Subject: Re: for loop is killing me

Posted by Conor on Thu, 08 Nov 2007 19:26:37 GMT

View Forum Message <> Reply to Message

Okay, here's one more for you. Your m loop and phi loop can both be made for loop-less. Here's your warning though: I haven't tested this code in the slightest. It's hard to test stuff when it is part of a much larger program, so I'll leave that to you. I'll just explain what this chunk of code is doing. So this replaces your m and your phi loops:

```
i_phi_max = 20.
phis = rebin(findgen(i_phi_max)+1,i_phi_max,i_phi_max)
i_ms = rebin(findgen(1,i_phi_max)+1,i_phi_max,i_phi_max)

gp = 2.0*(1./i_ms)*(r_minus/r_plus)^i_ms * cos(i_ms * phis*2*!PI/
i_phi_max )
gh = 2.0*(1./i_ms)*(r_minus*r_plus/a^2)^i_ms * cos(i_ms * phis*2*!PI/
i_phi_max )

G_in_phisum = total(gp,1)
G_out_phisum = total(gh,1)
```

The general idea here is to simply do all the calculations at once, combined into one big array operation. So for instance from looking at the phi loop you know that your phi's are going to go from 1-20 and are going to be used 20 times by the m-loop. Imagine this as a 20x20 array where each row contains the numbers 1-20. The second line (phis = rebin()) creates such a 20x20 array. Rebin is used to repeat the findgen() 20 times vertically.

Next consider your i_m's. These will also have the values 1-20 and will be used against the different values in the phis array. So you can consider it to be a 20x20 array where each column has the values 1-20. Once again, this array is created using rebin and findgen in a very similar way to how I created the phi array. Once this is accomplished you have two 20x20 arrays with all the 400 possible combinations of phi and m. Then, you perform your calculations exactly as you did in the for loop and in one go calculate all the possible results. Finally you use the dimension keyword to the total function to sum up along rows. Viola, no more for loops! To make sure things are clear, I want to explain this in a different way:

The idea is to do all your calculations at once. To make this happen I created a 20x20 array for the phi loop and a 20x20 array for the m loop. The idea is to pre-calculate all possible combinations of m and phi and calculate all the possible values at once. That's what the phis and ims arrays are for. Between them they contain all possible values for i_m and phi. Traveling along a row is the equivelent of looping over phi, and traveling along a column is the equivilent of looping over i_m. So for instance if, with your original set up, you were in the 3rd iteration of the m-loop and the 12th iteration of the phi loop the i_m variable would have the value of 3 and the phi variable would have the value of 12. With the new setup if you printed out the values of i_ms[11,2] and phis[11,2] you would get the values 3 and 12! Hopefully this all makes sense.

Also, please note that there is a potential source of confusion between my code and yours because for your phi-loop you looped from phi=0,19 but then added one later. In my code this meant that the findgen() for phi had a +1 after it.