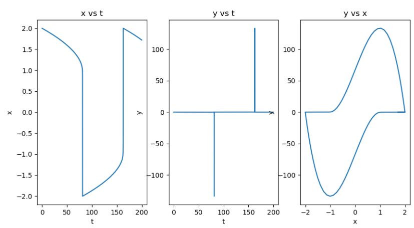
Scientific Computing Final Exam

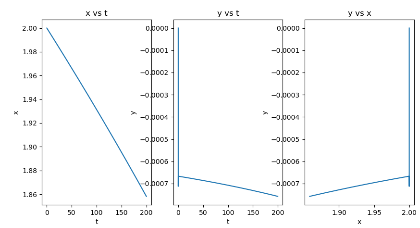
Ryan Chen

May 16, 2024

1. Code: `https://github.com/RokettoJanpu/scientific-computing-2-redux/blob/main/final%201.ipynb`

   (a) The solution for $\mu = 10^2$ is on the left and the one for $\mu = 10^3$ on the right.





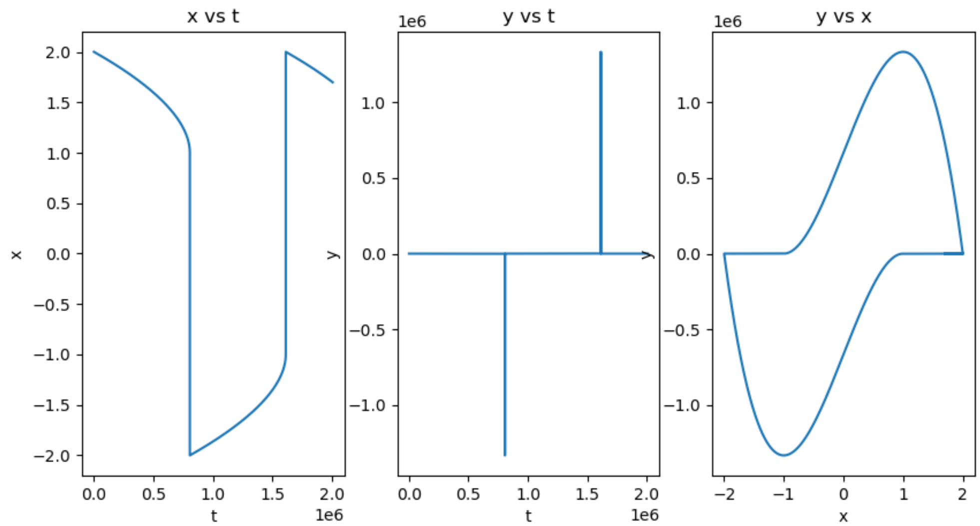
   (b) From the Butcher array, $\hat{b} = \begin{bmatrix} 1-\gamma \\ \gamma \end{bmatrix}$ and $c = \begin{bmatrix} \gamma \\ 1 \end{bmatrix}$. Pick $b := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Then the method using b instead of $\hat{b}$ is order 1 since $b_1 + b_2 = 1$ but not order 2 since $b \cdot c = \gamma \neq \frac{1}{2}$. The error estimate is

$$e := h \sum_{q=1}^{2} (b - \hat{b})_q k_q = h(\gamma k_1 - \gamma k_2) = h\gamma(k_1 - k_2) \implies \|e\| = h\gamma \|k_1 - k_2\|$$

   The following adaptive time step algorithm repeatedly multiplies or divides the time step by 2 to get as close as possible to satisfying the step acceptance criterion.

   err ← h*gamma*norm(k1-k2)
   tol ← atol + rtol*norm([x,y])
   if err < tol **then**
      while err < tol **do**
         h ← 2*h
         compute k1 and k2 using h
         err ← h*gamma*norm(k1-k2)
      end while
   end if
   if err > tol **then**
      while err > tol **do**
         h ← 0.5*h
         compute k1 and k2 using h
         err ← h*gamma*norm(k1-k2)
      end while
   end if

   Using the adaptive time step algorithm for $\mu = 10^6$ and $T_{\max} = 2 \cdot 10^6$, the CPU time is 409s.



