MATH 437 Homework 8 (20 points)

1. (5 points) Consider the following parabolic PDE:

$$\partial_t u - 0.5\partial_x^2 u = x^2$$

for $t > 0$ and $x \in (0,1)$ with initial condition $u(x,0) = 0$ and boundary conditions $u(0,t) = u(1,t) = 0$. Solve this PDE using forward Euler in time and centered differences in space. Let $h = 0.1$ denote the spatial mesh size and $\tau = 0.01$ denote the timestep size. Solve until final time $T = 0.1$ and plot the solution at the final time.

Solution. Let $t_n := n\tau$ and $x_i := ih$ denote the discrete time and space points. Let u_i^n denote the discrete approximation to $u(x_n, t_i)$. The finite difference equation at t_n and x_i reads:

$$\frac{u_i^{n+1} - u_i^n}{\tau} - 0.5\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = x_i^2.$$

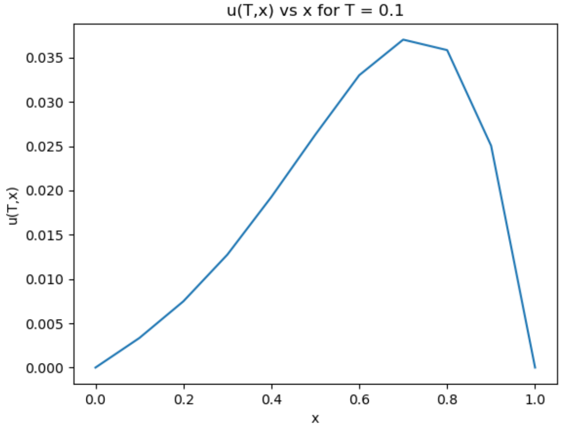
This holds for all discrete time points and all *interior* space points. At the boundary points x_0 and x_M, we impose the constraints

$$u_0^n = u_M^n = 0$$

for all $n > 0$. For the initial condition, we impose

$$u_i^0 = 0$$

for all i. At each timestep, we use the finite difference equation to compute u_i^{n+1}. See `problem_1.py` for an implementation.



$\square$

2. (5 points) Consider the following boundary-value problem

$$-y'' + y = x$$

with $y(0) = y(1) = 0$. Solve this problem using the finite element method with mesh size $h = 1/3$. Report the values (x_j, y_j) of the solution, where $x_j = ih$ and y_j is the approximation to $y(x_j)$.

Solution. We let $\{\phi_j\}_{j=1,2}$ be the standard piecewise linear finite element basis using the interior grid points $x_1 = 1/3$, $x_2 = 2/3$. We expand the finite-element solution y_h in this basis:

$$y_h = y_1\phi_1 + y_2\phi_2,$$

where y_j is the coefficient with respect to ϕ_j, which happens to approximate the exact solution $y(x_j)$ due to the properties of the ϕ_j. By replacing y with y_h in the PDE, multiplying by ϕ_i, and integrating by parts:

$$\int_0^1 y_h'\phi_i' + y_h\phi_i \, dx = \int_0^1 x\phi_i \, dx$$

for $i = 1, 2$. Notice that the boundary conditions vanish. Expanding y_h in the basis representation, we arrive at the following linear system

$$A\vec{y} = \vec{b},$$

where

$$A_{i,j} := \int_0^1 \phi_j'\phi_i' + \phi_j\phi_i \, dx, \qquad b_i := \int_0^1 x\phi_i \, dx.$$

The formula for ϕ_j is

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{h}, & x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{h}, & x \in [x_j, x_{j+1}] \\ 0, & \text{else} \end{cases}.$$

Therefore, by using a calculator to assist the computations,

$$A = \begin{bmatrix} 6.22222222 & -2.94444444 \\ -2.94444444 & 6.22222222 \end{bmatrix}, \qquad \vec{b} = \begin{bmatrix} 0.11111111 \\ 0.22222222 \end{bmatrix}.$$

Solving yields

$$\vec{y} = \begin{bmatrix} 0.04478685156651258 \\ 0.0569080636877247 \end{bmatrix}.$$

See `problem_2.py` for an implementation. □

3. Consider the following boundary-value problem

$$y'' + 2y^3 = 1$$

with $y(0) = y(1) = 0$. Let $h = 1/3$.

(a) (3 points) Write the second-order centered finite difference method for this equation.

(b) (2 points) Write Newton's method for the resulting linear system and perform 2 iterations starting from an $\vec{y}^{(0)} = \vec{0}$.

