Still water and other hard-to-model systems

Martin Büttner
Supervisor: Prof. Ted Johnson

16 March, 2015

The Shallow Water Equations

- Simplified fluid model:
  - Incompressible
  - Inviscid
  - *Shallow* (horizontal length scale $\gg$ typical depth)
- Used to model oceans
  and atmosphere

The Shallow Water Equations

- Simplified fluid model:
  - Incompressible
  - Inviscid
  - *Shallow* (horizontal length scale $\gg$ typical depth)
- Used to model oceans ($\sim$ 4km deep)
  and atmosphere ($\sim$ 10km deep)

The Shallow Water Equations

- Simplified fluid model:
  - Incompressible
  - Inviscid
  - *Shallow* (horizontal length scale $\gg$ typical depth)
- Used to model oceans ($\sim$ 4km deep) and atmosphere ($\sim$ 10km deep)

$$h_t + (hu)_x = 0 \tag{1}$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x = 0 \tag{2}$$

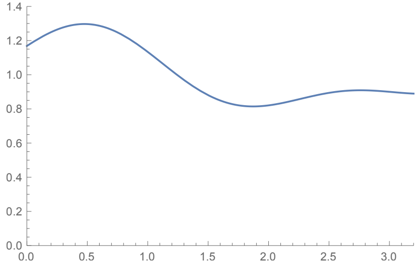
$$(hv)_t + (huv)_x = 0 \tag{3}$$

Finite Volume Methods

- Developed for *hyperbolic conservation laws*:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{4}$$

- Discretisation:

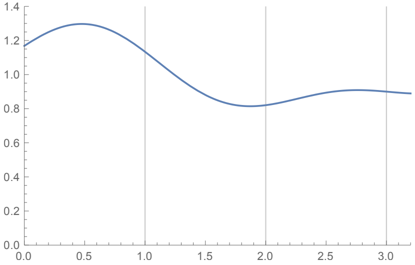


Finite Volume Methods

- Developed for *hyperbolic conservation laws*:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{4}$$

- Discretisation:

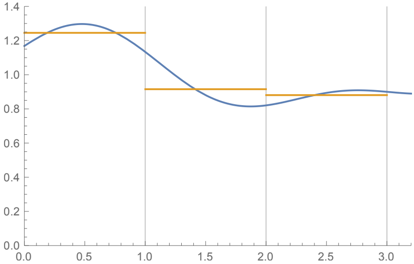


Finite Volume Methods

- Developed for *hyperbolic conservation laws*:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{4}$$

- Discretisation:

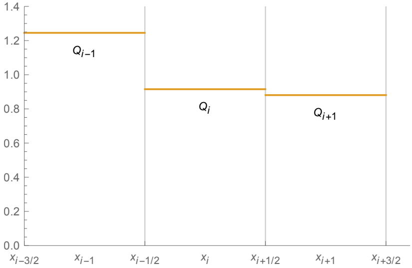


Finite Volume Methods

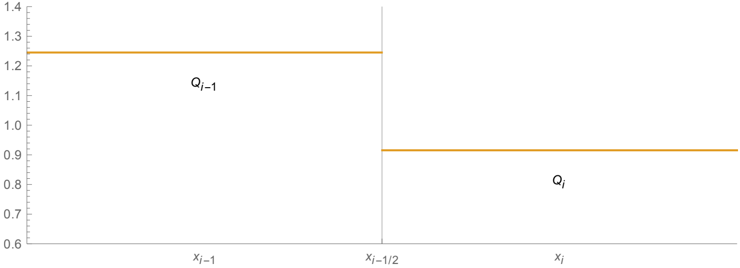
- Developed for *hyperbolic conservation laws*:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{4}$$

- Discretisation:

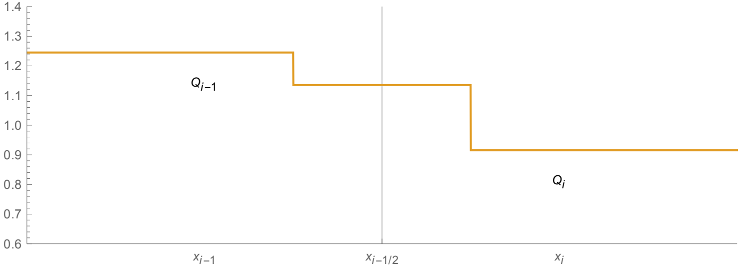


Riemann Problems



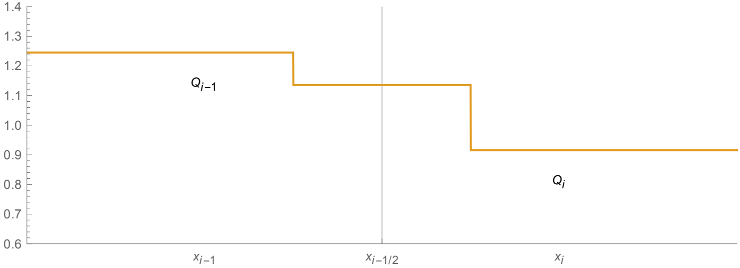
- Initial value problem: step function

Riemann Problems



- Initial value problem: step function
- Decompose into waves

Riemann Problems



- Initial value problem: step function
- Decompose into waves
- Finite speeds $\Rightarrow$ only affect neighbouring cells

Implementation: Clawpack

- "**C**onservation **law** **pack**age", by Randall J. LeVeque, 1994

Implementation: Clawpack

- "**C**onservation **law pack**age", by Randall J. LeVeque, 1994
- Implements "Godunov's method" in *wave-propagation form*

