Numerical methods for solving PDEs
Practicum 2: two-dimensional time-evolution problems

1 Aims

This practicum has the following **general** aims:

- To learn, by means of specific model problems, to convert partial differential equations into discrete problems and to solve these problems on a computer.
- To get insight in the concept of numerical stability.
- To get insight in the concepts accuracy and numerical approximation.

The **specific** aims of this practicum are:

- To learn to (correctly) implement a simple (explicit Euler or upward scheme) for the heat equation, the wave equation, and the transport equation.
- To check the stability condition of this scheme for these three equations.
- To check the consistence of the scheme and the order of convergence for these three equations.

2 Task

We study numerical methods for the solution of the heat equation,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{1}$$

the wave equation,

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{2}$$

and the transport equation,

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\ . \tag{3}$$

As solution domain we take the unit square $\Omega = [0,1] \times [0,1]$. The boundary conditions are of the homogeneous Dirichlet-type. There is also an initial solution $u_0(x,y)$ given. As discretisation technique we consider the simplest explicit methods (e.g. explicit Euler or upwind), on an equidistant grid with mesh size h in the x- and y-directions, and k in the t-direction. The exercise or task goes as follows:

1. Implement the explicit methods in Matlab. Write your programme as compact and efficient as possible. Why do you think your programme is correct? What tests did you do to be 'sure' about the accuracy? Provide a listing. Some hints:

   – Use a matrix for representing the solution on a time level. When the grid lines are numbered from 1 to J, that matrix has dimension $J \times J$. The internal grid points correspond to the part `u(2:J-1,2:J-1)` of this matrix. The other values correspond to the boundary points.

   – It is not necessary to store the approximation for all the time steps. It suffices to store it on two or three time levels.

   – The update of the points can best be performed block-wise. A *part of the code* for an implementation of a one-dimensional explicit scheme can read as follows:

   ```
   j=2:J-1;
   for n = 1:nf,
       unew(j)= a*u(j-1)+(1-2*a)*u(j)+a*u(j+1);  u=unew;
   end
   ```

   Use a similar, compact technique for your two-dimensional explicit scheme.

   – Visualize the solution with the Matlab-commands `surf` or `pcolor`. It is also very valuable to make a movie of the time evolution. This can, e.g., be achieved with the following commands:

   ```
   figure('Renderer','zbuffer');
   for n = 1:nf,
   ...
   surf(x,y,u); axis([0 1 0 1 -1 1 -1 1]);
   M(n)= getframe;
   end
   movie(M)
   ```

   You can store the movieframes collected in matrix M as an avi-
   le as follows:

   ```
   writerObj = VideoWriter('movie_name.avi'); open(writerObj);
   writeVideo(writerObj,M); close(writerObj);
   ```

2. *Illustrate the stability limitation.* Give for every explicit method the appropriate stability limitation : what is the condition to be satisfied by h and k for the scheme to be stable? Illustrate numerically the importance of this limitation : choose two sets of values for h and k (one within the stable domain and one outside this domain) and show what happens to the solution.

3. *Investigate the accuracy.* Use as initial solution the function

$$u_0(x, y) = \sin(\pi x) \sin(\pi y) \ .$$

Give the formula for the exact solution $u(x, y, t)$. Find the numerical solution for, e.g., $t = 0.1$ and compare this to the exact solution. Choose as norm the maximal value of the error. How do you determine this value?

How does the error vary as function of h and k, with constant k/h^2 (in the case of problem (1)), or with constant k/h (in the case of equations (2) and (3)). Illustrate this numerically by means of a table.

– Hint: to evaluate a function efficiently on a grid, you can use the Matlab command `meshgrid`. For example

```
[x,y]=meshgrid((0:J-1)/(J-1),(0:J-1)/(J-1));
u = sin(pi*x).*sin(pi*y);
```

4. Take as initial solution the function

$$u_0(x, y) = 15(x - x^2)(y - y^2)e^{-50\{(x-0.5)^2+(y-0.5)^2\}} ,$$

and plot the numerical solution of each of the three equations on the time levels $t = 0.1$ and $t = 0.2$. Choose an appropriate k and h.

3 Modalities

- The *maximal length of the report is 8 pages* (incl. listing and figures).
- The report has to be handed in before 23/12/2017.
- Also this practicum will be marked and the score is part of the total score for this course. The report will be evaluated on correctness of the implementation of the schemes, the delivery of the requested results and figures, and the correctness of the interpretation. The score accounts for 2 points (out of the total of 20 for this course).
- It is allowed (if wanted) to collaborate in groups of maximal two students. In this case only one report has to be handed in and both students get the same mark.

4 Help line and feedback

In case of unclearness about the task or the modalities you can contact me for more information (my coordinates are given below). You can also contact me in case you get stuck with the implementation in Matlab.

Stefaan Poedts (200B 02.22)
Email: Stefaan.Poedts@kuleuven.be
GSM: 0486/948655