Inverse methods for solving shallow water equations

Final Project Presentation

By Brian Kyanjo
Supervised by: Prof. Jodi Mead

Problem justification

We are finding parameter estimates for the shallow water equation (SWE) problem using inverse methods techniques. In this work, we chose various sets of initial estimates, m = [(hl,ul,hl*ul),(hm,um,hm*um)),(hr,ur,hr*ur)] near our true solution (mtrue) to the SWE , and use the inverse methods techniques to recover mtrue.

Phenomena description

The shallow water equations (SWE) are a system of hyperbolic partial differential equations (PDEs) describing the flow below a pressure surface in a fluid.

They have been frequently used to model several real-life problems i.e. the propagation of tsunamis waves in the ocean and modeling of atmospheric turbulence.

Deep knowledge is required to handle such events, therefore first, robust, and computationally efficient methods like inverse methods are required to parameter estimates to the SWEs problems

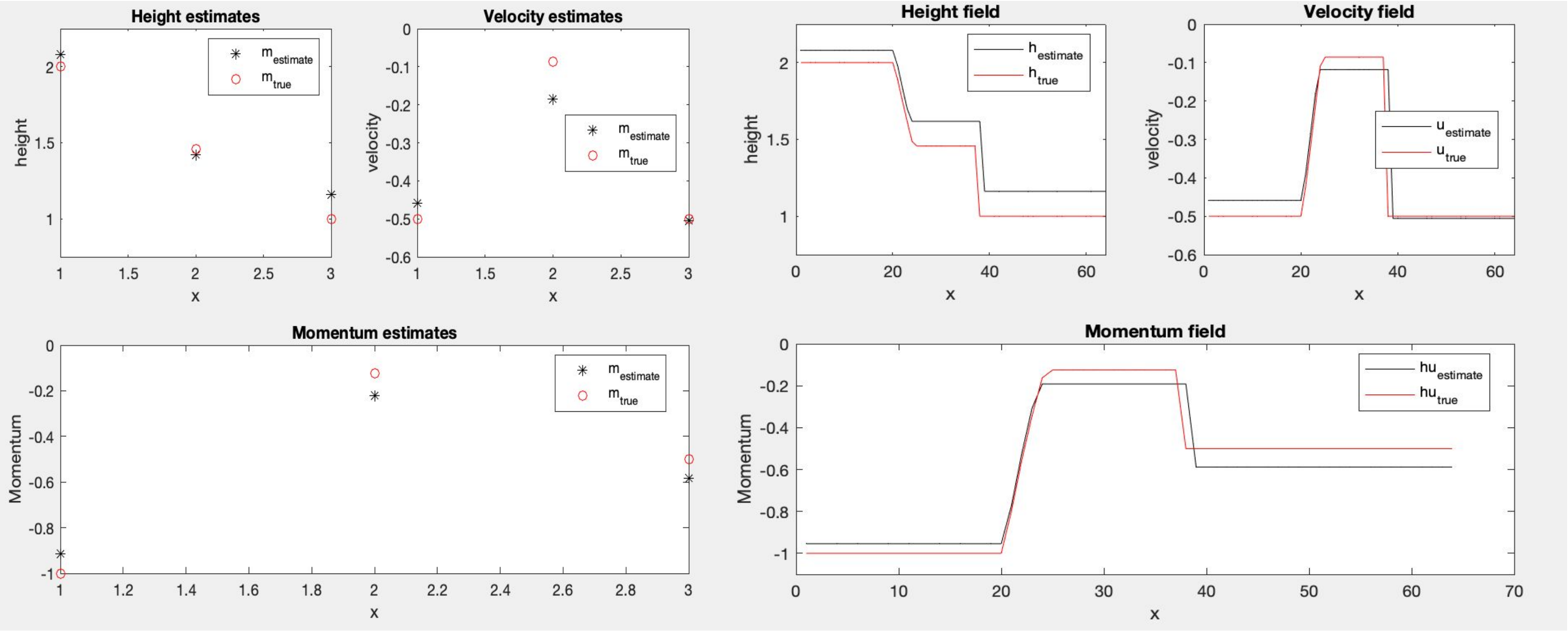
Questions

- How best can we choose the initial estimates for a SWE problem?
- How best can we recover SWEs true estimates, mtrue?
- Does the confidence interval of the estimates capture mture?
- Between the Gauss Newton method and the Occam method, which one recover closer estimates?
- How the recovered estimates relate?
- How the order of the roughening matrix impact the chi-square, p-value and L2-norm of the recovered estimates?

