MATH 934 – IMPLICIT/EXPLICIT METHODS

INSTRUCTOR: DR. ADAM LARIOS

Consider the viscous Burgers equation in 1D:

$$\begin{cases} u_t + uu_x = \nu u_{xx}, \\ u(x,t_0) = u_0(x). \end{cases} \tag{1}$$

with periodic boundary conditions on an interval of length L. Here, $\nu > 0$ is the viscosity, $f = f(x,t)$ is a *force* or *source-term*. The function u_0 is the initial data, which we will assume to be a periodic, square-integrate function, i.e., $\int_0^L |u_0(x)|^2\,dx < \infty$ (normally, this won't be a big issue in our computations).

A fully explicit method for solving an equation with a diffusion term νu_{xx} suffers from a viscous CFL restriction on the time step; namely, $\Delta t \leq C(\Delta x)^2/\nu$ is required for stability with a fully explicit approach. We might consider using an implicit method, but since Burgers equation is nonlinear, this would require a very expensive nonlinear solve each time step (e.g., Newton's method). Can we find a mixture of implicit and explicit methods to hopefully extract the best of both worlds? That is, keep the nonlinear term explicit, but make the linear diffusion term implicit?

To try to understand stability of our algorithms, we replace Burgers equation by the following linear transport-diffusion equation, for a constant speed $c > 0$,

$$u_t + cu_x = \nu u_{xx}. \tag{2}$$

This equation is linear, so we could handle the whole equation with an implicit method. However, since we are thinking of it as a model for Burgers equation, we will keep the transport term cu_x explicit. (If u is bounded, say by $|u| \leq M$, we can also think about this as replacing uu_x by the worst-case scenario Mu_x.)

As with the heat equation, if we formally express the solution and the forcing via its Fourier series, we have:

$$u(x,t) = \sum_k \widehat{u}_k(t) e^{ik\frac{2\pi x}{L}}.$$

This can be used to (formally) transform Burgers equation into the following system of coupled nonlinear ODEs

$$\frac{d}{dt}\widehat{u}_k = -\nu k^2 \widehat{u}_k - ikc\widehat{u}_k, \qquad \text{for each } k. \tag{3}$$

We will look at several methods for this. This is currently an area of active research, so we will keep things simple, and only look at first-order (i.e., Euler) methods in time.

IMEX

Perhaps the most straight-forward approach is just to apply a regular time-stepping (such as Euler or Runge-Kutta) for (3), but to use an implicit step for the nonlinear term, and an explicit step for the linear term. This method is often call IMEX (IM = implicit, EX = explicit). For example, using Euler, we would have

$$\frac{\widehat{u}_k^{n+1} - \widehat{u}_k^n}{\Delta t} = -\nu k^2 \widehat{u}_k^{n+1} - ikc\widehat{u}_k^n.$$

Exercise. Show that the CFL restriction for this method is $\Delta t < 2\nu/c^2$.

Integrating Factor Method

Another approach is to simplify the PDE by using an integrating factor to eliminate the linear term. This can be expensive if the linear term is given by a matrix, but in Fourier space, the linear term is just a scalar (in fact, it is multiplication by a diagonal matrix; with $-\nu k^2$ on the k^{th} diagonal entry). Multiplying (3) by

$e^{\nu k^2 t}$, we find

$$\frac{d}{dt}\widehat{u}_k e^{\nu k^2 t} + \nu k^2 \widehat{u}_k e^{\nu k^2 t} = -ikc\widehat{u}_k e^{\nu k^2 t} \tag{4}$$

$$\Rightarrow \quad \frac{d}{dt}\left(\widehat{u}_k e^{\nu k^2 t}\right) = -ikc\widehat{u}_k e^{\nu k^2 t} \tag{5}$$

We can then apply a time-stepping method to this, using the change-of-variables $\widehat{v}_k = \widehat{u}_k e^{-\nu k^2 t}$. Runge-Kutta methods have several simplifications that one can work out. However for simplicity, let us just use Euler's method here:

$$\frac{\widehat{u}_k^{n+1} e^{\nu k^2 t_{n+1}} - \widehat{u}_k^n e^{\nu k^2 t_n}}{\Delta t} = -ikce^{\nu k^2 t_n}\widehat{u}_k^n$$

Let us simplify this a little. Multiplying by Δt and then by $e^{-\nu k^2 t_{n+1}}$ (and noting that $t_{n+1} = t_n + \Delta t$), we find

$$\begin{aligned}\widehat{u}_k^{n+1} &= \widehat{u}_k^n e^{-\nu k^2 \Delta t} - ikc\Delta t e^{-\nu k^2 \Delta t}\widehat{u}_k^n \\ &= (1 - ikc\Delta t)\, e^{-\nu k^2 \Delta t}\widehat{u}_k^n\end{aligned}$$

