Subject: Re: Removing if then else loop for efficiency
Posted by Tom Ashbee on Sun, 10 Jan 2010 14:46:44 GMT
View Forum Message <> Reply to Message

On Jan 10, 12:06 pm, Tom Ashbee <tlash...@googlemail.com> wrote:
> Hello,
>
> apologies for my newb-ness, but I was hoping I could get some help
> with increasing the efficiency of the following program:
>
> At the moment it is like this
>
> for i=0, N-1 do begin
> for j=0, N-1 do begin
> if i ne j then begin
> ;stuff
> endif else begin
> ;more stuff
> endif
> endfor
> endfor
>
> It's way too slow at the moment, but I gather I can remove one of the
> for loops and the if then else loop but I have no idea how!
>
> Any help would be much appreciated and I can post the full program too
> if it helps. FTR it's the RHS of a Hamiltonian.
>
> Thanks again

OK here's the whole program:

```
function velocities, t, xyvec

N = 50
R = 5.0

xvec = xyvec(0: N-1)
yvec = xyvec(N: (N*2)-1)

dxvecdt = dblarr(N)
dyvecdt = dblarr(N)

for i = 0, N-1 do begin

  dxvecdt(i) = 0.0d
  dyvecdt(i) = 0.0d
```

```
   for j = 0, N-1 do begin

      gam = gamma(N, 2.0d, 10.0d)

      if i ne j then begin

      dxvecdt(i) = dxvecdt(i) + ( gam(j) /(2.0d*!pi))*(yvec(j)-yvec
(i))/((xvec(i)-xvec(j))^2+(yvec(i)-yvec(j))^2)  -   ( gam(j) /(2.0d*!
 pi))*((yvec(j)*R^2)/((xvec(j))^2+(yvec(j))^2)-yvec(i))/((xve c(i)-(xvec
 (j)*R^2)/((xvec(j))^2+(yvec(j))^2))^2+(yvec(i)-(yvec(j)*R^2) /((xvec(j))
^2+(yvec(j))^2))^2)

      dyvecdt(i) = dyvecdt(i) + ( gam(j) /(2.0d*!pi))*(xvec(i)-xvec
(j))/((xvec(i)-xvec(j))^2+(yvec(i)-yvec(j))^2)   -  ( gam(j) /(2.0d*!
 pi))*(xvec(i)-(xvec(j)*R^2)/((xvec(j))^2+(yvec(j))^2))/((xve c(i)-(xvec
 (j)*R^2)/((xvec(j))^2+(yvec(j))^2))^2+(yvec(i)-(yvec(j)*R^2) /((xvec(j))
^2+(yvec(j))^2))^2)

       endif else begin

       dxvecdt(i) =  dxvecdt(i) - ( gam(j) /(2.0d*!pi))*((yvec(j)*R^2)/
 ((xvec(j))^2+(yvec(j))^2)-yvec(i))/((xvec(i)-(xvec(j)*R^2)/( (xvec(j))
 ^2+(yvec(j))^2))^2+(yvec(i)-(yvec(j)*R^2)/((xvec(j))^2+(yvec (j))^2))
^2)

       dyvecdt(i) = dyvecdt(i) - ( gam(j) /(2.0d*!pi))*(xvec(i)-(xvec
 (j)*R^2)/((xvec(j))^2+(yvec(j))^2))/((xvec(i)-(xvec(j)*R^2)/ ((xvec(j))
 ^2+(yvec(j))^2))^2+(yvec(i)-(yvec(j)*R^2)/((xvec(j))^2+(yvec (j))^2))
^2)

       endelse

     endfor

endfor

z = [dxvecdt, dyvecdt]

return, z

end
```