Implementation: Clawpack

- “**C**onservation **law pack**age”, by Randall J. LeVeque, 1994
- Implements “Godunov’s method” in *wave-propagation form*
- Riemann solvers in Fortran 90
  - Take grid data, compute wave decomposition

Implementation: Clawpack

- "**C**onservation **law pack**age", by Randall J. LeVeque, 1994
- Implements "Godunov's method" in *wave-propagation form*
- Riemann solvers in Fortran 90
  - Take grid data, compute wave decomposition
- Configuration in Python 2
  - Grid setup
  - Initial conditions
  - Boundary conditions
  - Plotting instructions

Adding Source Terms

- Modelling additional effects:

$$h_t + (hu)_x = 0 \tag{5}$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x = 0 \tag{6}$$

$$(hv)_t + (huv)_x = 0 \tag{7}$$

Adding Source Terms

- Modelling additional effects:

$$h_t + (hu)_x = 0 \tag{5}$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x = -hB_x \tag{6}$$

$$(hv)_t + (huv)_x = 0 \tag{7}$$

Adding Source Terms

- Modelling additional effects:

$$
\begin{aligned}
h_t + (hu)_x &= 0 && (5)\\
(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x &= -hB_x + Khv && (6)\\
(hv)_t + (huv)_x &= -Khu && (7)
\end{aligned}
$$

Adding Source Terms

- Modelling additional effects:

$$h_t + (hu)_x = 0 \tag{5}$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x = -hB_x + Khv \tag{6}$$

$$(hv)_t + (huv)_x = -Khu \tag{7}$$

- Full form of conservation law:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x) \tag{8}$$

Adding Source Terms

- Split into two steps:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{9}$$

$$\mathbf{q}_t = \mathbf{s}(\mathbf{q}, x) \tag{10}$$

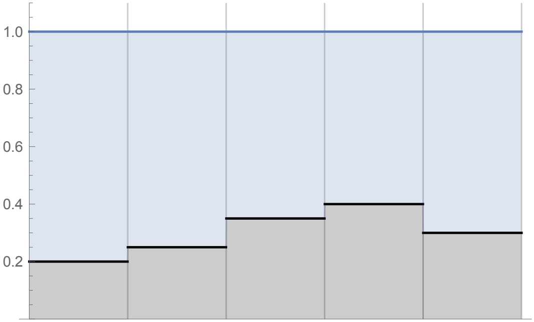
Adding Source Terms

- Split into two steps:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{9}$$

$$\mathbf{q}_t = \mathbf{s}(\mathbf{q}, x) \tag{10}$$

- The still water system:



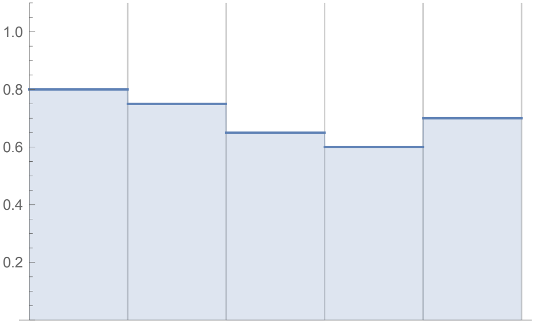
Adding Source Terms

- Split into two steps:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0 \tag{9}$$

$$\mathbf{q}_t = \mathbf{s}(\mathbf{q}, x) \tag{10}$$

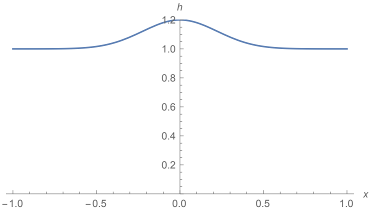
- The still water system:

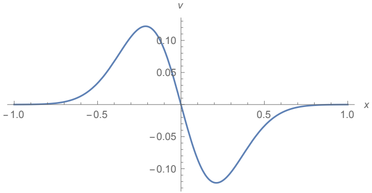


Geostrophic Equilibria

- Coriolis force admits non-trivial equilibria when

$$u = 0, \quad v = \frac{h(h_s)_x}{K} \tag{11}$$





- Geophysical flows close to geostrophic equilibrium at all times

Well-Balanced Methods

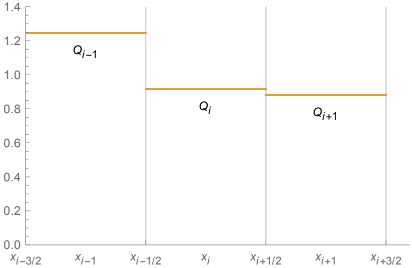
LeVeque

- LeVeque, 1998, "Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm"

Well-Balanced Methods

LeVeque

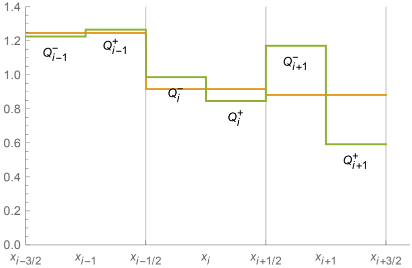
- LeVeque, 1998, "Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm"



Well-Balanced Methods

LeVeque

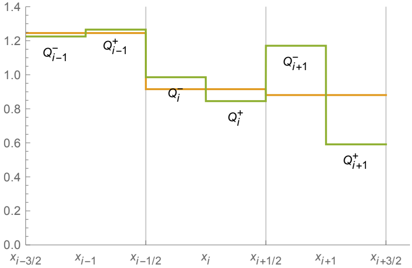
- LeVeque, 1998, "Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm"



Well-Balanced Methods

LeVeque

- LeVeque, 1998, “Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm”



$$\mathbf{f}(Q_i^+) - \mathbf{f}(Q_i^-) = \mathbf{s}(Q_i, x_i)\Delta x \tag{12}$$

Well-Balanced Methods

Rogers et al.

