Lab Assignment 14
Semi-Lagrangian Methods and Strang Splitting

ACS II

Spring 2020

Assigned April 16, 2020
Due April 23, 2020

Semi-Lagrangian Methods

Consider the 1D linear advection equation

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(u(x,t)c) = 0,$$

where $u(x,t)$ is a given velocity field, and $c(x,t)$ is the concentration of some species. This can be solved through a variety of finite difference/finite element approaches. There also exists an alternative approach to this problem, where one seeks to track not the flow in/out of a control volume, but the path of a specific fluid parcel in time. This gives rise to the semi-Lagrangian approach. The semi-Lagrangian approach is split up into two components for each time-step.

- For each node i at time step k, integrate back along the characteristic curve to time step $k-1$, identifying the location of the fluid parcel that ends up at node i, x_i^{k-1}
- Interpolate for the scalar quantity $c(x_i^{k-1}, t^k)$

The integration step can be accomplished by any ODE integrator, such as forward Euler or any scheme in the Runge-Kutta method. The interpolation step can be handled by a polynomial interpolant.

Data: number of intervals N, time step size Δt, time t_k, functions $u(x,t)$, $c^k(x)$
Result: $c^{k+1} = c(x, t + \Delta t)$
initialize :
$x \leftarrow$ linspace(0,1,N);
for *each node* x_i **do**
  solve $x_i'(t) = -u(x_i, t^k)$ for $x_i^{k-1} = x_i(t - \Delta t)$;
  interpolate to find $c^k(x_i^{k-1}, t^k)$;
  $c_i^{k+1} \leftarrow c^k(x_i^{k-1}, t^k)$
end

Algorithm 1: Algorithm for a single time step of the semi-Lagrangian method.

