -3B_o\frac{R_p^3}{r^4}\hat{r} = -3B\frac{1}{r}$$

$$\boxed{\bar{V}_G = \frac{3W_\perp}{qB}\frac{1}{R}\hat{\phi}}$$

c) pitch angle $\pi/2 \Rightarrow W_{\parallel} = 0$

$$\bar{V}_{Drift} = \bar{V}_G$$

Analy $W_\perp = 10^3$ eV $\Rightarrow 1.6\times10^{-16}$ J (Nm)
(pick an energy)

$R = R_{synch} = 6.68\, R_E =$

$R_E = 6378$ km $\approx 6.4\times10^6$ m

$q = 1.6\times10^{-19}$ C

$B_o = 30\times10^{-6}$ T $\Rightarrow B = \frac{B_o}{(6.68)^3} \simeq 10^{-7}$ Tesla

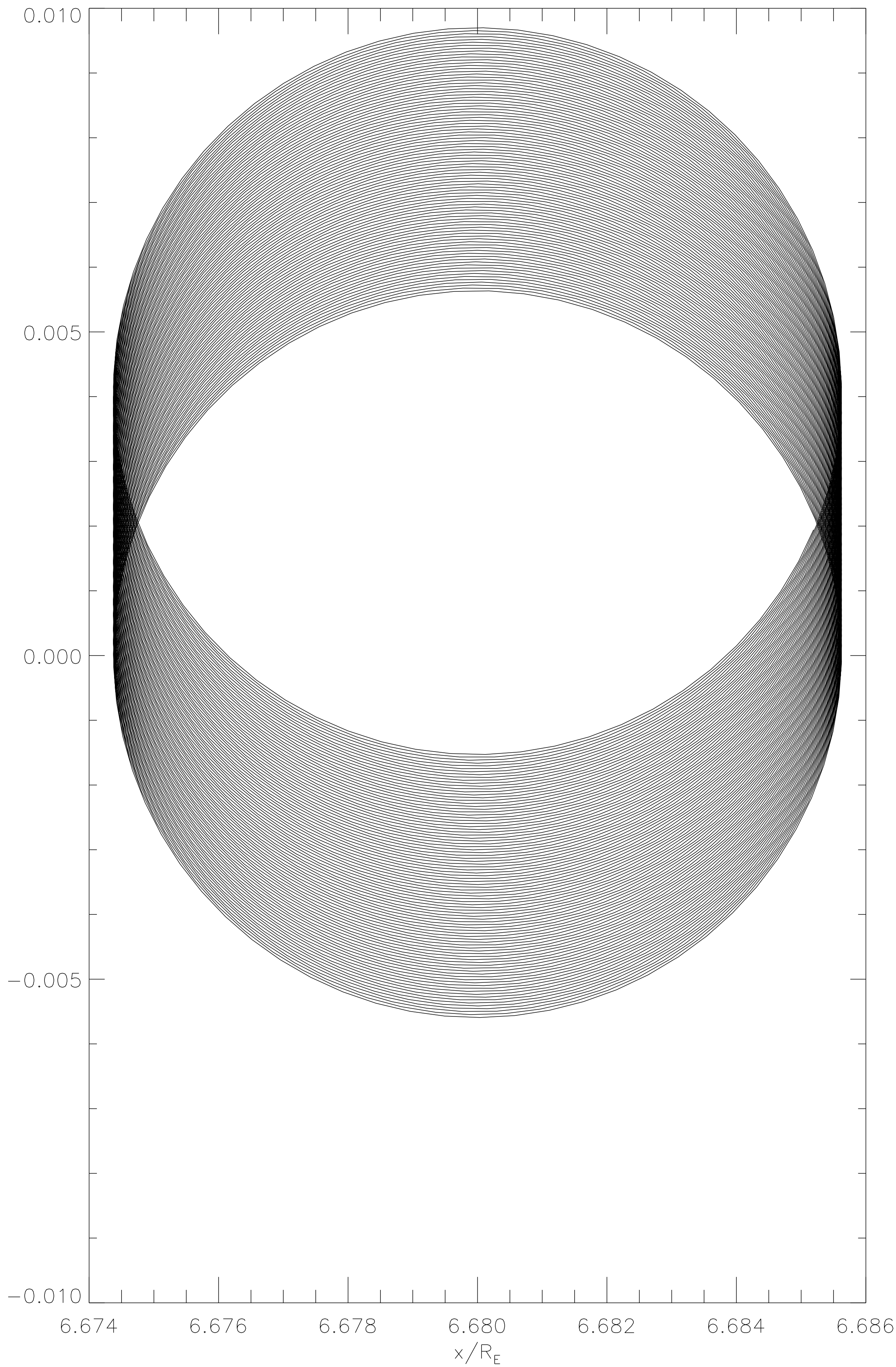
$$\bar{V}_G = \frac{3\times1.6\times10^{-16}}{1.6\times10^{-19}\cdot10^{-7}}\frac{1}{(6.68\times6.4\times10^6)}\ m/s$$

