ASTR 415

Evan Shipley-Friedt

October 18, 2022

Problem Set #3

1

1. As an example of an unstable algorithm, consider integer powers of the "Golden Mean" $\phi = (\sqrt{5}-1)/2$. It can be shown that $\phi^{n+1} = \phi^{n-1} - \phi^n$, i.e. successively higher powers of ϕ can be computed from a single subtraction rather than a more expensive multiply. Write a single-precision program to compute a table consisting of the columns n, ϕ^n computed from the recursion relation, and ϕ^n computed directly (i.e. $\phi^n = \phi\phi^{n-1}$), for n ranging from 1 to 20. Is the round-off error random? What happens in double precision?

Round-off error is not random, it originates when the value of $\phi^{n-1} - \phi^n$ becomes much smaller ratio such as when $\phi^{16} - \phi^{17}$ occurs, then $\phi^{17} - \phi^{18}$ produces a greater value in ϕ^{19} than ϕ^{18} which should not occur. From this the outrageously large error and actually a negative value in iteration 20 appears. In double precision this round-off error does not occur in 20 iterations. There is no sign of error to at least 3 significant digits. I get a stack smash with my code using double precision. Perhaps part of my code is not optimal for using double precision because my indexing had to start with i=1 which is not standard in C arrays.

Double Precision

```
| 19 |      0.000107      |    0.000107    |
-------------------------------------------
| 20 |      6.61e-05     |    6.61e-05    |
-------------------------------------------


*** stack smashing detected ***: terminated
make: *** [Makefile:49: run] Aborted
```

Floating-Point Precision Table

Golden Mean Power 'n' Recursion Relation

n	ϕn+1 = ϕ^(n-1)-ϕ^n	ϕn = ϕϕ^(n-1)
1	0.618	0.618
2	0.382	0.382
3	0.236	0.236
4	0.146	0.146
5	0.0902	0.0902
6	0.0557	0.0557
7	0.0344	0.0344
8	0.0213	0.0213
9	0.0132	0.0132
10	0.00813	0.00813
11	0.00503	0.00502
12	0.0031	0.00311
13	0.00192	0.00192
14	0.00118	0.00119
15	0.000743	0.000733
16	0.000437	0.000453
17	0.000306	0.00028
18	0.000131	0.000173
19	0.000176	0.000107
20	-4.49e-05	6.61e-05

2

2. Write a program to compute the instantaneous spin period of a rigid body made up of identical, discrete, point particles. Use the fact that the angular momentum is

$$\mathbf{L} = \sum_k m_k(\mathbf{r}_k \times \mathbf{v}_k) = \mathbf{I}\boldsymbol{\omega}, \tag{1}$$

where m_k is the mass of particle k, $\mathbf{r}_k$ and $\mathbf{v}_k$ are its position and velocity vectors with respect to the centre of mass, $\boldsymbol{\omega}$ is the spin vector, and $\mathbf{I}$ is the inertia tensor

$$\mathbf{I} = \sum_k m_k(r_k^2\mathbf{1} - \mathbf{r}_k\mathbf{r}_k),$$

where $\mathbf{1}$ is the unitmatrix.[1]

Write a program to solve Eq. (1) for $\boldsymbol{\omega}$ (I recommend you use the routines in *Numerical Recipes)*. The spin period is then $2\pi/|\boldsymbol{\omega}|$.