- Rogers, Borthwick, Taylor, 2003, "Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver"

Well-Balanced Methods

Rogers et al.

- Rogers, Borthwick, Taylor, 2003, "Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver"
- Change of variables: deviations from equilibrium

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x) \tag{13}$$

$$\to (\mathbf{q} - \mathbf{q}_{\mathrm{eq}})_t + (\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}_{\mathrm{eq}}))_x = \mathbf{s}(\mathbf{q}, x) - \mathbf{s}(\mathbf{q}_{\mathrm{eq}}, x) \tag{14}$$

$$\text{or} \quad \mathbf{q}^*_t + \mathbf{f}^*_x = \mathbf{s}^* \tag{15}$$

Well-Balanced Methods

Rogers et al.

- Rogers, Borthwick, Taylor, 2003, "Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver"
- Change of variables: deviations from equilibrium

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x) \tag{13}$$

$$\rightarrow (\mathbf{q} - \mathbf{q}_{\mathrm{eq}})_t + (\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}_{\mathrm{eq}}))_x = \mathbf{s}(\mathbf{q}, x) - \mathbf{s}(\mathbf{q}_{\mathrm{eq}}, x) \tag{14}$$

$$\text{or} \quad \mathbf{q}^*_t + \mathbf{f}^*_x = \mathbf{s}^* \tag{15}$$

- At equilibrium, all terms vanish

Well-Balanced Methods

Rogers et al.

- Rogers, Borthwick, Taylor, 2003, "Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver"
- Change of variables: deviations from equilibrium

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x) \tag{13}$$

$$\to (\mathbf{q} - \mathbf{q}_{\text{eq}})_t + (\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}_{\text{eq}}))_x = \mathbf{s}(\mathbf{q}, x) - \mathbf{s}(\mathbf{q}_{\text{eq}}, x) \tag{14}$$

$$\text{or} \quad \mathbf{q}^*_t + \mathbf{f}^*_x = \mathbf{s}^* \tag{15}$$

- At equilibrium, all terms vanish
- Jacobian of $\mathbf{f}$ remains unchanged
  - easy to adapt existing solver

Well-Balanced Methods

Rogers et al.

- Derivation in original paper leads to

$$\mathbf{q}_t^* + \mathbf{f}'(\mathbf{q})\mathbf{q}_x^* = \mathbf{s}^* \tag{16}$$

- Implementation based on this failed away from equilibrium

- Derivation in original paper leads to

$$\mathbf{q}_t^* + \mathbf{f}'(\mathbf{q})\mathbf{q}_x^* = \mathbf{s}^* \tag{16}$$

- Implementation based on this failed away from equilibrium
- Derived alternative form:

$$\mathbf{q}_t^* + \mathbf{f}'(\mathbf{q})\mathbf{q}_x^* = \mathbf{s}^* - \frac{\partial \mathbf{f}^*}{\partial \mathbf{q}_{\text{eq}}}(\mathbf{q}_{\text{eq}})_x \tag{17}$$

- This does work!

Project work

- Extend these methods the full SWEs with Coriolis force

Project work

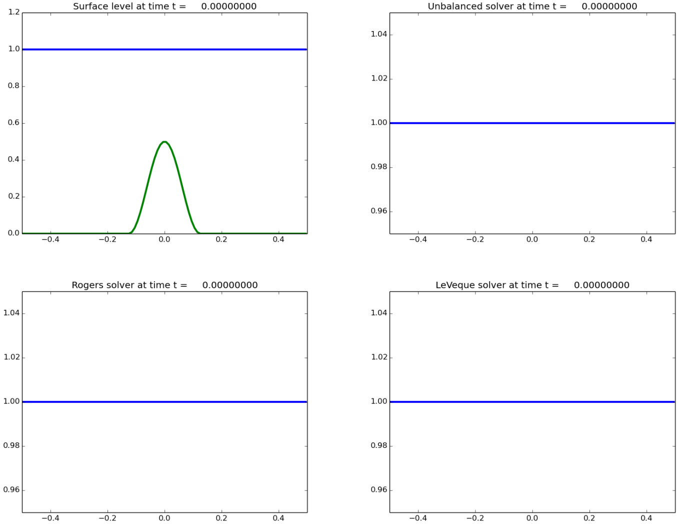
- Extend these methods the full SWEs with Coriolis force
- Also, derived a Rogers solver for geostrophic equilibria
  - Unfortunately, implementation is work in progress.