Approaches used

- Chosen my data to be m = [(hl,ul,hl*ul),(hm,um,hm*um),(hr,ur,hr*ur)]
- Extracted and implemented the forestclaw code to obtain G(m,x/t)
- Obtained the observed data d = G(m,x/t) + noise
- Scaled the observed data by 5% such that 0.05d_h,u,hu<std_h,u,hu
- Formed synthetic data, dnew = G(m,x/t) + 0.05d_h,u,hu
- Approximated the Jacobian J using the centered difference.
- Used the Occam model to recover mtrue

Results


Height estimates
Velocity estimates
Height field
Velocity field
Momentum estimates
Momentum field
height
velocity
Momentum
x
$m_{estimate}$
m_{true}
$h_{estimate}$
h_{true}
$u_{estimate}$
u_{true}
$hu_{estimate}$
hu_{true}

Confidence intervals and regions

mtrue =

	h	u	hu
r	2.0000	-0.5000	-1.0000
m	1.4571	-0.0858	-0.1250
l	1.0000	-0.5000	-0.5000

cofidence_interval_height =

-2.0528	6.2456
-2.7290	5.5694
-2.9935	5.3049

covariance_matrix =

4.4814	-4.4333	-0.2542
-4.4333	4.7621	-0.0597
-0.2542	-0.0597	0.3228

cofidence_interval_velocity =

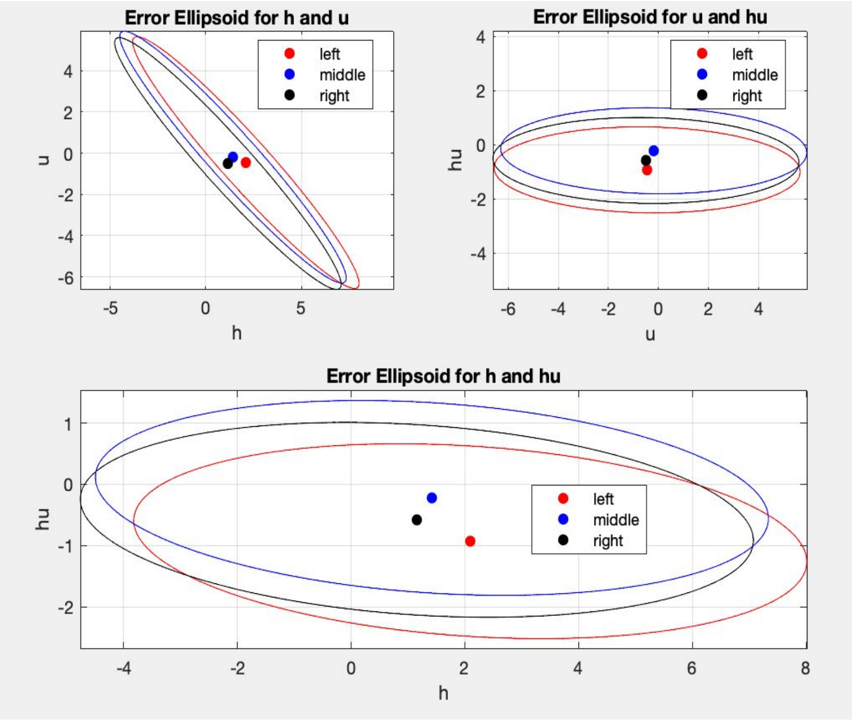
-4.7359	3.8184
-4.4700	4.0844
-4.7801	3.7742

Correlation_matrix =

1.0000	-0.9597	-0.2113
-0.9597	1.0000	-0.0481
-0.2113	-0.0481	1.0000

cofidence_interval_Momentum =

-2.0406	0.1867
-1.3347	0.8927
-1.6924	0.5349



Chi-squares, P-values, and L2-norms

	L0			L1			L2		
	h	**u**	**hu**	**h**	**u**	**hu**	**h**	**u**	**hu**
$\chi 2$	1.4564	0.20341	0.39861	1.4564	0.20356	0.39916	1.4559	0.20311	0.39858
p-value	0.69237	0.97704	0.94053	069237	0.97701	0.94042	0.6925	0.97709	0.94054
L2-norm	0.1888			0.1884			0.1880		

Findings

I discovered that recovering mtrue, depends on:

- The initial estimate ,mo, used.
- The value of the standard deviation used to generate the noise.
- The step size ,h,
- The inversion model used which high depends on the range of values of alpha used.
- The order of the roughening matrix used.

P-value is highly affected by input parameters: tfinal, h, roughening matrix, inversion model, N, e.t.c.