Fractional Finite Difference for Simulating the MMT Equation

Stephen Guth

MIT

The Majda-MacLaughlin-Tabak equation models 1D dispersive waves with intermittent focusing events.

$$i\frac{\partial u}{\partial t} = \left(\frac{\partial}{\partial x}\right)^{\frac{1}{2}} u + \lambda |u|^2 u + iDu \qquad (1)$$

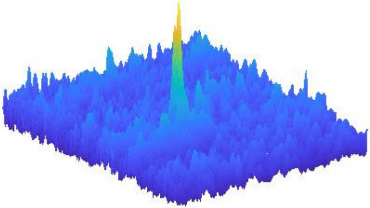

Figure: Sample intermittent event.

- want to simulate MMT to look at non-Gaussian statistics
- want to simulate *efficiently* because extreme events are rare

We have a perfectly good **spectral** solver, via **Riesz** derivative

- $\widehat{\left|\frac{\partial}{\partial x}\right|^{\frac{1}{2}} u} = -|k|^{\frac{1}{2}}\hat{u}$ is diagonal
- iDu is diagonal in Fourier space
- $|u|^2 u$ is local is real space, requiring only $n \log n$ operations (Fast Fourier transform)

But what if we insisted on using finite difference methods anyway?

First step: define a fractional derivative.

Riemann-Liouville fractional derivative

$$\frac{\partial^\alpha u}{\partial x^\alpha}(x) = \frac{1}{\Gamma(n-\alpha)} \frac{\partial^n}{\partial x^n} \int_a^x u(\xi)(x-\xi)^{n-\alpha-1} d\xi \qquad (2)$$

- looks like an integral, which is bad news in terms of locality
- dependence on the fiducial point a, for the limits of integration
- not even the only sensible definition–see **Caputo fractional derivative**

Second step: make it finite difference.

Grünwald-Letnikov formula

$$\frac{\partial^\alpha u}{\partial x^\alpha}(x) = \lim_{\Delta x \to 0} \frac{1}{\Delta x^\alpha} \Sigma_{k=0}^{\frac{x-a}{\Delta x}} (-1)^k \binom{\alpha}{k} u(x - k\Delta x) \qquad (3)$$

This looks like a finite difference stencil!

- need to evaluate real binomial coefficients (which helpfully decay quickly)
- still (formally) need a fiducial point a
- no symmetry

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \qquad (4)$$

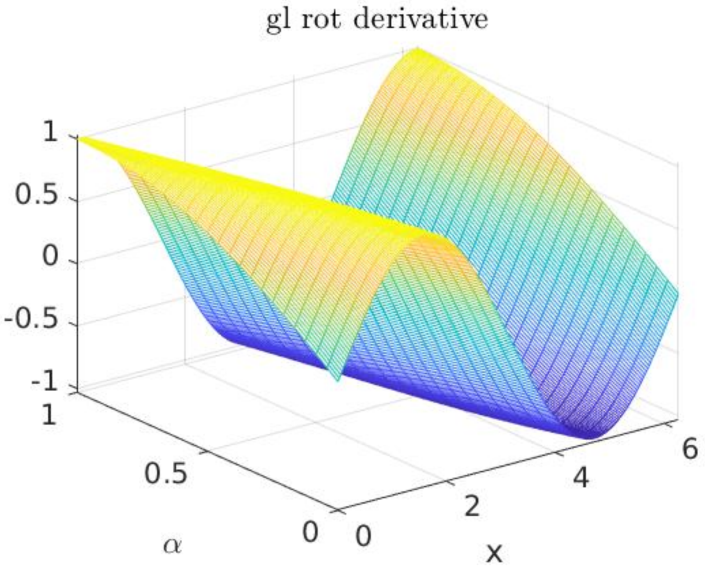

Figure: Plot of the Grünwald-Letnikov derivative of order α for the function $f(x) = \sin x$. At $\alpha = 1$, it agrees with $f'(x) = \cos(x)$.

The Grünwald-Letnikov formula by itself has **stability issues**

- shifted formula:

$$D^{\alpha}_{x,p}u = \frac{\partial^{\alpha} u}{\partial x^{\alpha}}(x + p\Delta x) \quad (5)$$

We'll use a particular linear combination called rot-3

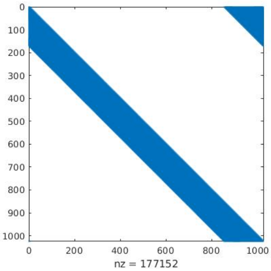

Figure: Sparsity pattern for gl-rot-3, $n = 1024$, $\alpha = 0.5$.

$$i\frac{\partial u}{\partial t} = \left(\frac{\partial}{\partial x}\right)^{\frac{1}{2}} u + \lambda |u|^2 u + iDu \tag{6}$$

- $\left(\frac{\partial}{\partial x}\right)^{\frac{1}{2}} u$ term: *handled*
- nonlinear term $|u|^2 u$: "straightforward"
- the selective Laplacian iDu is unimportant

How to handle the time integration?

The spectral code uses a **Cox Exponential integrator** called ETD4RK

- in Fourier space, $\left(\frac{\partial}{\partial x}\right)^{\frac{1}{2}} u$ and iDu are both **linear** and **diagonal**
- linear terms are solved via matrix exponential, nonlinear terms via explicit Runge Kutta

I tried

- Forward Euler
- Linearized Backward Euler
- ETD4RK

The matrix exponential in spectral ETD4RK does an impeccable job of conserving energy.

Unfortunately...

The finite difference fractional derivative does **not**.

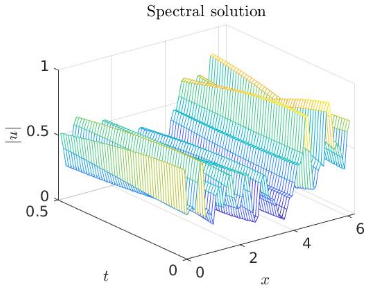

Figure: Spectral solution.

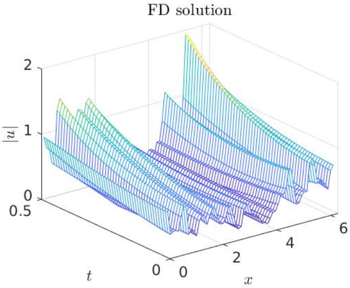

Figure: Finite difference solution, linearized backward Euler.

I tried a few methods to 'correct' for the growth and collect usable statistics anyway, but none worked out well.

Problem seems to be that FD methods don't handle the 'phase rotation' very conservatively.

Constructing the FD matrix exponentials for ETD4RK is literally melting my computer, because it requires accurate approximations to $Z^{-1}(\exp(Z) - 1)$.

Adaptive methods (Runge Kutta, etc) refine the time step into oblivion.