Project work

- Extend these methods the full SWEs with Coriolis force
- Also, derived a Rogers solver for geostrophic equilibria
  - Unfortunately, implementation is work in progress.
- Evaluation
  - Implemented evaluation framework in Clawpack (Python)
  - Supports 7 different bathymetries and 5 initial conditions
  - Implemented 4 Riemann solvers in Fortran (unbalanced, LeVeque, Rogers for still water, Rogers for geostrophic equilibrium)

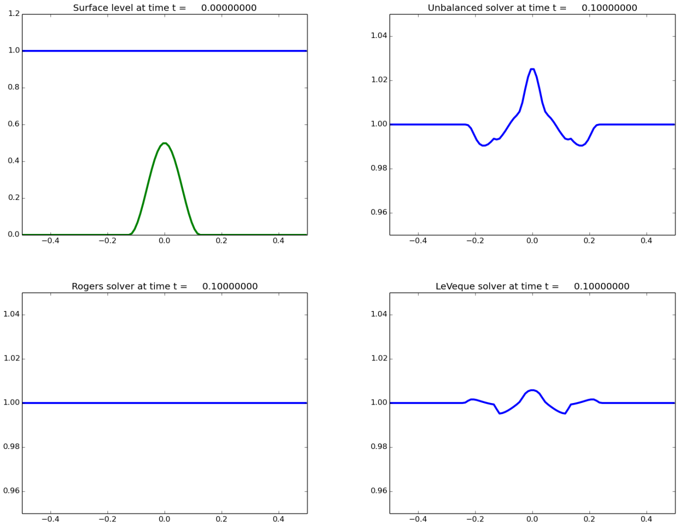
Results

Still water



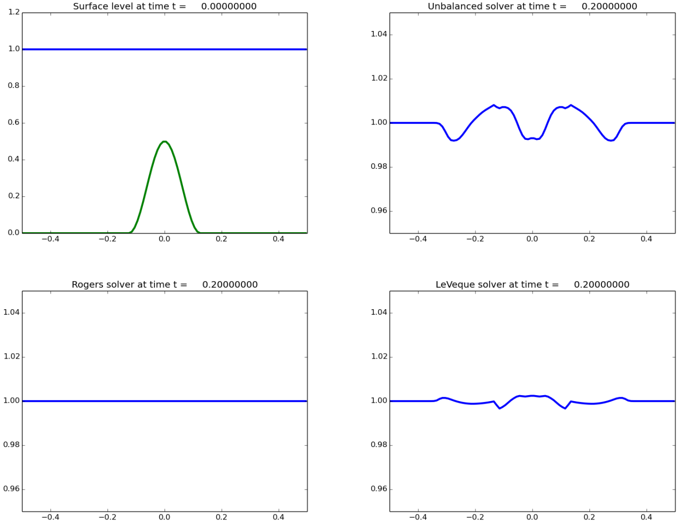
Results

Still water



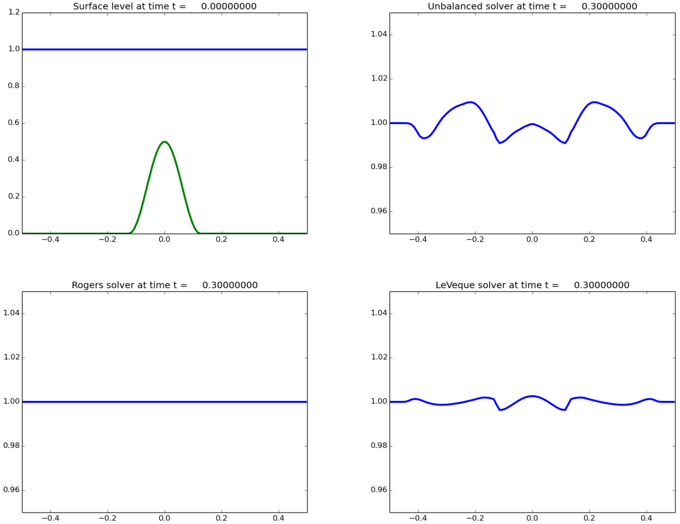
Results

Still water



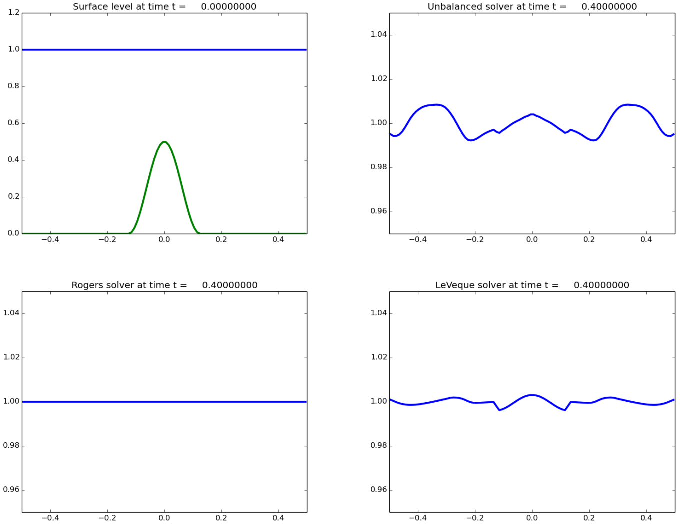
Results

Still water



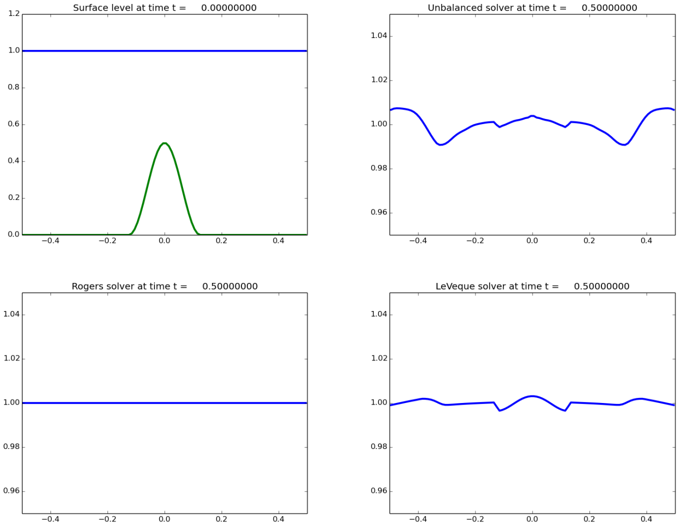
Results