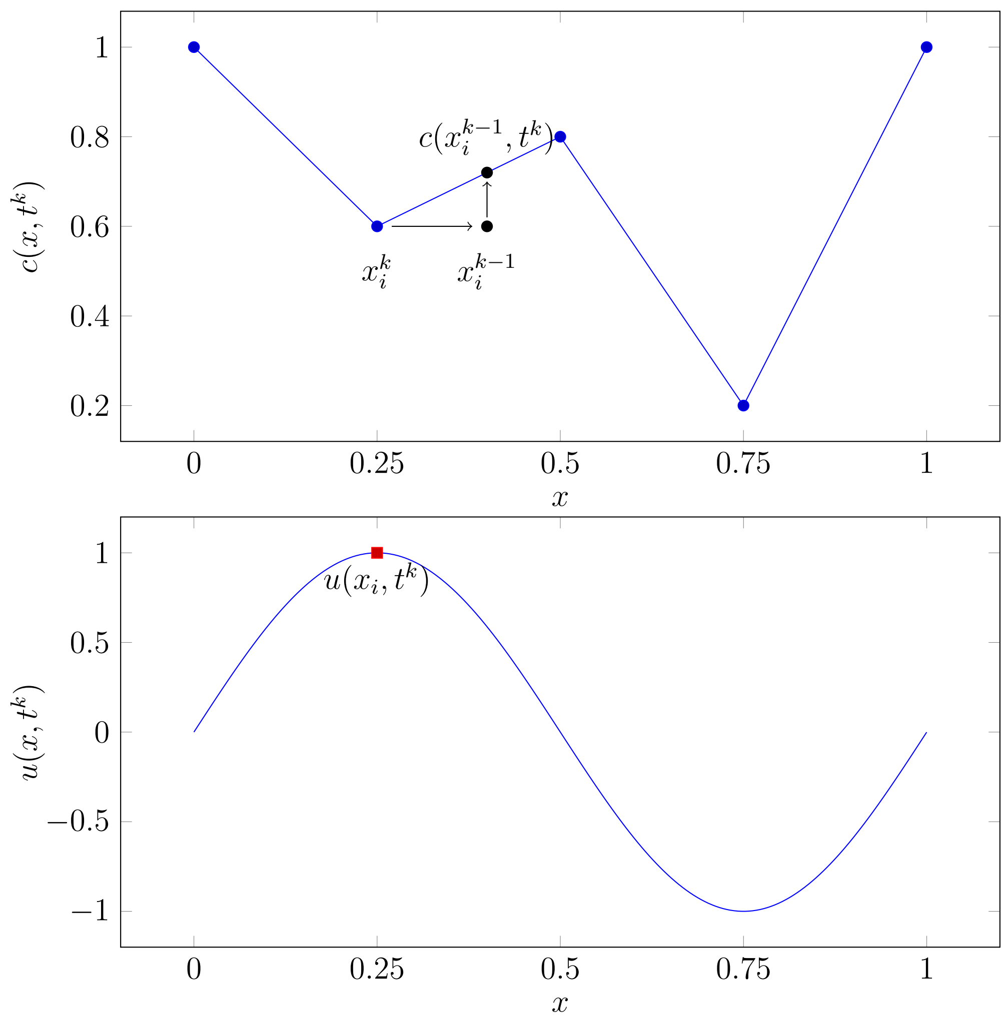

Figure 1: Top: $c(x, t^k)$. Bottom: given $u(x, t^k)$.

Exercise I

For the 1D linear advection problem

$$\frac{\partial c}{\partial t} + u(x,t)\frac{\partial c}{\partial x} = 0, \qquad 0 \le x \le 1,\ 0 \le t \le 8$$

Implement the semi-Lagrangian method for this problem with $c_0(x) = \sin^2(\pi x)$, $u(x,t) = 1$ and with periodic boundary conditions. Use the forward Euler method with $\Delta t = 0.01$ for the integration step and a linear interpolation scheme for the interpolation step. Estimate the spatial order of accuracy by doing a grid order convergence study, and discuss the observed accuracy of the method.

Operator Splitting

Consider the advection-diffusion equation

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(u(x,t)c) - \alpha\frac{\partial^2 c}{\partial x^2} = 0.$$

One way to solve this is to use a combination of methods discussed earlier in the course. In particular we will use centered in space-backward in time finite difference solve to solve the diffusion equation $c_t - c_{xx} = 0$. In the above section of this lab you've solved the advection equation $c_t + (u(x,t)c)_x = 0$ using the semi-Lagrangian method. In a seminal 1968 paper, Gilbert Strang introduced the notion of operator splitting, where separate operators in a full PDE are decoupled and solved separately, with there contributions from each being added together. For the advection-diffusion equation, letting $\mathcal{L}_A(c) = (u(x,t)c)_x$ and $\mathcal{L}_D(c) = -\alpha c_{xx}$, the second-order operator splitting method is given as.

- Solve $c_t + \mathcal{L}_A(c)$ for $\Delta t/2$
- Solve $c_t + \mathcal{L}_D(c)$ for Δt
- Solve $c_t + \mathcal{L}_A(c)$ for $\Delta t/2$

Centered in Space-Backwards in Time

To solve the diffusion equation, you will implement a second order centered finite difference solver in space and combine it with first order backwards Euler in time. Recall that on a standard uniform grid with spacing h, we can discretize c_{xx} as

$$c_{xx}(x_i) \approx \frac{C_{i-1} - 2C_i + C_{i+1}}{h^2}. \tag{1}$$

Term c_t can be discretized using a first order Euler approximation,

$$c_t(x_i) \approx \frac{C^{k+1} - C_i^k}{\Delta t}. \tag{2}$$

Combining (2) and using the implicit time level $k+1$ in (1), gives

$$-\alpha \Delta t C_{i-1}^{k+1} + (2\alpha \Delta t + h^2) C_i^{k+1} - \alpha \Delta t C_{i+1}^{k+1} = h^2 C_i^k. \tag{3}$$

Ignoring boundary conditions for the time being, this is a tridiagonal system that must be solved at each time step,

$$\begin{pmatrix} 1+2\lambda & -\lambda & & & \\ -\lambda & 1+2\lambda & -\lambda & & \\ & & \ddots & & \\ & & -\lambda & 1+2\lambda & -\lambda \\ & & & -\lambda & 1+2\lambda \end{pmatrix} \cdot \begin{pmatrix} C_0^{k+1} \\ C_1^{k+1} \\ \vdots \\ C_{N-1}^{k+1} \\ C_N^{k+1} \end{pmatrix} = \begin{pmatrix} C_0^k \\ C_1^k \\ \vdots \\ C_{N-1}^k \\ C_N^k \end{pmatrix},$$

where $\lambda = \alpha \Delta t / h^2$. This system can be solved efficiently using the Thomas algorithm. If we have periodic boundary conditions however, the system is no longer tridiagonal,

