MAP 502: Project in numerical modeling

by Thomas Wick

Basic course of the Master Program "STEEM"
Ecole Polytechnique
Academic year 2016-2017

List of projects

Nov 21, 2016

Online version on:

`http://www.cmap.polytechnique.fr/~wick/map_502_winter_2016_engl.html`

Exercise 1

(2 + 2 persons) Let the following ODE initial-value problem be given:

$$y'(t) = ay(t), \quad y(t_0) = 7 \tag{1}$$

on the time interval (t_0, T) where $t_0 = 5$ and $T = 500$. Furthermore consider four different test cases with $a = 0.1, 2, -0.1, -2$.

1. Implement the Euler method, backward Euler method, and the Crank-Nicolson method in octave (2 persons) and python (2 persons).
2. Using the backward Euler method and the Crank-Nicolson method, an implicit system arises. Formulate this system as root finding problem and formulate Newton's method (all details will be discussed in further meetings) to solve these implicit systems.
3. Recapitulate the stability regions for these three numerical schemes and compute the critical (time) step size for the Euler scheme.
4. Using different step sizes, investigate and analyze the the findings (unstability of the Euler method)
5. Recapitulate (maybe I give further materials) the convergence and order of convergence for these three schemes
6. Compute the convergence order from the numerical results and compare what the theory says (similar to the lecture notes, Section 4.7).
7. **(Optional)** Derive an error estimator based on the truncation error to adaptively determine the (time) step sizes (I will give further materials)
8. **(Optional)** Implement a higher-order Runge-Kutta method (e.g. of third order) and compare with the other schemes in terms of computational cost and accuracy (a third order method should converge very, very fast).
9. **(Optional)** Redo the implementation for the ODE:

   $$y'(t) = ay(t) + 3y(t)t^2 + t + 3$$

   Set $a = -0.1$ and compare the results of Euler, forward Euler and Crank-Nicolson.

Exercise 2

(2 + 2 persons) In this second project we consider a boundary-value problem. Let the Poisson problem in 1D be given:

$$-u''(x) = f, \quad \text{in } \Omega = (0, 1),$$
$$u(0) = u(1) = 0,$$

where $f = 1$.

1. Recapitulate the finite element (FE) method in 1D using linear splines (will be introduced in detail in the upcoming lectures).
2. **Implement** the above equation using finite elements in octave (2 persons) and python (2 persons). This task comprises several sub-tasks:
   - Write down the weak form
   - Localize the weak form on each mesh element
   - Derive the linear equation system
   - Incorporate the Dirichlet boundary conditions
   - Solve the linear system
   - Visualize the final solution
3. **Verification of correctness of the code**
   - Detect numerically the convergence order by carrying out computations on a sequence of refined meshes.
   - For a given right hand side f, construct a manufactured solution (I will explain how to do this).
   - Using this manufactured solution, compute $\|u - u_h\|_{L^2}$ where $\|\cdot\|_{L^2}$ is the L^2 norm (I will explain if not known).
4. **(Optional)** Recapitulate/derive (I will help!) the theory of FE error estimation.
5. **(Optional)** Implement quadratic polynomials for the FE method (I can help and explain).
6. Implement the previous steps for solving the PDE:

   $$-\varepsilon u''(x) + u'(x) = 1, \quad \text{in } \Omega = (0, 1),$$
   $$u(0) = u(1) = 0,$$

   where ε is a small but positive parameter, e.g., take $\varepsilon = 1, 10^{-2}, 10^{-4}$. What do you observe in the numerical results with respect to ε?

Exercise 3

Let Ω be an open, bounded subset of $\mathbb{R}^d, d = 1$ and $I := (0, T]$ where $T > 0$ is the end time value. The IBVP (initial boundary-value problem) reads: Find $u := u(x, t) : \Omega \times I \to \mathbb{R}$ such that

$$\begin{aligned} \rho \partial_t u - \nabla \cdot (\alpha \nabla u) &= f \quad \text{in } \Omega \times I, \\ u &= a \quad \text{on } \partial\Omega \times [0, T], \\ u(0) &= g \quad \text{in } \Omega \times t = 0, \end{aligned}$$

where $f : \Omega \times I \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ and $\alpha \in \mathbb{R}$ and $\rho > 0$ are material parameters, and $a \geq$ is a Dirichlet boundary condition. More precisely, g is the initial temperature and a is the wall temperature, and f is some heat source.