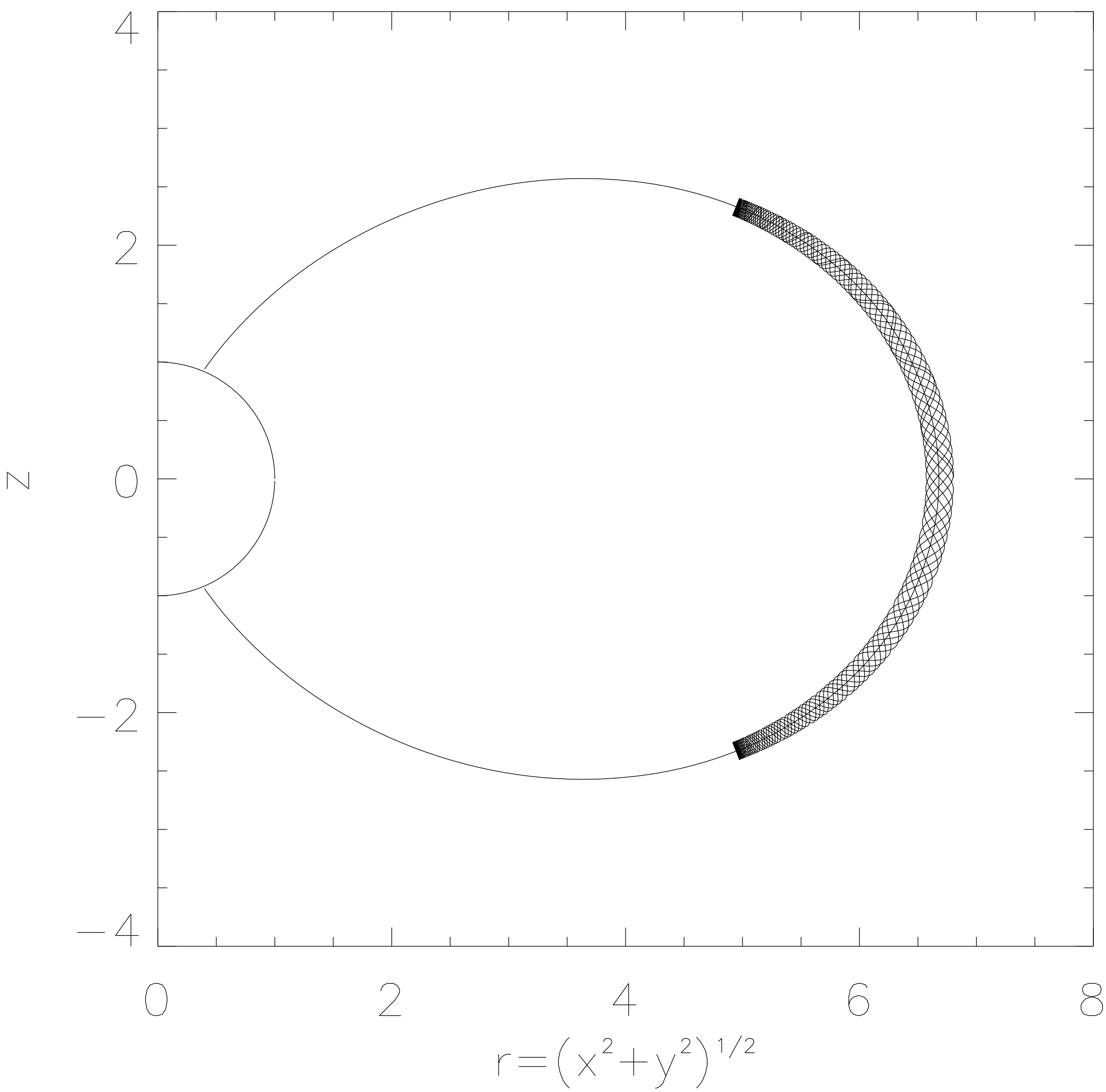
$V_G \approx 700$ m/s $\simeq 0.7$ km/s

---

from the Code $\bar{V}_{G\,numerical} \approx \frac{0.005\,R_E}{60\text{ sec}} \approx 500$ m/s

d) attached for $KE = 10^6$ eV $\alpha_{pitch} = \pi/4.$


0.010
0.005
0.000
−0.005
−0.010
y/R_E
6.674
6.676
6.678
6.680
6.682
6.684
6.686
x/R_E