Still water



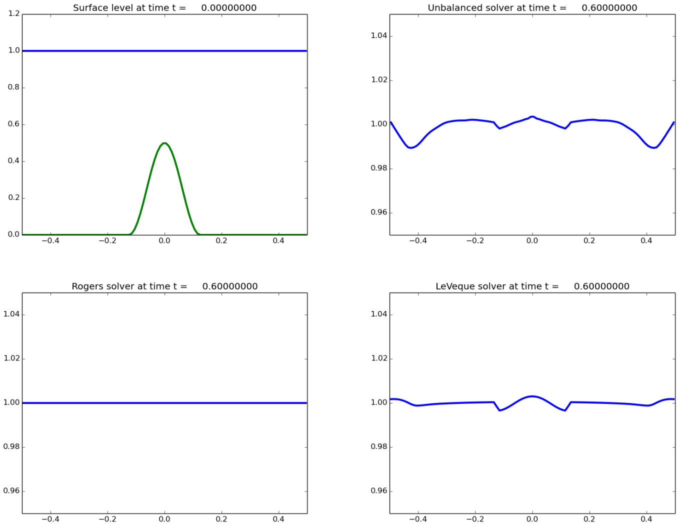
Results

Still water



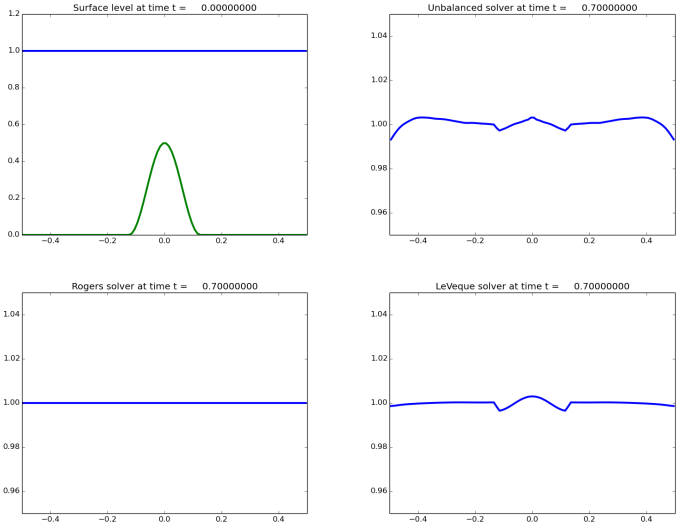
Results

Still water



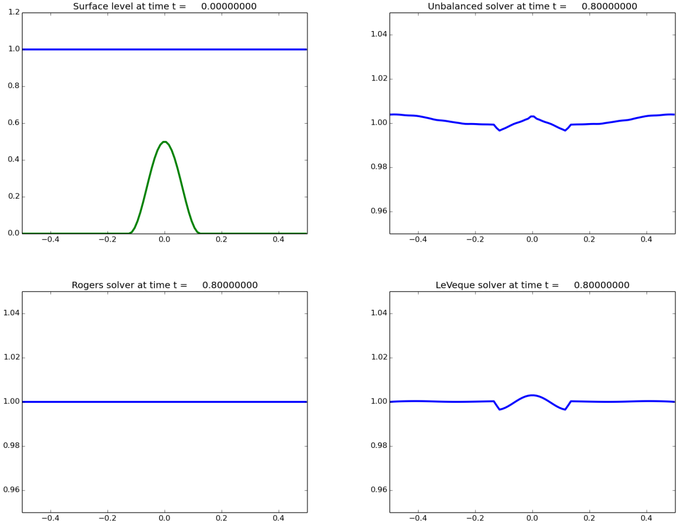
Results

Still water



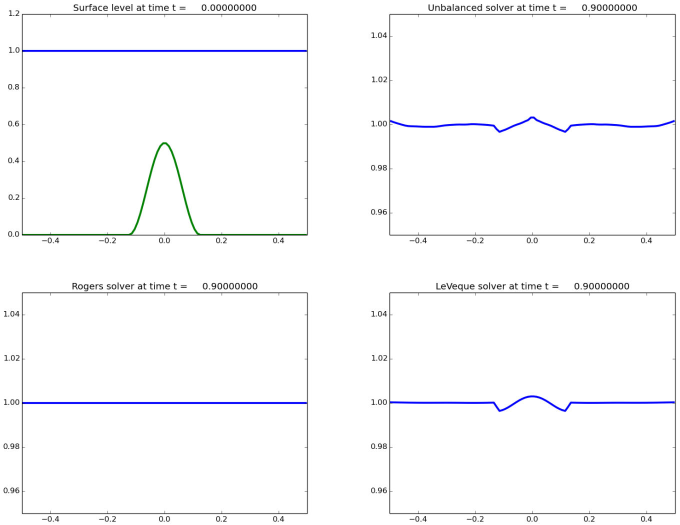
Results

Still water



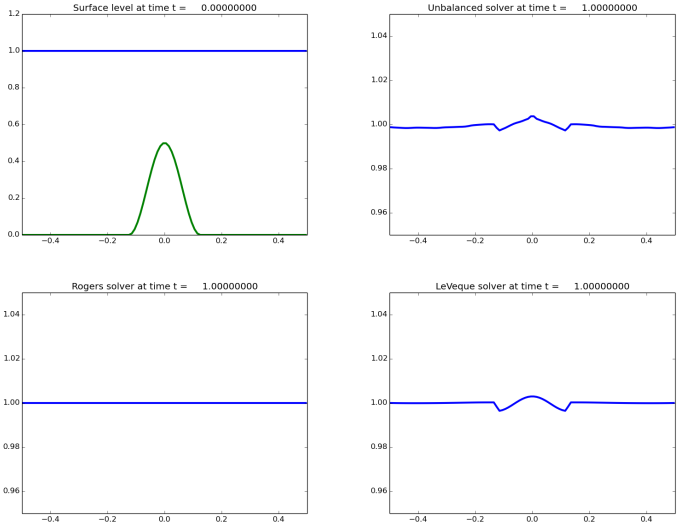
Results

Still water



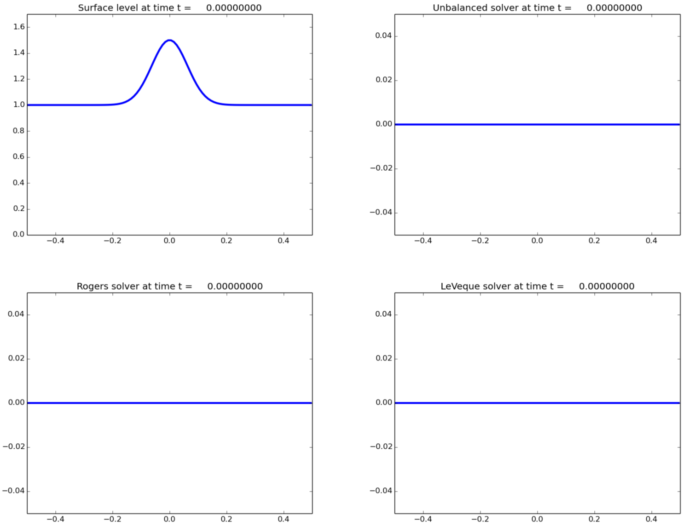
Results

Still water



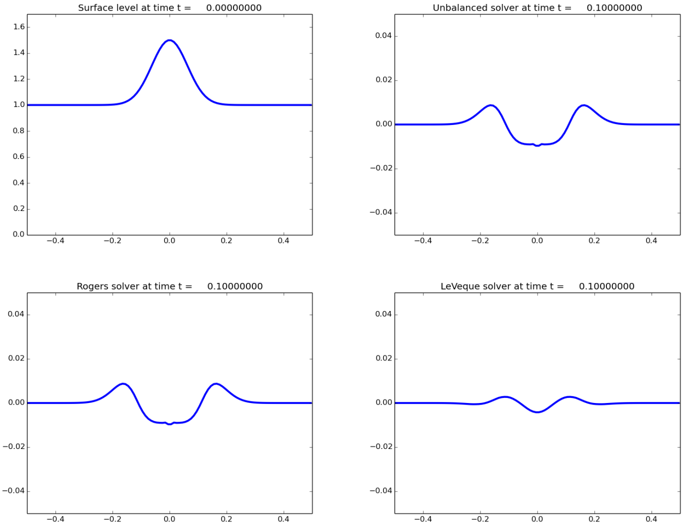
Results

Geostrophic equilibrium



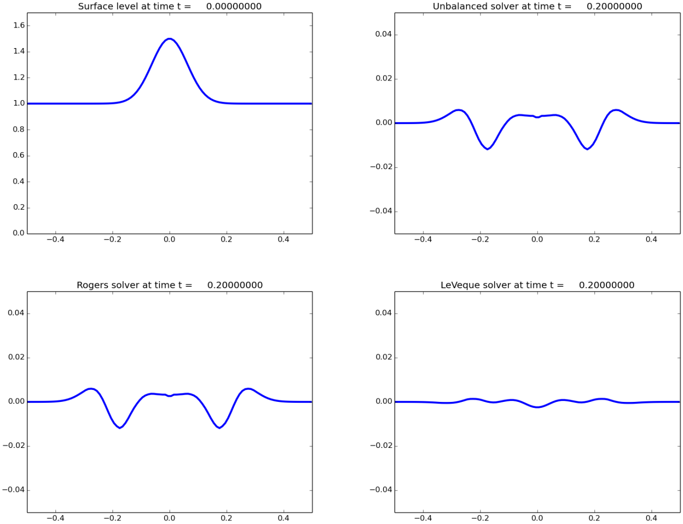
Results

Geostrophic equilibrium



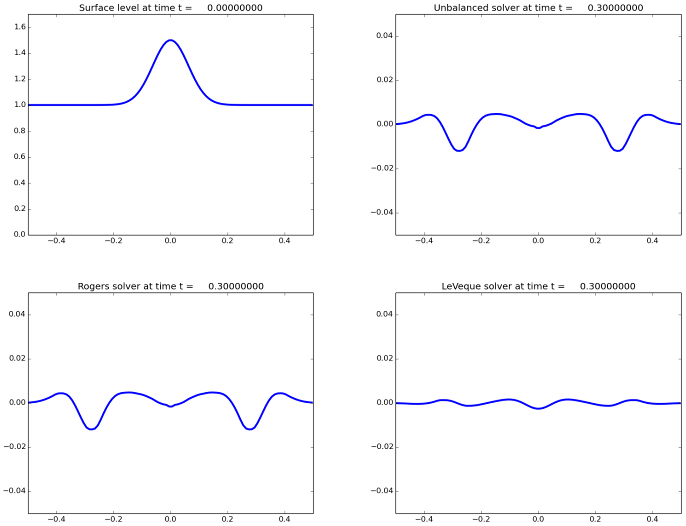
Results

Geostrophic equilibrium



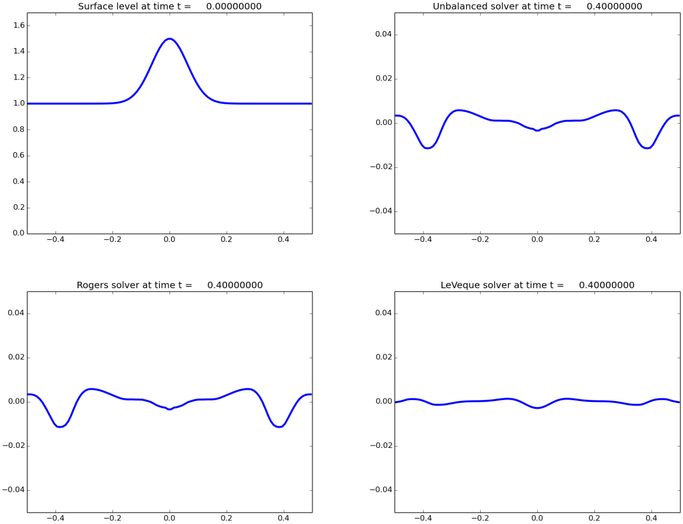
Results

Geostrophic equilibrium



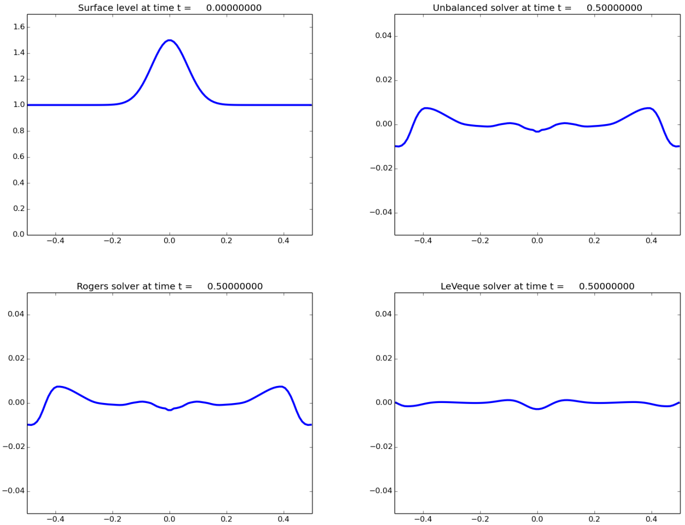
Results