We therefore require

$$|\,(1 - ikc\Delta t)\, e^{-\nu k^2 \Delta t}| < 1$$

That is,

$$\begin{aligned}&|1 - ikc\Delta t| < e^{\nu k^2 \Delta t} \\ \Rightarrow \quad &1 + k^2 c^2 \Delta t^2 < e^{\nu k^2 \Delta t}\end{aligned}$$

Let $x = \nu k^2 \Delta t$, and $a = c^2 \Delta t/\nu$, so that $ax = k^2 c^2 \Delta t^2$. The above inequality becomes

$$1 + ax < e^x, \text{ that is, } a < \frac{e^x - 1}{x}$$

The right-hand side of the last equation is bounded below by 1 for any $x > 0$, so we only need to require $a < 1$ to have the desired inequality. This means $c^2 \Delta t/\nu < 1$ from the definition of a, so the requirement on the time-step becomes

$$\Delta t < \frac{\nu}{c^2}.$$

ETD

A slight modification to the Integrating Factor Method is to solve for the nonlinear term exactly at the PDE level. This is called Exponential Time Differencing (ETD). It is also referred to as the exponential integrators method. Integrating (4) on the time interval $[t_n, t_{n+1}]$, we find the *exact* relation (no numerical approximation yet):

$$\widehat{u}_k(t_{n+1})e^{\nu k^2 t_{n+1}} - \widehat{u}_k(t_n)e^{\nu k^2 t_n} = -ikc\int_{t_n}^{t_{n+1}} \widehat{u}_k(t)e^{\nu k^2 t}\, dt$$

Multiplying by $e^{-\nu k^2 t_{n+1}}$ (and using the fact that $t_{n+1} = t_n + \Delta t$) yields

$$\widehat{u}_k(t_{n+1}) = \widehat{u}_k(t_n)e^{-\nu k^2 \Delta t} - ikc\int_{t_n}^{t_{n+1}} \widehat{u}_k(t)e^{-\nu k^2 (t_{n+1} - t)}\, dt \tag{6}$$

Since we do not know $\widehat{u}_k(t)$, we cannot perform the integration directly. Therefore, in the ETD method, we approximate it somehow. Let us try a very simple approximation; namely, on $[t_n, t_{n+1}]$, approximate $\widehat{u}_k(t)$ by the constant value $\widehat{u}_k(t_n)$. Then, we find

$$\widehat{u}_k(t_{n+1}) \approx \widehat{u}_k(t_n)e^{-\nu k^2 \Delta t} - ikc\widehat{u}_k(t_n)\int_{t_n}^{t_{n+1}} e^{-\nu k^2 (t_{n+1} - t)}\, dt$$

Performing the integration yields

$$\widehat{u}_k(t_{n+1}) \approx \widehat{u}_k(t_n)e^{-\nu k^2 \Delta t} - ikc\widehat{u}_k(t_n)\frac{e^{-\nu k^2(t_{n+1}-t_{n+1})} - e^{-\nu k^2(t_{n+1}-t_n)}}{\nu k^2}$$

$$\Rightarrow \quad \widehat{u}_k(t_{n+1}) \approx \widehat{u}_k(t_n)e^{-\nu k^2 \Delta t} - ikc\widehat{u}_k(t_n)\frac{1 - e^{-\nu k^2 \Delta t}}{\nu k^2}$$

This yields the following method

$$\widehat{u}_k^{n+1} = e^{-\nu k^2 \Delta t}\widehat{u}_k^n - \frac{1 - e^{-\nu k^2 \Delta t}}{\nu k^2 \Delta t}\left(ikc\Delta t\widehat{u}_k^n\right) \tag{7}$$

Let us remark that the function

$$f(x) = \begin{cases} \frac{1-e^{-x}}{x} & \text{if } x \neq 0, \\ 1 & \text{if } x = 0. \end{cases}$$

although continuous, can be very badly behaved near $x = 0$ from a computational perspective, due to cancellation error of terms which are nearby. Therefore, care must be taken in careful, accurate evaluation of the corresponding factor in (7). For example, see Higham, "Accuracy and Stability of Numerical Algorithms", p. 20, where the following algorithm is discussed. The seemingly pointless use of a logarithm actually helps to reduce the computational error.

```
% Compute f(x) = (exp(x) - 1)/x more accurately.
y = exp(x)
if y == 1
  f = 1;
else
  f = (y - 1)/log(y);
end
```