4
2
0
−2
−4
z
0
2
4
6
8
$r=(x^2+y^2)^{1/2}$

Contributors

Mihaly Horanyi email: horanyi@colorado.edu

```
c=============================================================c
c  (in this code  cgs units are used )                        c
c                                                             c
c=============================================================c
c earth.f
c PHYS 5150 Spring, 2009

      implicit real*8 (a-h,o-z)
      dimension ff(6)
      common /params/ B_0,pm,q,r_e,dpm,c,pi
      real L, l_m, l_p, l_pp
      external diff
      logical OK

      call startup
c initial conditions
      L=6.68
      x=r_e*L
      y=0.
      z=0.
c energy KE = 1.e6 ev

      xe=1.e6*1.e-12

      v=sqrt(2*xe/pm)
c mirror lattitude
      l_m=20.*pi/180.
c B at the equator
      B_e=B_0/L**3


c pitch angle l_p

      l_p=pi/4.
      Mihaly Hor…

Mihaly…
      vx=0.
      vy=v*sin(l_p)
      vz=v*cos(l_p)
c Larmor radius
      r_L=pm*vy*c/(q*B_e)
        x=x+r_L
c integrator settings:
      n=6
      acc=1.e-12
      h=1.e-1
      hmin=1.e-13
      jtest=0

      ff(1)=x
      ff(2)=y
      ff(3)=z
      ff(4)=vx
      ff(5)=vy
      ff(6)=vz
      t1=0.
      dt=1.e-2


      do 100 i=1,100*60
       t2=t1+dt
c to avoid crash
```

```
          r1=sqrt(ff(1)**2+ff(2)**2+ff(3)**2)
          if(r1.le.r_e) go to 1000
c integration begins
       call merson(t1,t2,ff,6,acc,h,hmin,jtest,OK,diff)

      t1=t2
      r=sqrt(ff(1)**2+ff(2)**2)/r_e
      xre=ff(1)/r_e
      yre=ff(2)/r_e
      zre=ff(3)/r_e
      write(8,66),t1,xre,yre,zre,ff(4)/1.e5,ff(5)/1.e5,ff(6)/1.e5

66    format(7e15.7)
100   continue

 1000 continue

      stop
      end

cccccccccccccccccccccccc

      subroutine startup
      implicit real*8 (a-h,o-z)
      common /params/ B_0,pm,q,r_e,dpm,c,pi

c proton
       pm=1.6726e-24
       q=4.8e-10

C Earth
       r_e=6384.4e5
       B_0=0.3
       dpm=B_0*r_e**3
c      print*,'dpm=',dpm

c speed of light
       c=3.e10
      pi=acos(-1.)
      return
      end


      subroutine diff(t,ff,ffp)
      implicit real*8 (a-h,o-z)
      common /params/ B_0,pm,q,r_e,dpm,c,pi
      common /force/ flx,fly,flz
      dimension ff(6), ffp(6)

      x=ff(1)
      y=ff(2)
      z=ff(3)
      vx=ff(4)
      vy=ff(5)
      vz=ff(6)
      call forces(x,y,z,vx,vy,vz,fx,fy,fz)
      ffp(1)=vx
      ffp(2)=vy
      ffp(3)=vz
      ffp(4)=fx/pm
      ffp(5)=fy/pm
      ffp(6)=fz/pm

      return
      end


      subroutine forces(x,y,z,vx,vy,vz,fxl,fyl,fzl)
```

```
      implicit real*8 (a-h,o-z)
      common /params/ B_0,pm,q,r_e,dpm,c,pi

      call fields(x,y,z,bx,by,bz)
c Lorentz force
       fxl=q*(vy*bz-vz*by)/c
       fyl=q*(vz*bx-vx*bz)/c
       fzl=q*(vx*by-vy*bx)/c

      return
      end


      subroutine fields(x,y,z,bx,by,bz)
      implicit real*8 (a-h,o-z)

      common /params/ B_0,pm,q,r_e,dpm,c,pi

c aligned, centered dipole
       r2=x**2+y**2+z**2
       r=sqrt(r2)
       r3=r**3
       bx=3.*x/r3*z*dpm/r2
       by=3.*y/r3*z*dpm/r2
       bz=(3.*z*z/r2-1.)*dpm/r3

      return
      end


      SUBROUTINE MERSON (X,XEND,Y,N,ACC,H,HMIN,JTEST,OK,DIFF)
      implicit real*8 (a-h,o-z)

c this is a nice integrator, one can force integration forward even
