Creating a One Dimensional Soliton Gas in Viscous Fluid Conduits

University of Colorado, Boulder
Advisors: Mark Hoefer, Michelle Maiden

March 4, 2017

Will Farmer
www.will-farmer.com

Funded by NSF EXTREEMS-QED

CU Boulder Applied Math

High Level Overview

1. What are Conduits
2. What are solitons?
3. What's a soliton gas?
4. Simulations...

The Conduit

TODO

Notes on Solitons

Our system is governed by the Conduit Equation,

$$A_t + \left(A^2\right)_z - \left(A^2\left(A^{-1}A_t\right)_z\right)_z = 0$$

- Solitons are *solitary travelling waves*.
- Solitons are a special solution with decaying boundary conditions to the conduit equation of the form

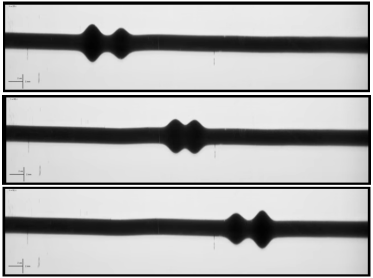
$$A(z,t) = f(\zeta) = f(z - ct)$$

- Solitons have nonlinear characteristics, most notably their speed is determined by their non-dimensionalized amplitude (a).

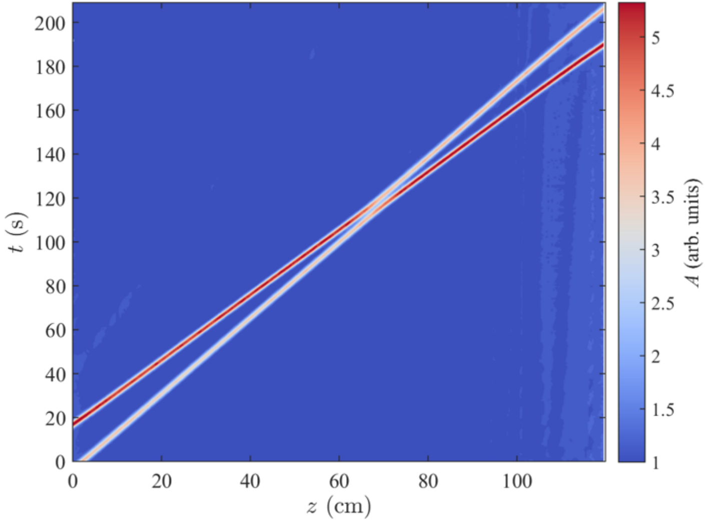
$$c = \frac{a^2 - 2a^2 \ln a - 1}{2a - a^2 - 1}$$

Soliton - Soliton Interactions

- Two solitons can interact if a bigger one chases a smaller one.
- The solitons' speed and amplitude are preserved save for a phase-shift



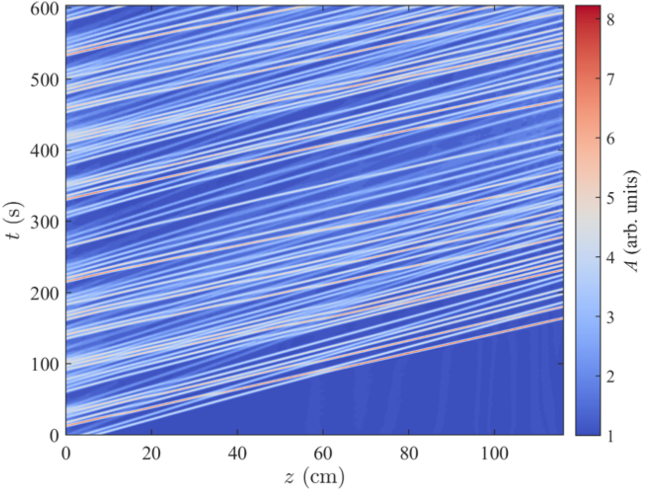
Particle-like Interactions



What's a Soliton Gas?

