Thomas Wick
CMAP
Ecole Polytechnique

MAP 502: Project in numerical modeling

Basic course of the Master Program "STEEM"
Ecole Polytechnique
Academic year 2018-2019

List of projects

Oct 1, 2018

Online version on: http://www.cmap.polytechnique.fr/~wick/map_502_winter_2018_engl.html

Exercise 1

OPTIONAL: Preparation for the final exam.

Go into the lectures notes and

1. Choose one of the three codes provided in Chapter 13.
2. Copy and paste or simply re-implement the given code snippets on your own computer.
3. Try to understand what is implemented, run the code, recapitulate what I discussed in the lectures, and compare the results with what I presented in the oral lectures and lecture notes in the corresponding sections.
4. Interprete your findings and recapitulate (with the help of my lectures and the lecture notes) why they are as they are.
5. Play with the codes and change certain parameters, values or numbers and see how the results change. Why do they change in that way?

- **The final choice of the subsequent projects and groups will be made in October. Each group of two (or three) must choose one of the following projects (Exercises).**
- **Please see in particular Exercise 7 for those who have their own ideas motivated through classes in the past or other classes in their current curriculum.**
- **For those who have already specific wishes of Exercises, please contact me.**

Exercise 2

Let the following ODE initial-value problem be given:

$$y'(t) = a(y(t) - g(t)) + g'(t), \qquad y(t_0) = y_0 \tag{1}$$

on the time interval (t_0, T) where $t_0 = 0$ and $T = 100$. In addition, let $g(t) = t$ and $y_0 = 1$. Furthermore consider four different test cases with $a = -1, -10, -100$.

1. Implement the Euler method, backward Euler method, and the Crank-Nicolson method in octave or python.
2. Recapitulate the stability regions for these three numerical schemes and compute the critical (time) step size for the (forward) Euler scheme.
3. 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.
4. Using different (time) step sizes, investigate and analyze the findings (instability of the Euler method)
5. Compute the convergence order from the numerical results and compare what the theory says (similar to the lecture notes; please see Computational Convergence Analysis).
6. Take $g(t) = t^2$ and re-do some of the previous calculation.
7. (OPTIONAL): Implement time step control (materials would be given) in which the time step sizes are chosen according to an error estimator resulting in a non-uniform, but problem-adapted, temporal discretization.

Exercise 3

Consider a basic model for the parachute problem in which the position x or the velocity v is sought. This models the situation of a skydiver subject to gravitational forces and air resistance. One possible model is a nonlinear IVP: Find v for $0 \le t \le T$, where $T > 0$ is the end time value, such that

$$mv'(t) = -mg + kv^2, \qquad (2)$$

$$v(0) = 0, \qquad (3)$$

where m is the mass of the skydiver, g the gravity with $g \approx 9.81 m/s^2$ and $k > 0$ is the force due to air resistance. Specifically, k highly depends on the different stages of a jump. Here, further materials and discussions will be made once a first version of the code has been developed. For the first tests, we use $k = 0.5 * \rho * 1.95 b_0$ with $\rho = 1 kg/m^3$ and $b_0 = 0.5 m^2$.

1. Make a dimension check whether all the above equation makes sense from the physical point of view.
2. Choosing $T = 10s$, implement the Euler method, backward Euler method, and the Crank-Nicolson method in octave or python.
3. 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 according to Chapter 10 of the lecture notes.
4. Using different step sizes, investigate and analyze the findings and compute the convergence order with the help of the formulas given in the lecture notes.

Exercise 4

In this 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 or python (or another open-source software). 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 (for the general procedure, please see the lecture notes).
   - Use the value of the right hand side f to re-run your code. Evaluate the value of u_h at $x = 0.5$ (here h indicates that u is obtained by the numerical method. In general h is the so-called discretization parameter). Compare this value to the exact value, which is obtained by evaluating the manufactured solution: $u(0.5)$.
   - Perform a quantitative convergence analysis via:

     $$|u_h(0.5) - u(0.5)|$$

     for various values for h (i.e., different meshes). Perform the computational convergence analysis as discussed in the lecture (see the lecture notes).
4. 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 5

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. **(Optional)** 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. **(Optional)** 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?

Exercise 6

Develop a Newton scheme in $\mathbb{R}^2$ to find the root of the problem:

$$f : \mathbb{R}^2 \to \mathbb{R}^2, \quad f(x, y) = \Big(2xay^2, 2(x^2 + \kappa)ay\Big)^T,$$

where $\kappa = 0.01$ and $a = 5$.

1. Justify first that integration of f yields $F(x, y) = (x^2 + \kappa)ay^2$. What is the relation between f and F?
2. Compute the root of f by hand. Derive the derivative f' and study its properties.
3. Finally, design the requested Newton algorithm. As initial guess, take $(x_0, y_0) = (4, -5)$.
4. What do you observe with respect to the number of Newton iterations?
5. How could we reduce the number of Newton steps?
6. Implement a simplified Newton scheme (i.e., the matrix is only build at the beginning or just at every other step). How does the number of iterations change?

Exercise 7

Search for a partner in class and realize your own idea. This was done by some of the previous students, for example for a load-flow analysis problem in winter 2016/2017, which finally led to a nonlinear system of equations, which we simplified in a proper way and which was then solved using Newton's method; Chapter 10 of the lecture notes.

- 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.
- Please contact me as early as possible if you have your own idea.
- The focus should then be on computational aspects:
- What happens if you refine or enlarged 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 (December 2018: 8.30 - 13.00h):

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/PowerPoint).

Remarks to all exercises:

- The above exercises should not be understand as in a typical exam; namely that ALL tasks need to be worked through!
- Each group chooses their exercise and works through the first tasks.
- Then we decide TOGETHER which further tasks shall be done.
- In the past, often own ideas went into the exercises and each group had a final individual character even if they started originally from the same exercise.
- 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 during the exam that you have really understood what you implemented/analyzed and how you can 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 or questions, please write me an email: thomas.wick@polytechnique.edu

This offer holds in particular true if you have specific questions of whatever kind to the lecture or to your chosen project.