2. (a) As a preliminary, recall a "generalized" divergence theorem: for all scalars $\varphi \in C^1(\Omega, \mathbb{R})$ and vectors $F \in C^1(\Omega, \mathbb{R}^2)$,

$$\int_\Omega \boldsymbol{\nabla}\varphi \cdot F dx = -\int_O \boldsymbol{\nabla} \cdot F dx + \int_{\partial\Omega} \varphi F \cdot n ds$$

Let $P(x,y) := e^{-\beta V(x,y)} M(x,y)$. Pick $u_D \in C^2(\mathbb{R}^2)$ such that $u_D = 0$ on ∂A, $u_D = 1$ on ∂B, and $u_D = 0$ outside some neighborhood of ∂B disjoint from Γ_N. We obtain a BVP for $v := u - u_D$,

$$\boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} v) = -\boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} u_D), \quad v = 0 \text{ on } \Gamma_D := \partial A \cup \partial B, \quad \frac{\partial v}{\partial n} = 0 \text{ on } \Gamma_N$$

Now fix $w \in C^1(\Omega)$ such that $w = 0$ on Γ_D, multiply the PDE for v and integrate over Ω.

$$\int_O w \boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} v) dx = -\int_O w \boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} u_D) dx$$

This equation, along with the generalized divergence theorem for $\varphi := w$ and $F := P\boldsymbol{\nabla} v$, gives

$$\int_\Omega (P \boldsymbol{\nabla} w) \cdot \boldsymbol{\nabla} v dx = \int_\Omega w \boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} u_D) dx + \int_{\Gamma_D} w (P \boldsymbol{\nabla} v) \cdot n ds + \int_{\Gamma_N} w (P \boldsymbol{\nabla} v) \cdot n ds$$

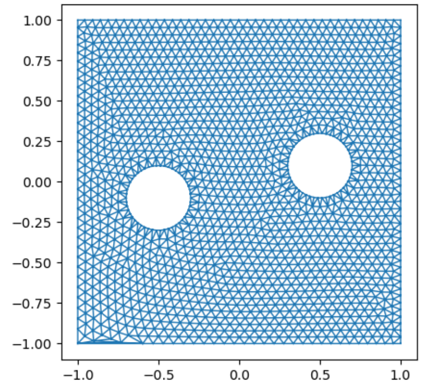
On the RHS, the second term vanishes since $w = 0$ on Γ_D, and the third term vanishes since $\frac{\partial v}{\partial n} = 0$ on Γ_N. The generalized divergence theorem for $\varphi := w$ and $F := P\boldsymbol{\nabla} u_D$ gives

$$\int_\Omega \boldsymbol{\nabla} w \cdot (P \boldsymbol{\nabla} u_D) dx = -\int_\Omega w \boldsymbol{\nabla} \cdot (P \boldsymbol{\nabla} u_D) dx + \int_{\Gamma_D} w (P \boldsymbol{\nabla} u_D) \cdot n ds + \int_{\Gamma_N} w (P \boldsymbol{\nabla} u_D) \cdot n ds$$

On the RHS, the second term vanishes since $w = 0$ on Γ_D, and the third term vanishes since $u_D = 0$ on Γ_N. Combining the last two equations gives the integral equation formulation.

$$\int_\Omega \boldsymbol{\nabla} w \cdot (P \boldsymbol{\nabla} v) dx = -\int_\Omega \boldsymbol{\nabla} w \cdot (P \boldsymbol{\nabla} u_D) dx, \quad P(x,y) = e^{-\beta V(x,y)} M(x,y)$$

(b) Triangulation of Ω:



(c) Assuming the notation of FEM, let $\{\tau_k\}_{k=1}^{N_{\text{tri}}}$ be the above triangulation of Ω, let $w = \eta_i$ for $i \in I_{\text{free}}$, and write $v = \sum_{j \in I_{\text{free}}} v_j \eta_j$. The LHS of the integral formulation becomes

$$\sum_{j \in I_{\text{free}}} \left[\underbrace{\sum_{k=1}^{N_{\text{tri}}} \int_{\tau_k} \boldsymbol{\nabla} \eta_i \cdot (P \boldsymbol{\nabla} \eta_j) dx}_{A_{ij}} \right] v_j$$

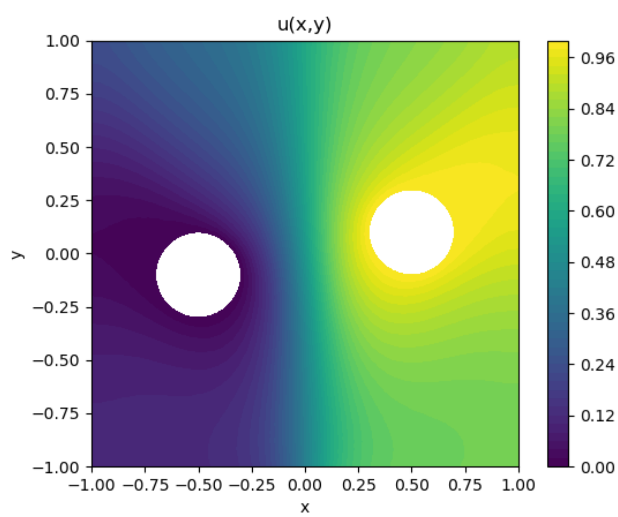
The stiffness matrix entries A_{ij} can be found by modifying the stima3 function in FEM_TPT.py,

$$0.5 D G P(v_{\text{mid}}) G^T$$

where $P(v_{\text{mid}})$ is the function P evaluated at the midpoint of the triangle τ_k, and

$$D = \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_{i+1} & x_{i+2} \\ y_i & y_{i+1} & y_{i+2} \end{vmatrix}, \quad G = \begin{bmatrix} (\boldsymbol{\nabla} \eta_i)^T \\ (\boldsymbol{\nabla} \eta_{i+1})^T \\ (\boldsymbol{\nabla} \eta_{i+2})^T \end{bmatrix}$$