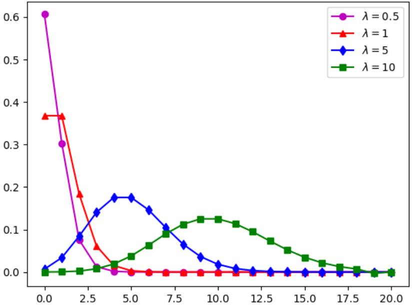
- A soliton can be thought of as a wave, but also as a particle (similar to a photon)
- A gas can be thought of as a random collection of particles interacting
- Thus a soliton gas is a random collection of solitons interacting
- Our system is one-dimensional, so we are generating a 1D gas
- A soliton gas has inherent random behavior dictated by two random variables:
  1. Frequency of solitons, Z
  2. Soliton amplitude, A

Plotting our Gas



Properties of a Soliton Gas

- Soliton gas theory developed for simpler (integrable) systems.
- Soliton centers and amplitudes $\Rightarrow$ compound Poisson process
- This means over long time, frequency of solitons, $Z \sim Poisson(\lambda)$, and A is preserved
- Poisson Distribution: Number of events in interval with known average rate and mutually independent events.



$$f(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Numerical Simulations

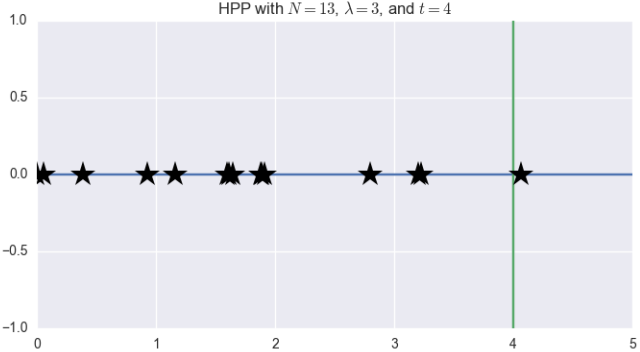
- Before running time-consuming experiments, useful to run numerical simulations
- Spatial discretization: 4th-order finite differences with periodic BC's
- Temporal discretization: medium-order adaptive Runge-Kutta (Matlab's ode45.m)

Finite Size Effects

- Since we have recurrence, eventually the simulation on $[0, L]$ will tend back to initial conditions. We want to stop before then.
- Therefore we'll run two simulations simultaneously, one on $[0, L]$ and the other on $[0, 2L]$.[1]
- At each timestep we'll check for a compound Poisson gas process ("gas metric") of each. If they differ by more than 20%, we restart with new initial conditions.
- Similarly, due to phase shifts, our gas metric should increase and then decrease. We want to stop before this.

What is our Gas Metric?

- We've established that a soliton gas should have Poisson-distributed solitons.
- This means that we can look at the problem as a Poisson-Point Process, i.e. at any given point in space we should see points appear over time.



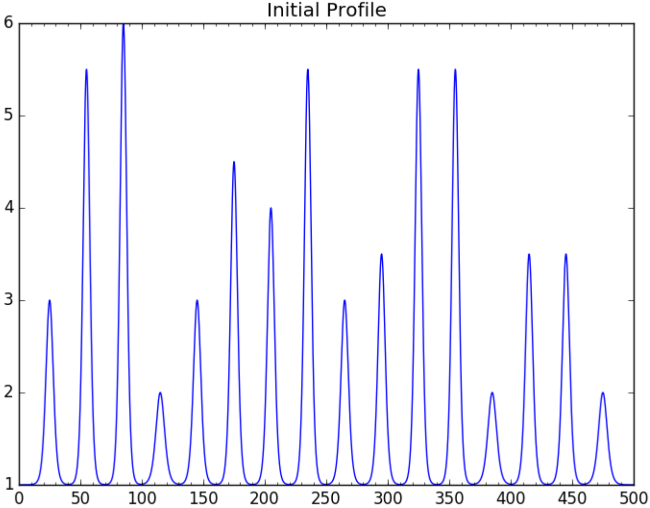
Meaning....

- Since this is a Poisson-Point Process the gap between points is exponentially distributed.
- Therefore the gas metric is a measure of how close our gaps of our solitons are to the exponential distribution.
- We use the residual sum squared on the QQ-plot as a metric of "distance" from one distribution to the other. This value is our gas metric.
- We find this gas metric at every timestep.

Initial Conditions

- We have two random variables to simulate.
- Very first case is easy
  - Z is one per minimum distance with exponentially small overlap.
  - A is $Unif(\{2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6\})$
- After a restart on $2L$ and $4L$ things are trickier... We can't feed initial conditions where solitons overlap (makes solver grumpy due to nonlinear phenomena) so instead we need to simulate initial conditions with same gas metric as last timestep before we quit.