Geostrophic equilibrium



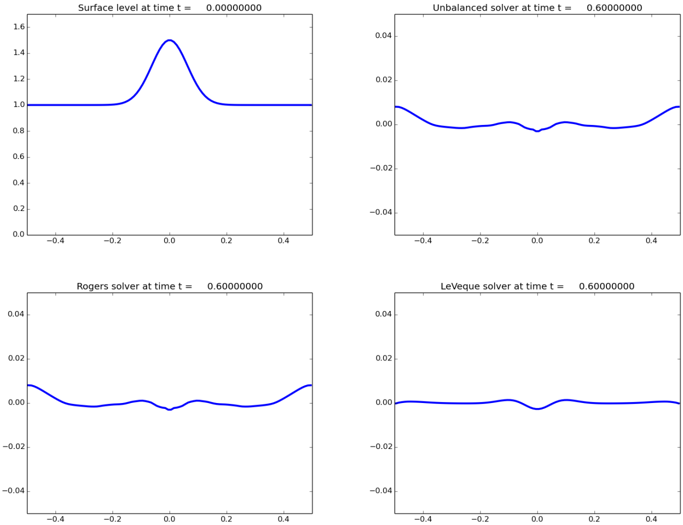
Results

Geostrophic equilibrium



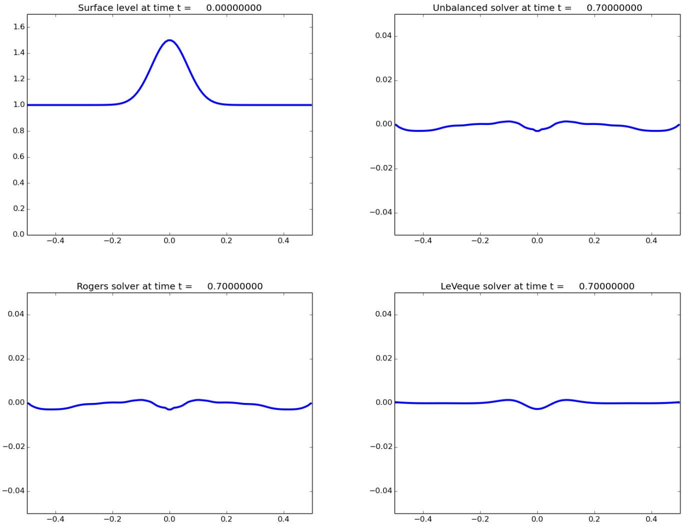
Results

Geostrophic equilibrium



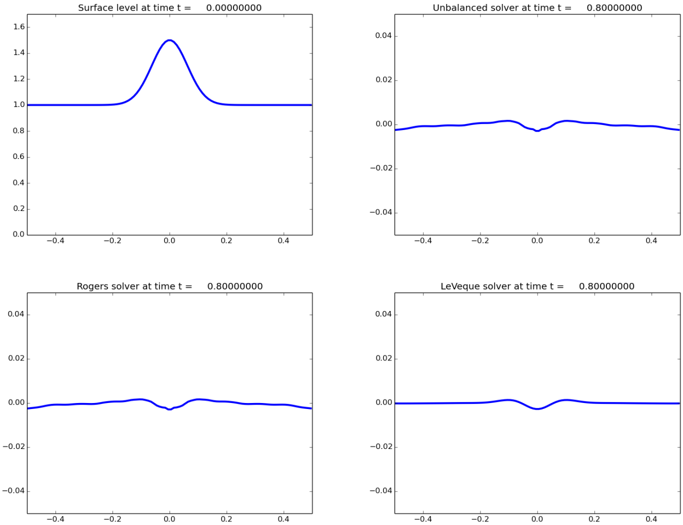
Results

Geostrophic equilibrium



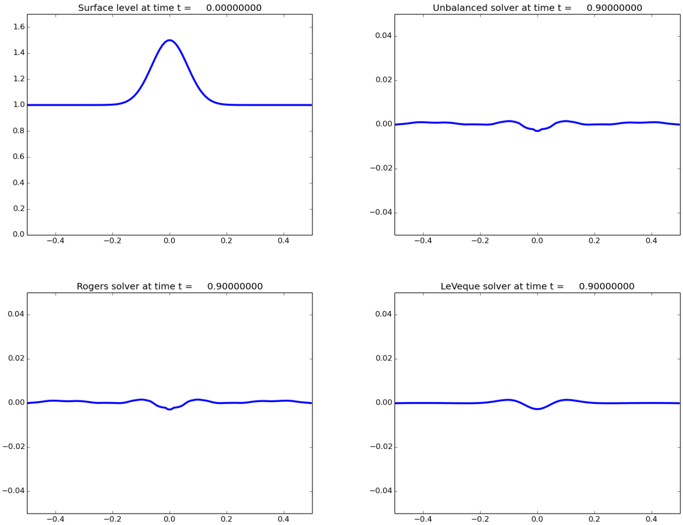
Results

Geostrophic equilibrium



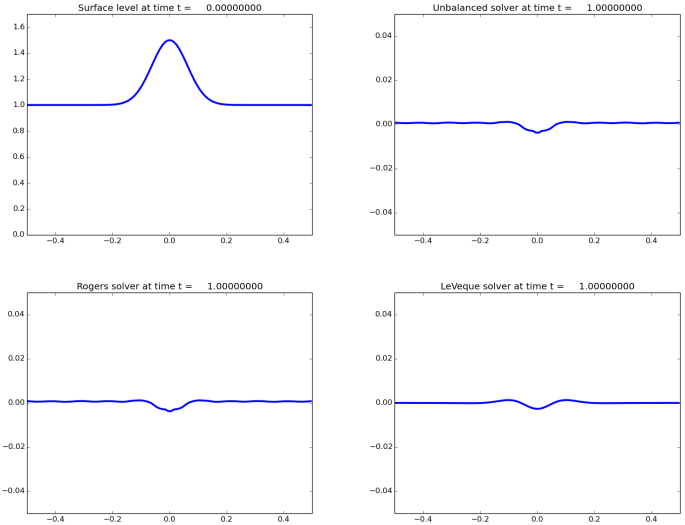
Results

Geostrophic equilibrium



Results

Geostrophic equilibrium



Conclusions

- LeVeque solver
  - Pro:
    - Well-balanced for all equilibria
  - Con:
    - Requires solving cubic for offset
    - Problematic away from equilibrium (transcritical flow)

Conclusions

- LeVeque solver
  - Pro:
    - Well-balanced for all equilibria
  - Con:
    - Requires solving cubic for offset