(a) *Solution.*

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + 2y_i^3 = 1 \qquad i = 1, 2$$
$$y_0 = y_3 = 0.$$

. □

(b) *Solution.* Let

$$\vec{F}(\vec{y}) = \begin{bmatrix} \frac{y_2 - 2y_1}{h^2} + 2y_1^3 - 1 \\ \frac{-2y_2 + y_1}{h^2} + 2y_2^3 - 1 \end{bmatrix},$$

so that solving the system in part a is equivalent to solving $\vec{F}(\vec{y}) = \vec{0}$. Newton's method then reads

$$\vec{y}^{(n+1)} = \vec{y}^{(n)} - \nabla\vec{F}(\vec{y}^{(n)})^{-1}\vec{F}(\vec{y}^{(n)}),$$

where

$$\nabla\vec{F}(\vec{y}) = \begin{bmatrix} -2/h^2 + 6y_1^2 & 1/h^2 \\ 1/h^2 & -2/h^2 + 6y_2^2 \end{bmatrix}.$$

Using a computer to perform the iterations, starting with $\vec{y}^{(0)} = \vec{0}$:

$$\vec{y}^{(1)} = \begin{bmatrix} -0.11111111 \\ -0.11111111 \end{bmatrix}, \qquad \vec{y}^{(2)} = \begin{bmatrix} -0.11141847 \\ -0.11141847 \end{bmatrix}.$$

See `problem_3.py` for an implementation. ☐

4. (a) (3 points) Use the continuation method to solve

$$\begin{aligned} x_1 + x_2^2 - 6 &= 0, \\ -x_1 + x_2 - 1 &= 0, \end{aligned}$$

with initial condition $x = (0, 0)$. Write down the differential equation for the solution.

(b) (2 points) Perform 5 iterations of the forward Euler method on the differential equation from the previous step with $\tau = 0.2$.

(a) *Solution.* We let

$$\vec{F}(\vec{x}) := \begin{bmatrix} x_1 + x_2^2 - 6 \\ -x_1 + x_2 - 1 \end{bmatrix}.$$

Then, the continuation method considers the function

$$\vec{G}(\lambda, \vec{x}) := \lambda F(\vec{x}) + (1 - \lambda)(\vec{F}(\vec{x}) - \vec{F}(\vec{0})).$$

We now suppose that there is a smooth function $\vec{x}(\lambda)$ such that

$$\vec{G}(\lambda, \vec{x}(\lambda)) = \vec{0}$$

for all λ. In particular, this implies that $\vec{G}(1, \vec{x}(1)) = \vec{F}(\vec{x}(1)) = \vec{0}$, so $\vec{x}(1)$ is a solution to the original problem. We also observe that $\vec{G}(0, \vec{x}(0)) = \vec{F}(\vec{x}(0)) - \vec{F}(\vec{0}) = \vec{0}$, which is trivially solved by requiring $\vec{x}(0) = \vec{0}$. Now, if we differentiate with respect to λ:

$$\begin{aligned} \vec{0} &= \frac{d}{d\lambda}\vec{G}(\lambda, \vec{x}(\lambda)) \\ &= \partial_\lambda \vec{G}(\lambda, \vec{x}(\lambda)) + \nabla_{\vec{x}}\vec{G}(\lambda, \vec{x}(\lambda))\vec{x}'(\lambda), \end{aligned}$$

where $\nabla_{\vec{x}}\vec{G}$ denotes the Jacobian matrix of $\vec{G}$ with entries

$$(\nabla_{\vec{x}}\vec{G})_{i,j} := \partial_{x_j} G_i.$$

Solving, we see that $\vec{x}(\lambda)$ satisfies the following ODE system:

$$\vec{x}'(\lambda) = -\nabla_{\vec{x}}\vec{G}(\lambda, \vec{x}(\lambda))^{-1}\partial_\lambda \vec{G}(\lambda, \vec{x}(\lambda))$$

with initial condition $\vec{x}(0) = \vec{0}$. Solving this ODE with a numerical method that gives an approximation to $\vec{x}(1)$ gives us an approximate solution to $\vec{F}(\vec{x}) = \vec{0}$. For this particular problem:

$$\begin{aligned} \partial_\lambda \vec{G}(\lambda, \vec{x}) &= \vec{F}(\vec{0}) = \begin{bmatrix} -6 \\ -1 \end{bmatrix}, \\ \nabla_{\vec{x}}\vec{G}(\lambda, \vec{x}) &= \nabla\vec{F}(\vec{x}) = \begin{bmatrix} 1 & 2x_2 \\ -1 & 1 \end{bmatrix}. \end{aligned}$$

Thus, the ODE to solve is

$$\begin{aligned} \vec{x}' &= -\nabla\vec{F}(\vec{x})^{-1}\vec{F}(\vec{0}) \\ &= -\begin{bmatrix} 1 & 2x_2 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -6 \\ -1 \end{bmatrix} \end{aligned}$$

with $\vec{x}(\vec{0}) = \vec{0}$. ☐

(b) *Solution.* The forward Euler method for the ODE from the previous step reads

$$\begin{aligned} \vec{x}^{n+1} &= \vec{x}^n - \tau \nabla \vec{F}(\vec{x})^{-1} \vec{F}(\vec{0}) \\ &= \vec{x}^n - \tau \begin{bmatrix} 1 & 2x_2^n \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -6 \\ -1 \end{bmatrix}. \end{aligned}$$

Using a computer to assist the computations, we compute 5 iterations. See `problem_4.py` for an implementation.

```
1 [1.2 1.4]
2 [1.36842105 1.76842105]
3 [1.47700574 2.07700574]
4 [1.54863882 2.34863882]
5 [1.59437022 2.59437022]
```

□