Subject: need to speed up Runge-Kutta section
Posted by pdeyoung on Wed, 04 May 2005 22:45:04 GMT
View Forum Message <> Reply to Message

The code below basically tracks particles through a magnetic field
using 4th order RK. We are still running validation checks to find the
typos but we know now that it is too slow. As currently written is
does 1000 trajectories in about 0.5 second but ultimately we will need
to generate about 10^6 for the project. The current speed is doable
but I wonder if anyone can see a way to speed up the RK section (search
for SYSTIME to find the beginning and end of the slow RK section). I
know there is an RK4 built-in but worried that all the function calls
would be even slower. Thankyou in advance for any suggestions. I am
using IDL6.1

Paul DeYoung
deyoung@hope.edu

```
pro rk_fullfield_v5

; I feel the need for speed!

common keep_track,x,y,z,vx,vy,vz
common
 keep_data,datax,datay,dataz,x_data,y_data,z_data,number_of_c
hannels_x,number_of_channels_y,number_of_channels_z
common constants,mass,gamma
;start map stuff

number_of_channels_x = 150
number_of_channels_y = 5
number_of_channels_z = 200 ;150 by 200 grid
x_bin_width = 0.0050336 ;(-550 to +200 in 149 steps)
y_bin_width = 0.22098;
z_bin_width = 0.0072864 ;(-300 to +1150 in 199 steps)
min_x_value = -0.550
max_x_value = (number_of_channels_x - 1) * x_bin_width + min_x_value
min_y_value = -2*.22098
max_y_value = (number_of_channels_y - 1) * y_bin_width + min_y_value
min_z_value = -0.300
max_z_value = (number_of_channels_z - 1) * z_bin_width + min_z_value
;print,max_z_value


;file = dialog_pickfile() ;generic way to pick the data file that need
to be preprocessed
file2 = "c:\user\map_tracking\map2.txt"
openr, lun2, file2, /get_lun     ;opens data file for reading
```

```
file3 = "c:\user\map_tracking\map3.txt"
openr, lun3, file3, /get_lun    ;opens data file for reading
file4 = "c:\user\map_tracking\map4.txt"
openr, lun4, file4, /get_lun    ;opens data file for reading
file5 = "c:\user\map_tracking\map5.txt"
openr, lun5, file5, /get_lun    ;opens data file for reading
file6 = "c:\user\map_tracking\map6.txt"
openr, lun6, file6, /get_lun    ;opens data file for reading



data2=fltarr(number_of_channels_x,number_of_channels_z)
data3=fltarr(number_of_channels_x,number_of_channels_z)
data4=fltarr(number_of_channels_x,number_of_channels_z)
data5=fltarr(number_of_channels_x,number_of_channels_z)
data6=fltarr(number_of_channels_x,number_of_channels_z)

x_data = fltarr(number_of_channels_x)
y_data = fltarr(number_of_channels_y)
z_data = fltarr(number_of_channels_z)

readf, lun2, data2,FORMAT='(150F0)'
readf, lun3, data3,FORMAT='(150F0)'
readf, lun4, data4,FORMAT='(150F0)'
readf, lun5, data5,FORMAT='(150F0)'
readf, lun6, data6,FORMAT='(150F0)'



free_lun,lun2
free_lun,lun3
free_lun,lun4
free_lun,lun5
free_lun,lun6

for i = 0,number_of_channels_x-1, 1 do begin
x_data(i) = float(i) * x_bin_width + min_x_value
endfor


for i = 0,number_of_channels_y-1, 1 do begin
y_data(i) = float(i) * y_bin_width + min_y_value
endfor

for i = 0,number_of_channels_z-1, 1 do begin
z_data(i) = float(i) * z_bin_width + min_z_value
endfor
```

```
;end map stuff


;make three d arrays

datax =
 fltarr(number_of_channels_x,number_of_channels_y,number_of_c hannels_z)
datay =
 fltarr(number_of_channels_x,number_of_channels_y,number_of_c hannels_z)
dataz =
 fltarr(number_of_channels_x,number_of_channels_y,number_of_c hannels_z)

;fill the datay array

datay(*,0,*) = data2(*,*)
datay(*,1,*) = data3(*,*)
datay(*,2,*) = data4(*,*)
datay(*,3,*) = data5(*,*)
datay(*,4,*) = data6(*,*)

;ISURFACE, data2(*,*),x_data,z_data
;ISURFACE, data3(*,*),x_data,z_data
;ISURFACE, data4(*,*),x_data,z_data
;ISURFACE, data5(*,*),x_data,z_data
;ISURFACE, data6(*,*),x_data,z_data


;generate the other components of the field
call_procedure, 'build_field',x_bin_width, y_bin_width, z_bin_width
;data_plot =
 fltarr(number_of_channels_x,number_of_channels_y,number_of_c hannels_z)