    - Problematic away from equilibrium (transcritical flow)
- Rogers solver
  - Pro:
    - Works away from equilibrium
    - Fairly simple to derive and implement
  - Con:
    - Tailored to a specific equilibrium
    - Less accurate than unbalanced method away from equilibrium

Conclusions

- LeVeque solver
  - Pro:
    - Well-balanced for all equilibria
  - Con:
    - Requires solving cubic for offset
    - Problematic away from equilibrium (transcritical flow)
- Rogers solver
  - Pro:
    - Works away from equilibrium
    - Fairly simple to derive and implement
  - Con:
    - Tailored to a specific equilibrium
    - Less accurate than unbalanced method away from equilibrium
- There is no free lunch!

Further work

- What's left?
  - Fix the bathymetry discretisation for LeVeque solver.
  - Get Rogers solver working for geostrophic equilibrium.

Further work

- What's left?
  - Fix the bathymetry discretisation for LeVeque solver.
  - Get Rogers solver working for geostrophic equilibrium.
- Future research
  - Allow for dry states
  - Look at non-Godunov type methods (Chertock et al., 2014, "Well-Balanced Schemes for the Shallow Water Equations with Coriolis Forces")

Questions?

Godunov's Method

- Wave propagation form:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}) \qquad (18)$$

$$\mathcal{A}^-\Delta Q_{i-1/2} = \sum_{p=1}^{M_w}(s_{i-1/2}^p)^-\mathcal{W}_{i-1/2}^p \qquad (19)$$

$$\mathcal{A}^+\Delta Q_{i+1/2} = \sum_{p=1}^{M_w}(s_{i+1/2}^p)^+\mathcal{W}_{i+1/2}^p \qquad (20)$$