Simulation of time-dependent PDE's with finite elements and high-order A-stable IRK timesteppers

Josh Engwer

November 30^{th} 2016

Explicit Runge-Kutta (ERK) timesteppers such as Forward Euler & Classic RK4 are desired due to their relative ease of implementation and low computational cost. However, they tend to be unstable for most real-world time-dependent ODE's & PDE's unless the timestep Δt is reduced to an impractically small value as dictated by possible constraints (such as the Courant-Friedrichs-Lewy condition for hyperbolic PDE's for example.) At best, this results in unbearably long running times in order to achieve a stable simulation with a meaningful time interval, $[0, T]$. At worst, the required timestep size Δt is smaller than machine precision ϵ_{mach}, effectively rounding down the timestep to zero!

To help mitigate these potential issues, one is encouraged by Dalquist's 2nd Barrier to resort to A-stable implicit Runge-Kutta (IRK) timesteppers, the simplest of which is Backward Euler. Unfortunately, Backward Euler is only a 1st-order timestepper, meaning halving the timestep only halves the solution error. One can do far better than that as there are several classes of higher-order A-stable IRK timesteppers to choose from. In particular, the s-stage Gauss-Legendre (GLs) family of IRK methods will be considered as they are A-stable as well as symplectic.

<table>
<tr><th>TIMESTEPPER:</th><th>BUTCHER TABLEAU:</th><th>ORDER:</th></tr>
<tr><td>GL1</td><td>$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$</td><td>$2^{nd}$-order</td></tr>
<tr><td>GL2</td><td>$\begin{array}{c|cc} \frac{1}{2}-\frac{1}{6}\sqrt{3} & \frac{1}{4} & \frac{1}{4}-\frac{1}{6}\sqrt{3} \\ \frac{1}{2}+\frac{1}{6}\sqrt{3} & \frac{1}{4}+\frac{1}{6}\sqrt{3} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$</td><td>$4^{th}$-order</td></tr>
</table>

Higher-order IRK timesteppers are typically introduced to solve certain stiff ODE's, and occasionally they are utilized together with finite difference methods (FDM's) in solving certain time-dependent PDE's. This talk will utilize these timesteppers with Lagrange finite elements in order to solve the non-linear Heat Equation equipped with inhomogeneous Dirichlet boundary conditions:

$$\begin{cases} \text{PDE:} & u_t - \nabla \cdot (q(u)\nabla u) = f(t; \mathbf{x}) & \text{in } \Omega := [0,1] \times [0,1] \\ \text{BC's:} & u = u_D & \text{on } \partial\Omega \\ \text{IC:} & u = u_0 & \text{at } t = 0 \end{cases}$$

$$\text{where} \quad q(u) := u^2 \quad \text{and} \quad \mathbf{x} = (x, y)^T \quad \text{and} \quad u \in C^{(2,1)}(\Omega \times [0, \infty))$$

The simulation codes use the open-source FEniCS finite element framework to provide the necessary finite element method, and additional Python code is used to produce the timestepping, solution visualization, L^2 solution error, validation tests, and convergence rate plots of Heat Equation problems constructed using the 'Method of Manufactured Solutions'. Some remarks regarding implementation in FEniCS will be provided.