
Subject: Re: need to speed up Runge-Kutta section
Posted by [Kenneth P. Bowman](#) on Thu, 05 May 2005 02:48:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1115246704.219698.144050@o13g2000cwo.googlegroups.com>, "pdeyoung" <deyoung@hope.edu> wrote:

> 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

Do you want to do 1000 steps, or 1000 particles? If 1000 particles, you can vectorize your code. As it is, you have the whole RK section inside a FOR loop, so it is running as purely scalar, interpolated code. (Very slow in IDL.) You need to think IDL-style and do all these operations as vectors.

When you convert to vector operations, you will probably find that the bottleneck is the interpolation, as that tends to access memory in random order (which uses the cache inefficiently). The RK section should vectorize essentially completely.

You can do some minor optimizations in the RK code by converting divisions to multiplications and avoiding unnecessary type conversions. Some of your constants are integers, but should be floats.

After converting to vectors, you can insert a few SYSTIME calls to determine where the slow sections are.

Ken Bowman