Plot of Initial Conditions



Leveraging Parallelism

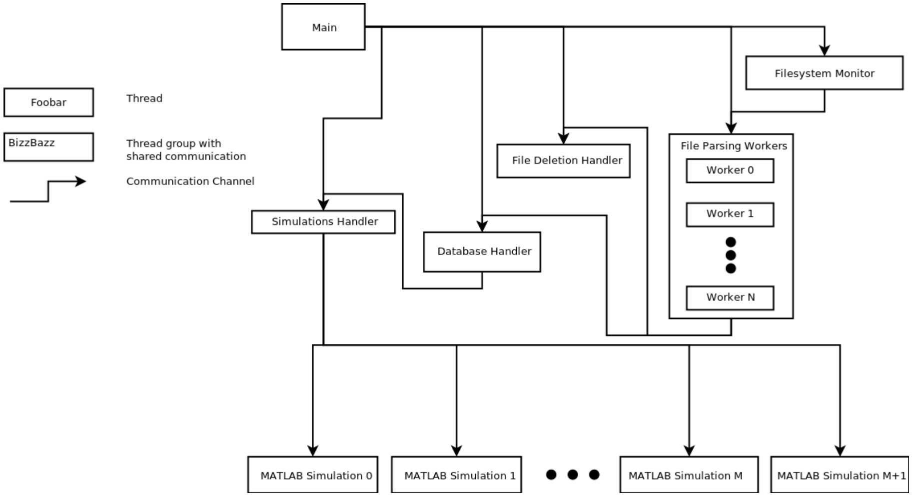
The big flaw so far is that we're only looking at a single run of the simulation. We could easily get bad results from only a single run.

Let's instead consider running a hundred different simulations simultaneously, or even a thousand. We have to adjust our simulation to be able to handle running in a massively parallel environment such as the CU supercomputer, Summit.

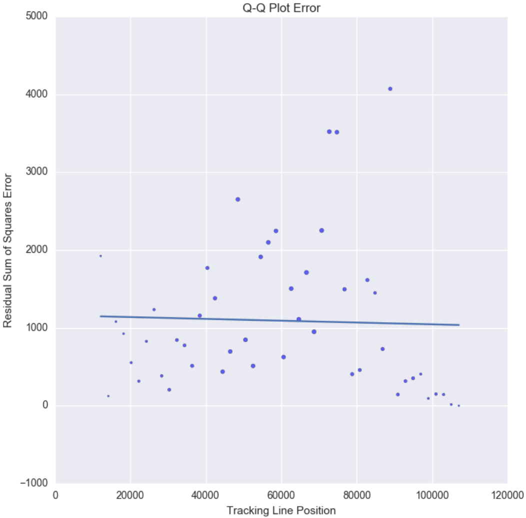
Leveraging Parallelism

- If we want to run many simulations at once, this problem can be described as *embarrassingly parallel* since the simulations don't need to talk to each other.
- So how can we design a multi-threaded program to take into account the availability of tens or hundreds of threads?
- Can this be written *safely* so we don't have any undefined behavior?

Multi-Threaded Design



Preliminary Results



Next Steps

- Large scale supercomputer simulations
- Experiments for validation of simulations
- Research paper

References and Acknowledgements

References

- M. D. Maiden, N. K. Lowman, D. V. Anderson, M. E. Schubert, and M. A. Hoefer, Observation of dispersive shock waves, solitons, and their interactions in viscous fluid conduits, Physical Review Letters 116, 174501 (2016).
- N. K. Lowman, M. A. Hoefer, and G. A. El, Interactions of large amplitude solitary waves in viscous fluid conduits, Journal of Fluid Mechanics 750, 372-384 (2014).
- D. S. Agafontsev and V. E. Zakharov, Nonlinearity 28, 2791 (2015).

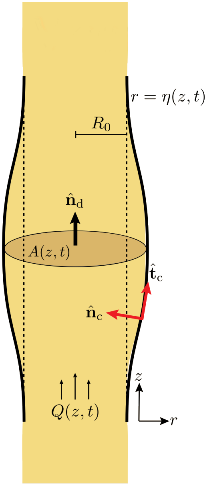
Acknowledgements

- Mark Hoefer
- Michelle Maiden
- Funded by NSF EXTREEMS-QED

Environment Details

Viscous Fluid Conduits

- Two viscous fluids, with inner forming axisymmetric conduit.
- Exterior Fluid: $\rho^{(e)}$ density and $\mu^{(e)}$ viscosity
- Interior Fluid: $\rho^{(i)}$ density and $\mu^{(i)}$ viscosity
- $\rho^{(i)} < \rho^{(e)} \Rightarrow$ buoyant flow
- $\mu^{(i)} << \mu^{(e)} \Rightarrow$ minimal drag
- $\mathrm{Re} << 1 \Rightarrow$ low Reynold's number (implies Laminar flow)



Integrable System: KDV

$$u_t + uu_x + u_{xxx} = 0$$