Code: `https://github.com/RokettoJanpu/scientific-computing-2-redux/blob/main/final%202.ipynb`



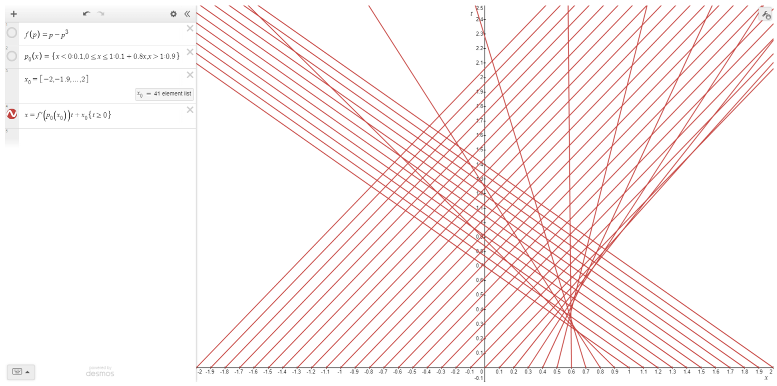
3. (a) Rewrite the PDE as

$$\rho_t + [f(\rho)]_x = 0, \quad f(\rho) := \rho v(\rho) = \rho - \rho^3$$

Characteristics are given by

$$x(t) = f'(\rho_0(x_0))t + x_0, \quad f'(\rho) = 1 - 3\rho^2$$

They are plotted below.



(b) The breaking time, when the first shock occurs, is

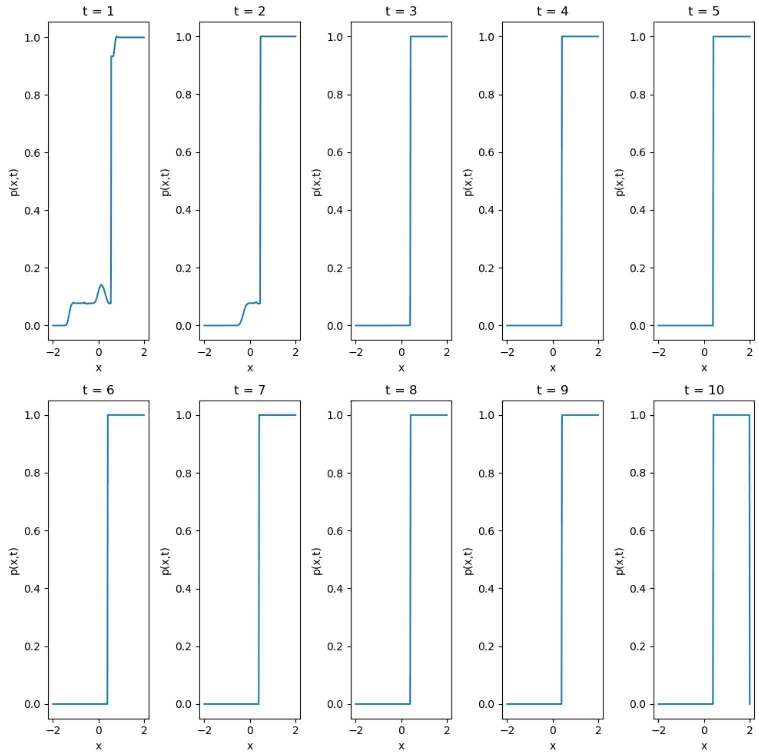
$$T_b = -\left[\min_{\xi} f''(\rho_0(\xi))\rho_0'(\xi)\right]^{-1} \approx 0.231$$

(c) The equation for ρ_0 gives $\rho_L = 0.1$ and $\rho_R = 0.9$, so the eventual shock speed is

$$s = \frac{f(\rho_L) - f(\rho_R)}{\rho_L - \rho_R} = 0.09$$

(d) The numerical flux is

$$F(\rho_L, \rho_R) = \begin{cases} \min_{\rho_L \le \rho \le \rho_R} f(\rho), & \rho_L \le \rho_R \\ \max_{\rho_R \le \rho \le \rho_L} f(\rho), & \rho_L > \rho_R \end{cases}$$


t = 1
t = 2
t = 3
t = 4
t = 5
t = 6
t = 7
t = 8
t = 9
t = 10
p(x,t)
x