c if the requested accuracy is not achieved
c

C     CERN LIBRARY NO D 208.
C
C     REVISED VERSION JULY 1971.
C
C     THIS VERSION OF MERSON IS A MODIFICATION OF THE PROGRAM RECEIVED
C     FROM KJELLER COMPUTER INSTALLATION NORWAY.THE MAIN DIFFERENCE
C     BEEING A CHANGE OF THE TEST FOR STEPLENGTH HALVING.
C
C
C     PURPOSE = STEP BY STEP INTEGRATION OF A SYSTEM OF FIRST ORDER
C               DIFFERENTIAL EQUATIONS
C
C               DYK(X)/DX=FK(X,Y1(X),Y2(X),.....,YN(X)) , K=1(1)N
C
C               WITH AUTOMATIC ERROR CONTROL USING THE METHOD DUE TO
C               MERSON.
C
C     PARAMETERS
C
C     X       = START VALUE FOR THE DOMAIN OF INTEGRATION (INPUT POINT).
C     XEND    = END VALUE FOR THE DOMAIN OF INTEGRATION (OUTPUT POINT).
C     Y       = ARRAY CONTAINING THE DEPENDENT VARIABLES.WHEN ENTERING
C               THE ROUTINE THE FUNCTION VALUES YK(X),K=1(1)N AND WHEN
C               RETURNING TO THE CALLING PROGRAM THE COMPUTED FUNCTION
C               VALUES YK(XEND),K=1(1)N.
C     N       = THE NUMBER OF DIFFERENTIAL EQUATION,EQUAL OR LESS 100.
C     ACC     = PRESCRIBED RELATIVE ACCURACY (TO BE OBTAINED FOR ALL
C               FUNCTION VALUES IN THE ARRAY Y).
C     H       = INITIAL STEPLENGTH.
C     HMIN    = ABSOLUTE VALUE OF MINIMUM STEPLENGTH WANTED.
C     JTEST   = TEST PARAMETER RELATED TO THE STEPLENGTH IN THE FOLLOW-
```

```
C              ING WAY.
C              JTEST = 0 , IF DURING THE CALCULATION WE GET
C                          ABS(H).LT.HMIN (BY REPEATED HALVING OF THE
C                          STEPLENGTH),THEN AN ERROR MESSAGE IS PRINT-
C                          ED,OK SET EQUAL TO .FALSE. FOLLOWED BY RE-
C                          TURN TO THE CALLING PROGRAM.
C              JTEST = 1 , CHECKING AS FOR JTEST=0,BUT THE CALCULATION
C                          WILL CONTINUE WITH THE FIXED STEPLENGTH HMIN
C     OK     = A LOGICAL VARIABLE WHICH IS SET EQUAL TO .TRUE. WHEN EN-
C              TERING THE ROUTINE.THE VALUE LATER IS DEPENDING OF JTEST
C              IN THE FOLLOWING WAY
C              JTEST = 1 , OK = .TRUE. ALWAYS.
C              JTEST = 0 , OK = .FALSE. IF THE STEPLENGTH BECOMES TOO
C              SMALL (SEE DESCRIPTION FOR JTEST).
C     DIFF   = USER SUPPLIED SUBROUTINE FOR THE CALCULATION OF THE
C              RIGHT HAND SIDES OF THE SYSTEM OF DIFFERENTIAL EQUATIONS
C              (I.E.THE FIRST ORDER DERIVATIVES).CALLING SEQUENCE
C              CALL DIFF(X,W,F) , WHERE
C              X = THE CURRENT VALUE OF THE ARGUMENT
C              W = AN ARRAY WITH THE ELEMENTS WK(X),K=1(1)N I.E. THE
C                  FUNTION VALUES FOR WHICH THE DERIVATIVES ARE TO BE
C                  COMPUTED.
C              F = AN ARRAY WITH THE ELEMENTS FK(X),K=1(1)N (WE HAVE
C                  FK(X)=FK(X,W1(X),W2(X),....,WN(X))) TO BE COMPUTED
C                  BY  DIFF AND RETURNED TO MERSON.
C              THE CHOSEN NAME FOR DIFF MUST APPEAR IN AN EXTERNAL
C              STATEMENT IN THE PROGRAM CALLING MERSON.
C
C     THE ARRAYS IN THE COMMON BLOCK BELOW ARE ONLY USED INTERNALLY IN
C     MERSON (AND DIFF) AND ARE TO FREE DISPOSAL OUTSIDE MERSON.