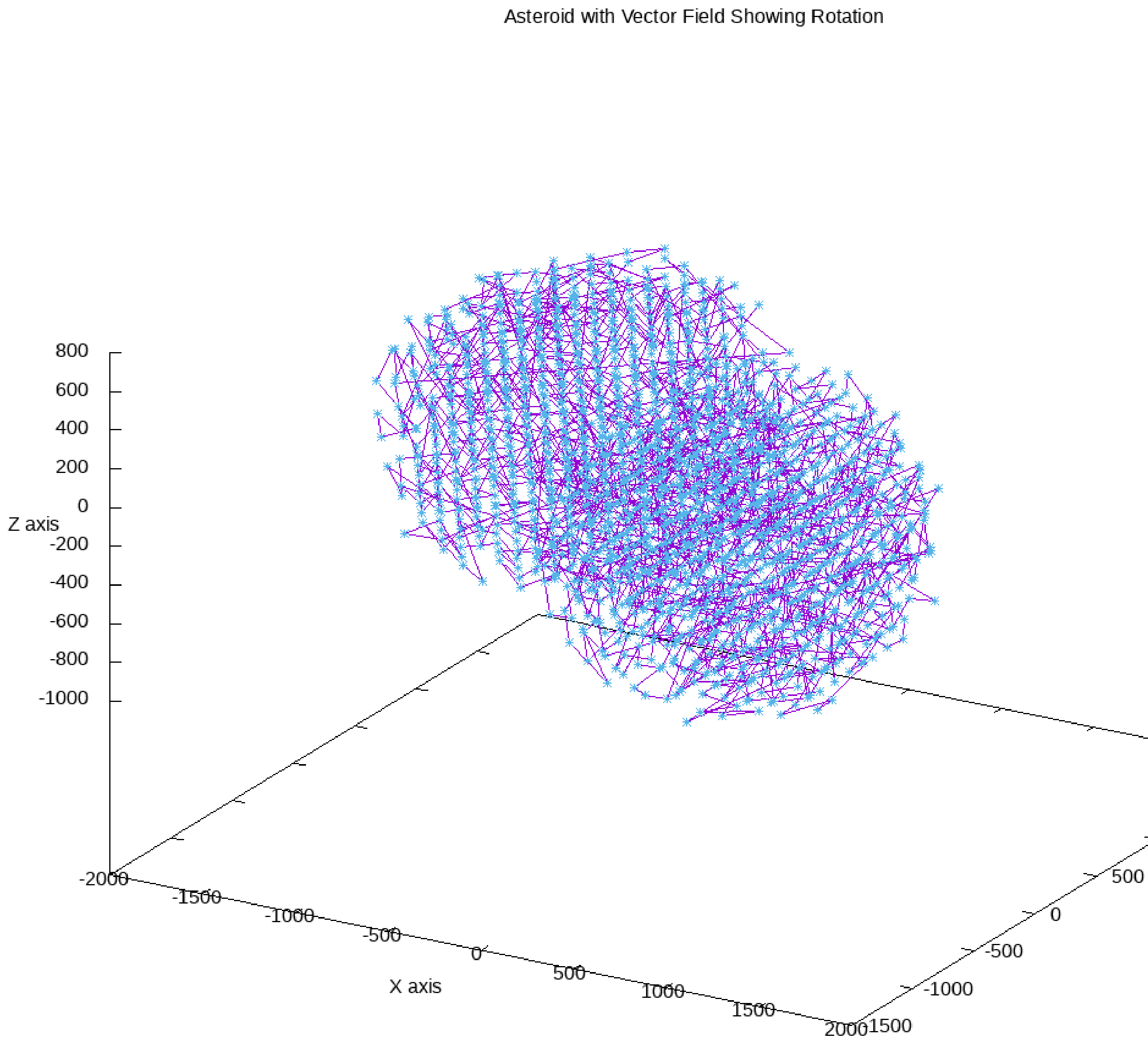
(a) Test your code by reading the data file

`http://www.astro.umd.edu/~ricotti/NEWWEB/teaching/ASTR415/ps2.dat`

which is in the format $x\ y\ z\ v_x\ v_y\ v_z$ (i.e. 6 values to a line separated by white space). The units are mks (SI). I have also uploaded ps2.dat to ELMS. What is the spin period in hours?

(b) Make a graphical representation of the body using your favorite graphing package. If you use 2-D projections, be sure to include enough viewing angles to get a complete picture.

```
./2

data row count: 1988

Converted I =

        1.73974e+09        7.38005e+08        -1.26285e+07
        7.38005e+08        1.73974e+09        -1.29503e+06
       -1.26285e+07       -1.29503e+06         2.28012e+09


Converted L =

           -3089.88           -240.005             -923897

 ---Gauss Jordan---

Gauss A-inverse =

         7.0096e-10       -2.97347e-10         3.71341e-12
       -2.97347e-10        7.00932e-10        -1.24876e-12
        3.71341e-12       -1.24876e-12         4.38593e-10

a times a-inverse:

        5.79774e-19          -4.16853e-19           4.60295e-21
       -4.16853e-19           5.79722e-19          -2.52717e-21
        4.60295e-21          -2.52717e-21           1.92379e-19


 Gauss-Jordan Solution:

         ω = < -5.52533e-06 , 1.90427e-06 , -0.000405226 >


 Period Solution: P = 2π/sqrt(ω_x^2 + ω_y^2 + ω_z^2) = 1.55e+04 sec = 4.3 hours

gnuplot < asteroid.plt
```


Asteroid with Vector Field Showing Rotation
800
600
400
200
0
Z axis
-200
-400
-600
-800
-1000
-2000
-1500
-1000
-500
0
500
1000
1500
2000
X axis
-1500
-1000
-500
0
500