Extra Credit Problem Set

Chris Rackauckas

May 19, 2019

This problem set is for making up points lost on the midterm. You cannot exceed 100% on the midterm by completing this problem set.

1 Pseudospectral PDE Solver (8 points)

Write a pseudospectral method for solving the 1D semilinear Heat equation

$$u_t = \Delta u + u^2 - tu - u^3$$

with $u_0(x) = x(x - 2\pi)$ on $[0, 2\pi]$ with periodic boundary conditions on $t \in (0, 5)$. Use an FFT and inverse FFT with a series of 100 points to transform between point space and coefficient space.

Hint: a simple way to setup an FFT for these transformations is to use ApproxFun.jl:

```
using ApproxFun

#initialize Values
n  = 100
S  = Fourier()
F  = ApproxFun.plan_transform!(S, n)
iF = ApproxFun.plan_itransform(S, n)
x  = points(S, n)
Dxx = Derivative(S,2)[1:n,1:n]
```

If you use this version, explain why the D_{xx} operator has the values it does by looking at the definition at http://juliaapproximation.github.io/ApproxFun.jl/latest/usage/spaces.html.

2 Implicit Time Stepping

Write a function that uses the Trapezoid method with Newton iterations and fixed time steps to solve the system of N ODEs resulting from a method of lines discretization of

$$u_t = \Delta u + u^2 - tu - u^3$$

with $u_0(x) = -x^2 + \pi^2$ on $[-\pi, \pi]$ with boundary conditions $u(-\pi, t) = u(\pi, t) = 0$ with $N = 64$ and $t \in (0, 5)$. Let $\Delta t = 0.1$. Your solver function should not use any matrix inversions and should use quasi-Newton iteration, factorizing only a single matrix each time step.

Hints: For your Newton iterations, note that $fact = lu(L)$ in Julia allows you to do $fact \backslash b_1$ and $fact \backslash b_2$ without re-factorizing. Doing $L \backslash b_1$ and $L \backslash b_2$ would re-factorize. In your Newton iteration you will need to choose a tolerance to stop iterating at. One example tolerance is 10^{-8}.