
Subject: Nearest neighbor search in position vector array and curl of vector field
Posted by [p.j.sutton](#) on Wed, 21 May 2014 07:46:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

I want to be able to perform a search for the nearest neighbor to particle (i) in a three dimensional array (actually only really need to use the x and y positions as my model is two-dimensional). So basically I need to find the nearest neighbor to all particles and I have 6.5 million particles to consider. I need to find the position vector of the nearest and the velocity vector, so essentially need to particle or index value along with the vector.

I have looked at a few functions built in but am unsure which is the best approach and how to go about it. I have only really had a go at writing the initial conditions and some simply subtraction and multiplication of arrays so not that familiar with how to use of the more advanced functions.

The position vectors are obviously not on a uniform grid and neither are the velocity vectors. Essentially what I wanted to do was find the curl of the vector field, so it might be that there an easier way of trying to calculate this?

Any help would be greatly appreciated.

Regards

Phil

Subject: Re: Nearest neighbor search in position vector array and curl of vector field
Posted by [Craig Markwardt](#) on Fri, 23 May 2014 00:04:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, May 21, 2014 3:46:38 AM UTC-4, p.j.s...@lboro.ac.uk wrote:

> The position vectors are obviously not on a uniform grid and neither are the velocity vectors. Essentially what I wanted to do was find the curl of the vector field, so it might be that there an easier way of trying to calculate this?

Can you resample onto a uniform grid. That would obviously make things *much* easier, and amenable to vectorizing in IDL.

Even if you can't resample your data, you might consider using a 2 step search. First use a regular grid to narrow down the part of your sample that is near to your particle point. Then use standard cartesian to search the remaining samples. Even if you do this, I don't think you will be able to vectorize.

Craig