C     THE MAXIMUM NUMBER OF EQUATIONS WHICH  MAY BE INTEGRATED ARE 100.
C     THIS NUMBER MAY BE CHANGED BY CHANGING THE DIMENSION IN THE
C     COMMON BLOCK BELOW ACCORDINGLY.
C
      COMMON / LOCAL / YZ(100) , A(100) , B(100) , F(100) , W(100)
C
      LOGICAL OK
      DIMENSION Y(N)
C
C     RZERO IS A NUMBER WITH A MAGNITUDE OF ORDER EQUAL TO THE NOISE
C     LEVEL OF THE MACHINE I.E. IN THE RANGE OF THE ROUNDING OFF ERRORS.
C
      DATA RZERO / 1.E-13 /
C
C     CHECK NUMBER OF EQUATIONS,EQUAL OR LESS THAN 100.
C
      IF (N.GT.100) GO TO 86
C
      OK=.TRUE.
C
C     STORE INTERNALLY PARAMETERS IN LIST
C
      NN=N
      DO 1 K=1,NN
    1 W(K)=Y(K)
      Z=X
      ZEND=XEND
      BCC=ACC
      ZMIN=HMIN
      ITEST=JTEST
      S   =H
      ISWH=0
C
    2 HSV=S
      COF=ZEND-Z
      IF (ABS(S).LT.ABS(COF)) GO TO 8
      S=COF
      IF (ABS(COF/HSV).LT.RZERO) GO TO 50
```

```
      ISWH=1
C
C     IF ISWH=1 THEN  S IS EQUAL TO MAXIMUM POSSIBLE STEPLENGTH
C     WITHIN THE REMAINING PART OF THE DOMAIN OF INTEGRATION.
C
    8 DO 10 K=1,NN
   10 YZ(K)=W(K)
   12 HT=.3333333333333*S
C
      CALL DIFF(Z,W,F)
C
      Z=Z+HT
      DO 20 K=1,NN
      A(K)=HT*F(K)
   20 W(K)=A(K)+YZ(K)
C
      CALL DIFF(Z,W,F)
C
      DO 22 K=1,NN
      A(K)=.5*A(K)
   22 W(K)=.5*HT*F(K)+A(K)+YZ(K)
C
      CALL DIFF(Z,W,F)
C
      Z=Z+.5*HT
      DO 24 K=1,NN
      B(K)=4.5*HT*F(K)
   24 W(K)=.25*B(K)+.75*A(K)+YZ(K)
C
      CALL DIFF(Z,W,F)
C
      Z=Z+.5*S
      DO 26 K=1,NN
      A(K)=2.*HT*F(K)+A(K)
   26 W(K)=3.*A(K)-B(K)+YZ(K)
C
      CALL DIFF(Z,W,F)
C
      DO 28 K=1,NN
      B(K)=-.5*HT*F(K)-B(K)+2.*A(K)
      W(K)=W(K)-B(K)
      A(K)=ABS(5.*BCC*W(K))
      B(K)=ABS(B(K))
      IF (ABS(W(K)).LE.RZERO) GO TO 28
      IF(B(K).GT.A(K)) GO TO 60
   28 CONTINUE
C
C     REQUIRED ACCURACY OBTAINED FOR ALL COMPUTED FUNCTION VALUES.
C
      IF (ISWH.EQ.1) GO TO 50
C
C     TEST IF STEPLENGTH DOUBLING IS POSSIBLE.
C
   40 DO 42 K=1,NN
      IF(B(K).GT. .03125*A(K)) GO TO 2
   42 CONTINUE
      S=S+S
      GO TO 2
C
C     CALCULATION FINISHED.REPLACE INPUT FUNCTION VALUES WITH THE FUNC-
C     TION VALUES COMPUTED FOR THE OUTPUT POINT XEND.REPLACE INPUT STEP-
C     LENGTH H WITH NEW COMPUTED STEPLENGTH.
C
   50 H=HSV
      X=Z
      DO 52 K=1,NN
   52 Y(K)=W(K)
C
```

```
      RETURN
C
C     REQUIRED ACCURACY NOT OBTAINED.
C
C
   60 COF=.5*S
      IF (ABS(COF).GE.ZMIN) GO TO 80
      IF (ITEST.EQ.0) GO TO 84
C
C     JTEST=1,CONTINUE WITH CONSTANT STEPLENGTH EQUAL HMIN.
C
      S=ZMIN
      IF (HSV.LT.0.) S=-S
      IF (ISWH.EQ.1) GO TO 50
      GO TO 2
C
C     DO CALCULATIONS RELATED TO HALVING OF STEPLENGTH.
C
   80 DO 82 K=1,NN
   82 W(K)=YZ(K)
      Z=Z-S
      S=COF
      ISWH=0
      GO TO 2
C
C     JTEST=0 AND ABS(S).LT.HMIN.PRINT ERROR MESSAGE,SET OK=.FALSE. AND
C     RETURN TO CALLING PROGRAM.
C
   84 PRINT 88 , ITEST , S , ZMIN , Z
      OK=.FALSE.
      GO TO 50
C
   86 PRINT 90 , N
      STOP
C
   88 FORMAT(//,5X,31H*** SUBROUTINE MERSON ERROR ***,2X,8HJTEST = ,I2,
     12X,4HH = ,E12.5,2X,7HHMIN = ,E12.5,2X,4HX = ,E12.5,//)
   90 FORMAT(//,5X,31H*** SUBROUTINE MERSON ERROR ***,4H  N=,I4,1X,55HGR
     1EATER THAN THE MAXIMUM NUMBER OF EQUATIONS PERMITTED.,//)
C
      END
C
C
C
C
```