1. Using finite differences in time and finite elements in space, implement the heat equation in octave or python (Hint: Try to implement a general One-Step-θ scheme with $\theta \in [0, 1]$ for temporal discretization and linear finite elements for spatial discretization).
2. Set $\Omega = (-10, 10)$, $f = 0$, $\alpha = 1$, $\rho = 1$, $a = 0$, $T = 1$, and

   $$g = u(0) = \max(0, 1 - x^2).$$

   and carry out simulations for $\theta = 0, 0.5, 1$. What do you observe? Why do you make these observations?
3. Justify (either mathematically or physically) the correctness of your findings.
4. Why do you observe difficulties using $\theta < 0.5$. What is the reason and how can this difficulty be overcome?
5. Detecting the order of the temporal scheme: Choose a sufficiently fine spatial discretization (that is make the spatial discretization parameter h be sufficiently small) and compute with different time step sizes δt the value of the point $u(x_0, T) := u(x = 0; T = 1)$. Compute the error

   $$|u_{\delta t_l}(x = 0; T = 1) - u_{\delta t_{fine}}(x = 0; T = 1)|, \quad l = l_0, l_0/2, l_0/4, \ldots$$

   How does the error behave with respect to different θ?
6. Detecting the order of spatial discretization: Choose a small δt and compute a sequence of solutions for various h, e.g., $h = h_0, h_0/2, h_0/4, \ldots$. Observe again the point $u(x_0, T) := u(x = 0; T = 1)$. How does the error behave? What order of the spatial scheme do you detect?
7. **(Optional)** Implement quadratic polynomials for the spatial discretization (I can help and explain). Redo the previous exercise a see how the spatial error behaves.
8. **(Optional)** Implement the above problem in 2D spatial dimensions on e.g., $\Omega = (-2, 2) \times (-2, 2)$ and redo some of the previous tasks (up to your choice).

Exercise 4

(2+2 persons) The goal of this project is to investigate damping techniques for the elastic wave equation. These are interesting from physical and numerical/mathematical point of view. The reason is that damping (e.g., friction) is often present in a system. Mathematically, damping leads to better regularity of the solid PDE equation, which is important for theoretical analysis of fluid-structure interaction for instance.

Let $\Omega = (0,1)$ to be an open and bounded set. Furthermore, let $\partial\Omega$ be the boundary. The time interval is given by $I := (0,T]$ for some fixed end time value $T > 0$. We consider the initial/boundary-value problem: Let $f : \Omega \times I \to \mathbb{R}$ and $g, h : \Omega \to \mathbb{R}$ be given. Furthermore, let $\gamma_w, \gamma_s \geq 0$. We seek the unknown function $u : \bar{\Omega} \times I \to \mathbb{R}$ such that

$$\partial_t^2 u + Lu + \gamma_w \partial_t u + \gamma_s \partial_t Lu = f \quad \text{in } \Omega \times I, \tag{2}$$

$$u = 0 \quad \text{on } \partial\Omega \times [0,T], \tag{3}$$

$$u(0) = g \quad \text{in } \Omega \times \{t = 0\}, \tag{4}$$

$$\partial_t u(0) = h \quad \text{in } \Omega \times \{t = 0\}. \tag{5}$$

Furthermore, the linear second-order differential operator is defined by:

$$Lu := -\sum_{i,j=1}^{n} \partial_{x_j}(a_{ij}(x,t)\partial_{x_i} u) = -\nabla \cdot (a\nabla u),$$

for a given (possibly spatially and time-dependent) coefficient function a_{ij}. In 1D (one dimension) it holds:

$$Lu := -\frac{d}{dx}(a(x)u'(x)).$$

1. Give an explication/idea (mathematically and/or physically) why the first term with γ_w is called weak damping and the second one with γ_s is called strong damping.
2. Derive a mixed form of the above formulation in which the velocity $v := \partial_t u$ appears explicitly.
   Hint: Replace whenever possible $\partial_t u$ by v and create a 2nd equation: $\partial_t u - v = 0$.
3. Based on the mixed weak form of the previous question, derive a One-Step-θ scheme based on the weak mixed formulation.
4. Take the previous system and discretize now in space with the finite element method (for the specific steps see for example the previous exercise 2).
5. Numerical simulations: Implement the temporally and spatially discretized mixed formulation in 1D (one space dimension). For the numerical simulations in 1D, the boundary $\partial\Omega$ consists actually of two points: $u(0)$ and