;data_plot =
 reform(datax(*,0,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="x2 "
;data_plot =
 reform(datax(*,1,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="x3 "
;data_plot =
 reform(datax(*,2,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="x4 "
;data_plot =
 reform(datax(*,3,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="x5 "
;data_plot =
 reform(datax(*,4,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="x6 "
;data_plot =
```

```
 reform(datay(*,0,*),number_of_channels_x,number_of_channels_ z)
;ICONTOUR,data_plot,x_data,z_data,N_LEVELS=10,TITLE="y2"
;data_plot =
 reform(datay(*,1,*),number_of_channels_x,number_of_channels_ z)
;ICONTOUR,data_plot,x_data,z_data,N_LEVELS=10,TITLE="y3"
;data_plot =
 reform(datay(*,2,*),number_of_channels_x,number_of_channels_ z)
;ICONTOUR,data_plot,x_data,z_data,N_LEVELS=10,TITLE="y4"
;data_plot =
 reform(datay(*,3,*),number_of_channels_x,number_of_channels_ z)
;ICONTOUR,data_plot,x_data,z_data,N_LEVELS=10,TITLE="y5"
;data_plot =
 reform(datay(*,4,*),number_of_channels_x,number_of_channels_ z)
;ICONTOUR,data_plot,x_data,z_data,N_LEVELS=10,TITLE="y6"
;data_plot =
 reform(dataz(*,0,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="z2 "
;data_plot =
 reform(dataz(*,1,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="z3 "
;data_plot =
 reform(dataz(*,2,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="z4 "
;data_plot =
 reform(dataz(*,3,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="z5 "
;data_plot =
 reform(dataz(*,4,*),number_of_channels_x,number_of_channels_ z)
;ISURFACE,data_plot,x_data,z_data,ZRANGE=[-10,10],TITLE="z6 "

;start of SLOW rk stuff

T = SYSTIME(1)

magic_cos = cos(2.7381*!PI/180.)
magic_sin = sin(2.7381*!PI/180.)

;initial conditions
x_initial = -0.04805 ;meter - set by map
y_initial = 0.0
z_initial = -0.100 ; set by map

;generate 1000 tracks

for ienergy=.9,1.1,.001 do begin
;print,"factor",ienergy
```

```
constant = (10*ienergy*3.2718*300/931.494/26.2114)^2
beta = sqrt(constant/(1+constant))
;fix this for three dimension

theta_inplane = 0.0*!PI/180.
theta_outplane = 1.0*!PI/180.
tan_theta_inplane = tan(theta_inplane)
tan_theta_outplane = tan(theta_outplane)

beta_inplane_squared =
 (tan_theta_inplane)^2*beta^2/(1+(tan_theta_outplane)^2+(tan_ theta_inplane)^2)
beta_inplane = sqrt(beta_inplane_squared)
beta_outplane_squared =
 (tan_theta_outplane)^2*beta^2/(1+(tan_theta_outplane)^2+(tan _theta_inplane)^2)
beta_outplane = sqrt(beta_outplane_squared)
beta_z = sqrt(beta^2-beta_inplane_squared-beta_outplane_squared)
;print,tan(theta_inplane),tan(theta_outplane)
;print,beta,beta_inplane,beta_outplane,beta_z



vx_initial = (magic_cos*beta_inplane-magic_sin*beta_z)*3e8
vy_initial = beta_outplane*3e8
vz_initial = (beta_inplane*magic_sin+magic_cos*beta_z)*3e8
;meter/second
;print,vx_initial,vy_initial,vz_initial
;By = 1 ; Tesla
mass = 26.2114*(931.494e6)*(1.6e-19)/(3e8)^2 ;kilograms

q = 10*1.6e-19 ; coulombs

gamma = 1/sqrt(1-(beta)^2) ; unitless

;print,gamma,mass


x = fltarr(1001)
y = fltarr(1001)
z = fltarr(1001)

vx = fltarr(1001)
vy = fltarr(1001)
vz = fltarr(1001)
```

```
x(0) = x_initial
y(0) = y_initial
z(0) = z_initial
vx(0) = vx_initial
vy(0) = vy_initial
vz(0) = vz_initial

step = 2e-11 ; seconds
times = findgen(1001) * step


;do the r-k here
for i = 0, 999, 1 do begin

;get field value

x_index = ((x(i) - min_x_value) / x_bin_width)
y_index = ((y(i) - min_y_value) / y_bin_width)
z_index = ((z(i) - min_z_value) / z_bin_width)

Bx = INTERPOLATE(datax, x_index, y_index, z_index, MISSING = 0)
By = INTERPOLATE(datay, x_index, y_index, z_index, MISSING = 0)
Bz = INTERPOLATE(dataz, x_index, y_index, z_index, MISSING = 0)

if (Bx+By+Bz EQ 0) then begin
   step = step*500
   times(i+1) = times(i)+step
endif

vx_temp = vx(i)
vy_temp = vy(i)
vz_temp = vz(i) ;this was faster than replacing the *_temp below

k1x = step*(vx_temp)
k2x = step*(vx_temp+k1x/2.)