Contributors

 **Mihaly Horanyi** email: horanyi@colorado.edu

```
pro traj
; IDL routine to plot the trajectory calculated in earth.f
```

Mihaly Horanyi

```
L=6.68
lm=(findgen(180)-90)*!pi/180.0
r=L*cos(lm)^2
x=r*cos(lm)
y=r*sin(lm)
ind=where(r gt 1)
set_xy,0,7,-4,4
plot,x(ind),y(ind),/isotropic,xtit='r=(x!e2!n+y!e2!n)!e1/2!n', ytit='z', chars=2

lm=findgen(360)*!Pi/180
x=cos(lm)
y=sin(lm)
oplot,x,y

lm=20*!pi/180.
x=findgen(100)*0.1
y=x*tan(lm)
;oplot,x,y
;oplot,x,-y

lm=40*!pi/180.
x=findgen(100)*0.1
y=x*tan(lm)
;oplot,x,y
;oplot,x,-y

lm=60*!pi/180.
x=findgen(100)*0.1
y=x*tan(lm)
;oplot,x,y
;oplot,x,-y
read7,'fort.8',t,x,y,z,vx,vy,vz
r=sqrt(x*x+y*y)
en=sqrt(vx*vx+vy*vy+vz*vz)
err=(en-en(0))/en(0)
oplot,r,z

stop
end
```