

# Re: request for algorithm

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2006-01/msg00090.html>

---

- *From:* [spellucci@xx](mailto:spellucci@xx) (Peter Spellucci)
  - *Date:* Thu, 12 Jan 2006 15:07:17 +0000 (UTC)
- 

In article <slrndscqj.bij.ndros@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>, ndros <ndros@xxxxxxxxxxxxxxxxxxxx> writes:  
>On 2006-01-12, loic-dev@xxxxxxxx <loic-dev@xxxxxxxx> wrote:  
>> Hi,  
>>  
>>  
>> I could eventually understand why you don't want to have a forward  
>> reference in time (you would have to solve a system of equations). But  
>> I can't really understand why you don't want to depend on the points  
>>  $U(x+1,y,t)$ ,  $U(x,y+1,t)$  for instance...  
>>  
>> What is the problem you are faced to, exactly?  
>>  
>> As already pointed out by Peter, such scheme can only work with  
>> particular equation, that has a flow of information coming from the  
>> points  $x-k$ ,  $y-m$ ,  $t-l$  with  $k,m,l > 0$ .  
>>  
>> This can't be the case e.g. for elliptic equation like the Poisson  
>> equation, because by nature the flow of information comes from every  
>> direction (hence, you need value at  $x+k,y+m$ ). Of course, you can always  
>> use an explicit discretization in time (e.g. the value at time  $t$  is  
>> computed with the values at previous times  $t-l$ ), but you will need  
>> value of  $U(x+k, y+m, t)$   
>>  
>  
>First of all, thank you both for two very helpful responses, you've  
> certainly given me something to look into! Loic, you are quite right about  
> all your remarks, the problem is that my perspective on the subject is a  
> bit different: I've been studying computer parallelization techniques for  
> iterative algorithms exhibiting the properties I described (i.e. three-  
> or more dimensional loops, with nonnegative dependencies between data  
> points, so that only previously computed points are required for the  
> current computation), where I mostly used micro-kernels or parts of real  
> applications. My study has been focused on high performance issues,  
> from the computer science point of view; I wasn't targeted to solving  
> a specific physical problem.

this is a wrong approach: parallelization makes sense only for HPC

## Re: request for algorithm

and HPC applications always require that you consider quite carefully the properties of the specific problem, the properties of your computing equipment coming second

>  
>Given this restrictions, I am now additionally looking for real, practical  
>problems, where these techniques might be applicable. It would really  
>be helpful because of all the reasons mentioned above, if all data  
>dependencies of the algorithm where nonnegative, that's why I am asking  
>for those \*particular\* cases, where this assumption holds. I have been  
>studying discretization techniques of boundary value problems (initial  
>value problems don't seem to help, as there is this flow of information  
>you describe), for which backward discretization seems to deliver a  
>differences equation consistent with the initial differential equation,  
>at least according to my poor theoretical knowledge on the subject :-)  
>  
>For example, determination of the values of U, given that  
>laplacian(U(x,y,z))=F(x,y,z), domain [0,X]\*[0,Y]\*[0,Z], known values  
>U(x,y,z)=Gxyz on the "left" domain border, so that I have one-sided  
>flow of information, discretization parameters dx,dy,dz. The iterative

no!

laplacian(U(x,y,z))=F(x,y,z), domain [0,X]\*[0,Y]\*[0,Z]  
makes no sense without specifying boundary values (Dirichlet, von Neumann, Robin) on the complete boundary. the initial value problem for the laplacian is not well posed in the sense of Hadamard (existence, uniqueness of solution and continuous dependence of the solution from the problem data)

>solution of this equation has lead me to following conclusions:

>  
>a) the convergence depends heavily on both the function F, as well as  
>the 3D problem domain.

>  
>b) generally, increasing dx, dy and dz reduces the approximation error,  
>which kind of contradicts my intuition. However, my thoughts on this  
>are not very clear, I have to look into it more thoroughly.

>  
see above: it is exponentially unstable

>Once again, thank you very much fellows, you make up for the lack of  
>interdisciplinary cooperation I am facing in the University :-) Sorry  
>for the long post.

>  
>> HTH,  
>> Loic.

what you really want: all problems (?) where "new data" depend explicitly on the "old data" and there exist of course lots of problems which can be solved in this manner, for example all explicit iteration processes, and, as mentioned already, time stepping for hyperbolic equations

Re: request for algorithm

hth  
peter

- 
- *Follow-Ups:*
    - ◆ *Re: request for algorithm*
      - ◇ *From: ndros*
  
  - *References:*
    - ◆ *request for algorithm*
      - ◇ *From: ndros*
    - ◆ *Re: request for algorithm*
      - ◇ *From: loic-dev*
    - ◆ *Re: request for algorithm*
      - ◇ *From: ndros*
  
  - Prev by Date: *Re: request for algorithm*
  - Next by Date: *Re: request for algorithm*
  - Previous by thread: *Re: request for algorithm*
  - Next by thread: *Re: request for algorithm*
  - Index(es):
    - ◆ *Date*
    - ◆ *Thread*