k3x = step*(vx_temp+k2x/2.)
k4x = step*(vx_temp+k3x/2.)

k1y = step*(vy_temp)
k2y = step*(vy_temp+k1y/2.)
k3y = step*(vy_temp+k2y/2.)
k4y = step*(vy_temp+k3y/2.)

k1z = step*(vz_temp)
k2z = step*(vz_temp+k1z/2.)
k3z = step*(vz_temp+k2z/2.)
k4z = step*(vz_temp+k3z/2.)
```

```
k1vx = step*q*(vy_temp*Bz-vz_temp*By)/gamma/mass
k1vy = step*q*(vz_temp*Bx-vx_temp*Bz)/gamma/mass
k1vz = step*q*(vx_temp*By-vy_temp*Bx)/gamma/mass

k2vx = step*q*((vy_temp+k1vy/2)*Bz-(vz_temp+k1vz/2)*By)/gamma/mass
k2vy = step*q*((vz_temp+k1vz/2)*Bx-(vx_temp+k1vx/2)*Bz)/gamma/mass
k2vz = step*q*((vx_temp+k1vx/2)*By-(vy_temp+k1vy/2)*Bx)/gamma/mass

k3vx = step*q*((vy_temp+k2vy/2)*Bz-(vz_temp+k2vz/2)*By)/gamma/mass
k3vy = step*q*((vz_temp+k2vz/2)*Bx-(vx_temp+k2vx/2)*Bz)/gamma/mass
k3vz = step*q*((vx_temp+k2vx/2)*By-(vy_temp+k2vy/2)*Bx)/gamma/mass

k4vx = step*q*((vy_temp+k3vy/2)*Bz-(vz_temp+k3vz/2)*By)/gamma/mass
k4vy = step*q*((vz_temp+k3vz/2)*Bx-(vx_temp+k3vx/2)*Bz)/gamma/mass
k4vz = step*q*((vx_temp+k3vx/2)*By-(vy_temp+k3vy/2)*Bx)/gamma/mass


x(i+1) = x(i)+(1/6.)*(k1x+2*k2x+2*k3x+k4x)
y(i+1) = y(i)+(1/6.)*(k1y+2*k2y+2*k3y+k4y)
z(i+1) = z(i)+(1/6.)*(k1z+2*k2z+2*k3z+k4z)
vx(i+1) = vx(i)+(1/6.)*(k1vx+2*k2vx+2*k3vx+k4vx)
vy(i+1) = vy(i)+(1/6.)*(k1vy+2*k2vy+2*k3vy+k4vy)
vz(i+1) = vz(i)+(1/6.)*(k1vz+2*k2vz+2*k3vz+k4vz)

if (step GE 1e-9) then break

endfor

xlast = x(i+1)
xntlast = x(i)
ylast = y(i+1)
yntlast = y(i)
zlast = z(i+1)
zntlast = z(i)

;print,i,z(i),z(i+1),times(i),times(i+1),step

;get the angle and position in CRDC frame

thetarad = atan((xlast-xntlast)/(zlast-zntlast))
theta = thetarad*180/!PI
;print,"x angle, x angle in fp frame",theta,theta+45.7381

;eq of crdc1 is x=.97456*z-1.995
;find B for track
trackb=xlast-tan(thetarad)*zlast
trackz=(-1.995-trackb)/(tan(thetarad)-0.97456)
```

```
trackx = 0.97456*trackz-1.995
crdcx=sqrt((trackx+0.80314)^2+(trackz-1.22304)^2)
if (trackz LT 1.22304) then begin
crdcx = -crdcx
endif
;print,"z,x at crdc1, x dist from crdc center",trackz,trackx,crdcx

;get the angle and position in CRDC frame

;note angle correction for 43 degrees
thetarad = atan((ylast-yntlast)/((zlast-zntlast)/cos(43*!PI/180.)))