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & \dots & -\lambda \\ -\lambda & 1+2\lambda & -\lambda & & \\ 0 & & \ddots & & 0 \\ & & -\lambda & 1+2\lambda & -\lambda \\ -\lambda & 0 & \dots & -\lambda & 1+2\lambda \end{pmatrix} \cdot \begin{pmatrix} C_0^{k+1} \\ C_1^{k+1} \\ \vdots \\ C_{N-1}^{k+1} \\ C_N^{k+1} \end{pmatrix} = \begin{pmatrix} C_0^k \\ C_1^k \\ \vdots \\ C_{N-1}^k \\ C_N^k \end{pmatrix}. \tag{4}$$

We can no longer apply the Thomas algorithm directly, however we can still solve this system in $O(N)$ operations using the Sherman-Morrison formula:

$$(A + \mathbf{u} \otimes \mathbf{v})^{-1} = A^{-1} - \frac{(A^{-1}\mathbf{u}) \otimes (\mathbf{v}^T A^{-1})}{1 + \mathbf{v}^T A^{-1} \mathbf{u}}. \tag{5}$$

We are trying to solve the system $(A + \mathbf{u} \otimes \mathbf{v})\mathbf{x} = \mathbf{b}$. Let $\mathbf{y} = A^{-1}\mathbf{b}$ and $\mathbf{z} = A^{-1}\mathbf{u}$. Then

(5) gives:

$$\begin{aligned}
\mathbf{x} &= \left(A^{-1} - \frac{(A^{-1}\mathbf{u}) \otimes (\mathbf{v}^T A^{-1})}{1 + \mathbf{v}^T A^{-1}\mathbf{u}} \right) \mathbf{b} \\
&= A^{-1}\mathbf{b} - \frac{(A^{-1}\mathbf{u}) \otimes (\mathbf{v}^T A^{-1})}{1 + \mathbf{v}^T A^{-1}\mathbf{u}} \mathbf{b} \\
&= \mathbf{y} - \frac{\mathbf{z}(\mathbf{v} \cdot \mathbf{y})}{1 + \mathbf{v} \cdot \mathbf{z}}
\end{aligned}$$

The algorithm to compute $\mathbf{x}$ proceeds as follows:

1. solve $A\mathbf{y} = \mathbf{b}$ using Thomas algorithm
2. solve $A\mathbf{z} = \mathbf{u}$ using Thomas algorithm
3. $\mathbf{x} = \mathbf{y} - \mathbf{z}(\mathbf{v} \cdot \mathbf{y})/(1 + \mathbf{v} \cdot \mathbf{z})$

It remains to define $\mathbf{u}$, $\mathbf{v}$ and A. The matrix in (4) can be replaced by $A + \mathbf{u} \otimes \mathbf{v}$, where

$$A = \begin{pmatrix} 2\lambda & -\lambda & & & \\ -\lambda & 1+2\lambda & -\lambda & & \\ & & \ddots & & \\ & & -\lambda & 1+2\lambda & -\lambda \\ & & & -\lambda & 1+2\lambda-\lambda^2 \end{pmatrix}, \qquad \mathbf{v} = \mathbf{u} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -\lambda \end{pmatrix}.$$

Exercise II

Implement the Strang splitting method using semi-Lagrangian for the advection term and centered space, backwards in time for the diffusion term to solve the advection-diffusion equation

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(u(x,t)c) - \alpha \frac{\partial^2 c}{\partial x^2} = 0, \qquad 0 \le x \le 1,\ 0 \le t \le 8,$$

with $c_0(x) = \sin^2(\pi x)$, $u(x,t) = 1$, $\alpha = 0.01$, with periodic boundary conditions. Plot your results at $t = 0,\ 2,\ 4,\ 8$ and discuss the behavior of your solution.

Submission and Grading

To get credit for this assignment, you must upload the following information to Canvas 11:59pm, April 23, 2020

- your source code files
- Your report as a single file in pdf format, including results from your work and relevant discussion of your observations, results, and conclusions.

This information must be received by 11:59pm, April 23, 2020. Upload the required documents to Canvas. As stated in the course syllabus, late assignment submissions will be subject to a 10% point penalty per 24 hours past the due date at time of submission, to a maximum reduction of 50%, according to the formula:

*[final score] = [raw score] - min(0.5, 0.1 * [# of days past due]) * [maximum score]*