$u(1)$. On $u(0)$ we prescribe a time-dependent non-homogeneous Dirichlet condition:

$$u(0) = g(t) = \sin(t).$$

On $u(1)$ we either fix the solution $u(1) = 0$ or we leave it free (homogeneous Neumann conditions). The right hand side is chosen as $f = 0$. The coefficient vector is given by $a = a_{ij} = 1$. The two initial conditions are $u(0) = v(0) = 0$.

1. Run simulations using the Crank-Nicolson scheme for the undamped wave equation.
2. Run simulations using backward Euler for the undamped wave equation.
3. Run simulations using Crank-Nicolson for the damped wave equation.
4. Run simulations using backward Euler for the damped wave equation.

6. Develop and analyze the following scheme: Run wave undamped equation with Crank-Nicolson and add 'from time to time' damping. What do you observe?
7. **(Difficult and optional)** We have a string and oscillate the string up and down on the left boundary. Due to this oscillation, the string will move in wave form. And these waves are transported from left to the right. At some time, these waves reach the right boundary. It is of importance which kind of boundary conditions are prescribed on this right boundary since Dirichlet conditions and Neumann conditions both will reflect these 'outgoing' waves back to the left. Question: What could be done to avoid reflection of outgoing waves?

Exercise 5

(2 persons) In this exercise we take all the very basic implementation of the FE (finite element) method for granted and work with a software that has this feature in order to tackle immediately 'more complicated' equations.

1. Install deal.II (http://www.dealii.org/)
2. Go into step-3 (Poisson's problem in 2D) and understand the introduction
3. Run step-3 and analyze the results
4. Change from $f = 1$ to something else and analyze the results
5. Change from homogeneous Dirichlet conditions to Neumann conditions and analyze the results.
6. Implement a time-stepping scheme (for example backward Euler) (is not too difficult and I will help) and run the heat equation.
7. Implement the (forward) Euler method and see whether we get again unstable solutions if the time step size is chosen too big. What is the critical time step size?
8. Perform convergence analysis in space and time of the end time point solution $u_h(T)$ where T is the final time.

Exercise 6

(2 persons) Alternatively (or in addition) to exercise 4, we consider now the Stokes system (flow of a viscous fluid) in 2D. This system has been implemented in step-22 in deal.II. However, there are other additional features which would require some more work to correctly understand the programming code.

1. Implement a time-dependent Stokes system using forward or backward Euler
2. Implement the Crank-Nicolson scheme for temporal discretization
3. Perform convergence analysis (as in exercise 4) at the end time point $\{v(t), p(T)\}$.
4. Test different finite elements and observe the results (I will help and discuss these aspects in more detail with you)
5. Write down the block system for the linear solver and explain the idea of the solution algorithm.

Exercise 7

Search for a partner and realize your own idea. It is important that we try to simplify as much we can complicated equations from other disciplines that we are still able to implement them. Here the focus should then be on computational stability: What happens if you refine or enlargen the step size? How do the results then change? Can you construct a manufactured solution for this problem? How do different methods (in case of initial-value problems for example Euler, backward Euler, and Crank-Nicolson) compare?

Evaluation (March 6, 2017: 13.00 - 18.00h): The final exam consists of

- A report (word or latex) of your task, which contains the problem statement, the numerical approach(es), set-up of the numerical example(s), analysis/interpretation of the numerical results;
- A 15-20 minutes presentation (with blackboard or beamer).

Remark to all exercises: There is no problem to search on the internet to find code snippets or to use other programs for the solution of the above projects. For me it is important that I get the impression after the exam on March 6, 2016, that you have really understood what you implemented/analyzed and how to interpret your results. The goal of this class is to teach you various numerical schemes but also to get a feeling about their differences in using them and that different numerical schemes yield different results and one has to be careful when to use which scheme.

In case you have concerns: Please write me if you have questions or concerns:

`thomas.wick@polytechnique.edu`