theta = thetarad*180/!PI
;print,"y angle in fp frame",theta

;recalc theta in map frame
;find B for track
thetarad = atan((ylast-yntlast)/(zlast-zntlast))
trackb=ylast-tan(thetarad)*zlast
tracky = tan(thetarad)*trackz+trackb
crdcy = tracky
;print,"y dist from crdc center",crdcy


;find the flight time to the crdc1
;find closest index in z array

tof = INTERPOL(times(0:i+1),z(0:i+1),trackz)
;print,"tof to crdc",tof
dist = beta*0.30*tof*1e9
;print,"flight path",dist
endfor

PRINT, SYSTIME(1) - T, 'Seconds'



IPLOT,z(0:i+1),x(0:i+1),COLOR=[255,0,0]
IPLOT,z(0:i+1),y(0:i+1),COLOR=[255,0,0]


;ICONTOUR,data4,x_data,z_data
;IPLOT,x,z,COLOR=[255,0,0],OVERPLOT=1


end


pro interp_B, x_value, y_value, z_value, x_bin_width, min_x_value,
```

```
y_bin_width, min_y_value, z_bin_width, min_z_value, Bx, By, Bz

common
 keep_data,datax,datay,dataz,x_data,y_data,z_data,number_of_c
hannels_x,number_of_channels_y,number_of_channels_z

x_index = ((x_value - min_x_value) / x_bin_width)
y_index = ((y_value - min_y_value) / y_bin_width)
z_index = ((z_value - min_z_value) / z_bin_width)

Bx = INTERPOLATE(datax, x_index, y_index, z_index, MISSING = 0)
By = INTERPOLATE(datay, x_index, y_index, z_index, MISSING = 0)
Bz = INTERPOLATE(dataz, x_index, y_index, z_index, MISSING = 0)

end



pro build_field, x_bin_width, y_bin_width, z_bin_width
common
 keep_data,datax,datay,dataz,x_data,y_data,z_data,number_of_c
hannels_x,number_of_channels_y,number_of_channels_z

;above midplane
for j=2,3,1 do begin
for i=0,number_of_channels_x-2,1 do begin
for k=0,number_of_channels_z-1,1 do begin


datax(i,j+1,k) =
 (datay(i+1,j,k)/x_bin_width-datay(i,j,k)/x_bin_width)*y_bin_ width+datax(i,j,k)

endfor
endfor
endfor


for j=2,3,1 do begin
for i=0,number_of_channels_x-1,1 do begin
for k=0,number_of_channels_z-2,1 do begin


dataz(i,j+1,k) =
 (datay(i,j,k+1)/z_bin_width-datay(i,j,k)/z_bin_width)*y_bin_ width+dataz(i,j,k)

endfor
endfor
```

```
endfor

;below midplane
for j=2,1,-1 do begin
for i=0,number_of_channels_x-1,1 do begin
for k=0,number_of_channels_z-2,1 do begin

dataz(i,j-1,k) =
 (datay(i,j,k)/z_bin_width-datay(i,j,k+1)/z_bin_width)*y_bin_ width+dataz(i,j,k)

endfor
endfor
endfor

for j=2,1,-1 do begin
for i=0,number_of_channels_x-2,1 do begin
for k=0,number_of_channels_z-1,1 do begin

datax(i,j-1,k) =
 (datay(i,j,k)/x_bin_width-datay(i+1,j,k)/x_bin_width)*y_bin_ width+datax(i,j,k)

endfor
endfor
endfor


for j=0,number_of_channels_y-1,1 do begin
for i=0,number_of_channels_x-1,1 do begin
for k=0,number_of_channels_z-1,1 do begin

datax(i,j,k) = datax(i,j,k)>(-10)
datax(i,j,k) = datax(i,j,k)<10
dataz(i,j,k) = dataz(i,j,k)>(-10)
dataz(i,j,k) = dataz(i,j,k)<10


endfor
endfor
endfor


end
```