

## MATH 5620 CLASS PROJECT SUGGESTIONS

### NUMERICAL ANALYSIS II

#### 1. PROJECT GUIDELINES

- It is possible to do projects in pairs. Please inform me about this before you choose your project so that we can adjust the workload accordingly.
- Two persons (or groups) can do the same project
- A short talk (10-20min) where you will present your results to the class will be part of your project grade (20% talk, 80% write up). The time for your talk is flexible, just let me know well in advance if you will need more time.
- Please do **keep me informed on the progress** of your project.
- Please do let me know by email when you have chosen a project.
- You are **strongly encouraged** to come discuss the project with me, to make sure you are on the right track and that we agree on the outcome of the project.

#### 2. TENTATIVE SCHEDULE

- Choose a project by Mon 04/02 (but the earlier the better).
- Project reports are due the last day of classes Wed 04/25.
- Project presentations are on the last week of classes (04/23–04/25).

#### 3. SUGGESTED CLASS PROJECTS

This is by no means an exclusive list. If you are interested in a particular application, we can figure out a numerical analysis project based on it. Don't be thrown off by the name of the method, some of them are very simple. If you need any of the referenced books I own them and can let you borrow them. Also the referenced papers are from journals that our university subscribes to. Let me know if you have problems obtaining them.

1. (\*\*) Model propagation of fire using the Eikonal equation (or any other application of level-sets in [3]).
2. A simple mesh generator: [2]. In addition to understanding the method you need to come up with at least one example of your own.
3. Implement a solver for the Laplace equation using  $P_n$  finite elements ( $P_1$  will be seen in class,  $P_n$  means that the polynomials used to approximate the solution in each element are of degree  $n$ ). You need to demonstrate the efficacy of your code with at least a convergence plot.
4. Compare the ADI, LOD and Crank-Nicholson methods for solving the heat equation in 2D on the unit square.
5. Implement the Chebyshev Spectral method for solving  $u'' = f(x)$  [4, 1]
6. (\*\*) Two dimensional advection-diffusion equation with a constant velocity field. Lax-Wendroff should work well with this one.

7. (\*\*) The two dimensional wave equation. This can be done by writing the wave equation as the first order system:

$$\begin{aligned}u_t &= v \\v_t &= c^2 \Delta u,\end{aligned}$$

discretize in space using the usual second order 2D Laplacian discretization, and in time using say Euler or implicit trapezoidal rule. You may also formulate the wave equation as a multidimensional advection problem and use Lax-Wendroff, but this is more involved. If you feel you have time, you may consider the case where the propagation velocity  $c$  is non constant.

#### 4. REPORT AND TALK

What I expect is that you identify the problem (why is the method used?), tell me how the method deals with the problem and show a (simple) implementation of the method. You have to convince me in your report that you have understood the method. For the talk you would present the method to the class in a way that your class mates should be able to understand and show some numerical results.

#### REFERENCES

- [1] Randall J. LeVeque. *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007. Steady-state and time-dependent problems.
- [2] Per-Olof Persson and Gilbert Strang. A simple mesh generator in Matlab. *SIAM Rev.*, 46(2):329–345 (electronic), 2004.
- [3] J. A. Sethian. *Level set methods and fast marching methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, second edition, 1999. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.
- [4] Lloyd N. Trefethen. *Spectral methods in MATLAB*, volume 10 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.