Development of a Coastal Inundation Model using a Triangular Discontinuous Galerkin Method

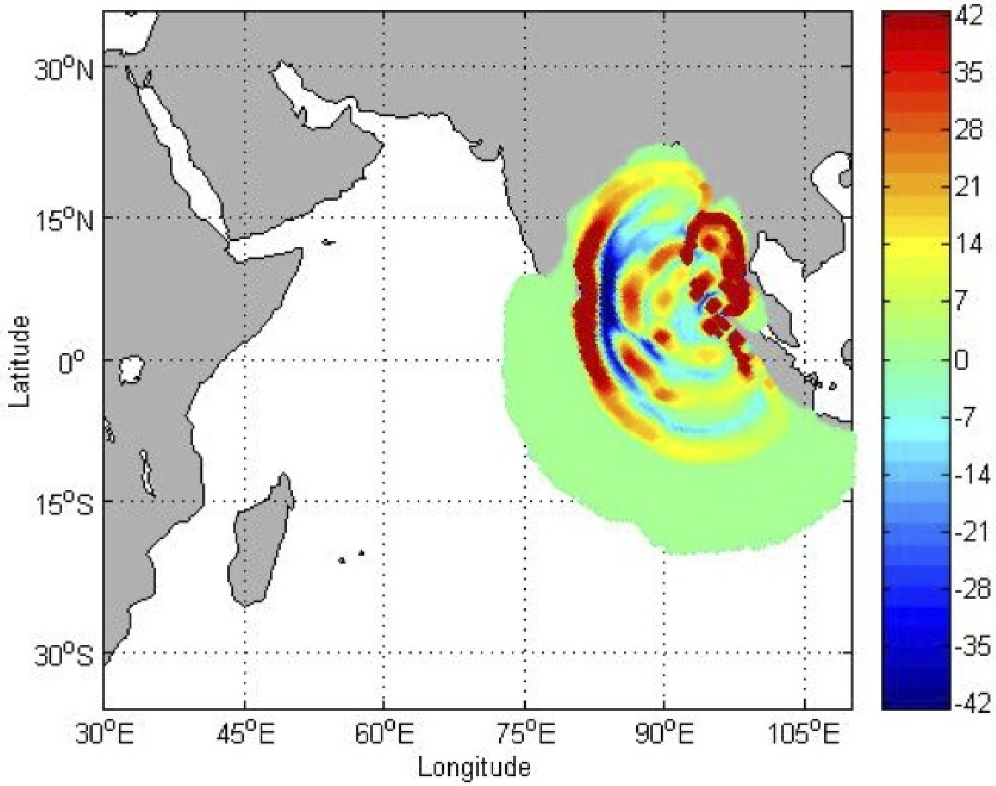
Shivasubramanian Gopalakrishnan
Francis X. Giraldo

Department of Applied Mathematics
Naval Postgraduate School, Monterey, California

21st October 2010

Motivation

Numerical modeling of storm surges and tsunamis



Alevras 2009

©Anders Garwin 2006

Outline

- Discontinuous Galerkin Method applied to SWE
- Coastal Ocean Modeling
- Adaptive Mesh Refinement

Shallow Water Equations

$$\frac{\partial q}{\partial t} + \nabla \cdot F(q) = S(q), \textit{ where } \; q = (\phi, U)^T$$

$$F(q) = \begin{pmatrix} U \\ \frac{U \otimes U}{\phi} + \frac{1}{2}\left(\phi^2 - \phi_b^2\right) I_2 \end{pmatrix}$$

$$S(q) = -\begin{pmatrix} 0 \\ f(k \times U) - \phi_s \nabla \phi_b - \frac{\tau}{\rho H} + \gamma U \end{pmatrix}$$

where,

$$\phi = g(h_s + h_b)$$

$$U = \phi \bar{u}$$

h_s- free surface height, h_b- bathymetry
$g-$ gravitational acceleration
$f = f_0 + \beta(y - y_m) -$ Coriolis parameter
$\tau-$ wind stress, $\gamma-$ bottom friction

Discontinuous Galerkin Method

The domain Ω is decomposed into N_e conforming elements.

$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e$$

For the operators, a non−singular mapping

$$x = \Psi\left(\xi\right)$$

transforms the physical coordinate system $X = \left(x, y\right)^T$ to local reference coordinate system $\xi = \left(\xi, \eta\right)^T$.

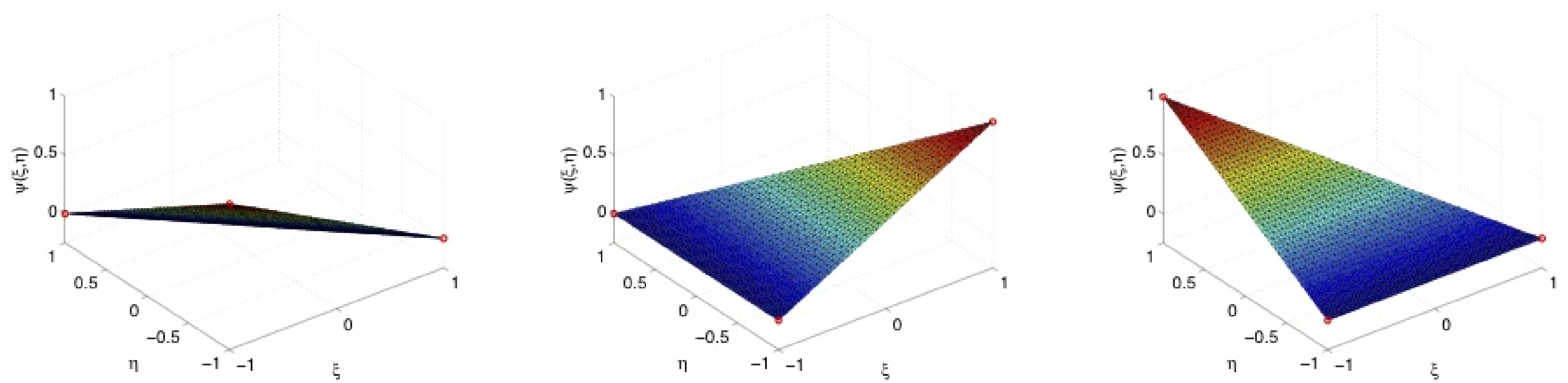
Discontinuous Galerkin Method

The local elementwise solution is approximated by Nth order polynomial in ξ by

$$q_N(\xi) = \sum_{i=1}^{M_n} \psi_i(\xi)\, q_N(\xi_i)$$

where $M_n = \frac{1}{2}(N+1)(N+2)$ is the number of interpolation points and $\psi_i(\xi)$ the associated Lagrange polynomials.

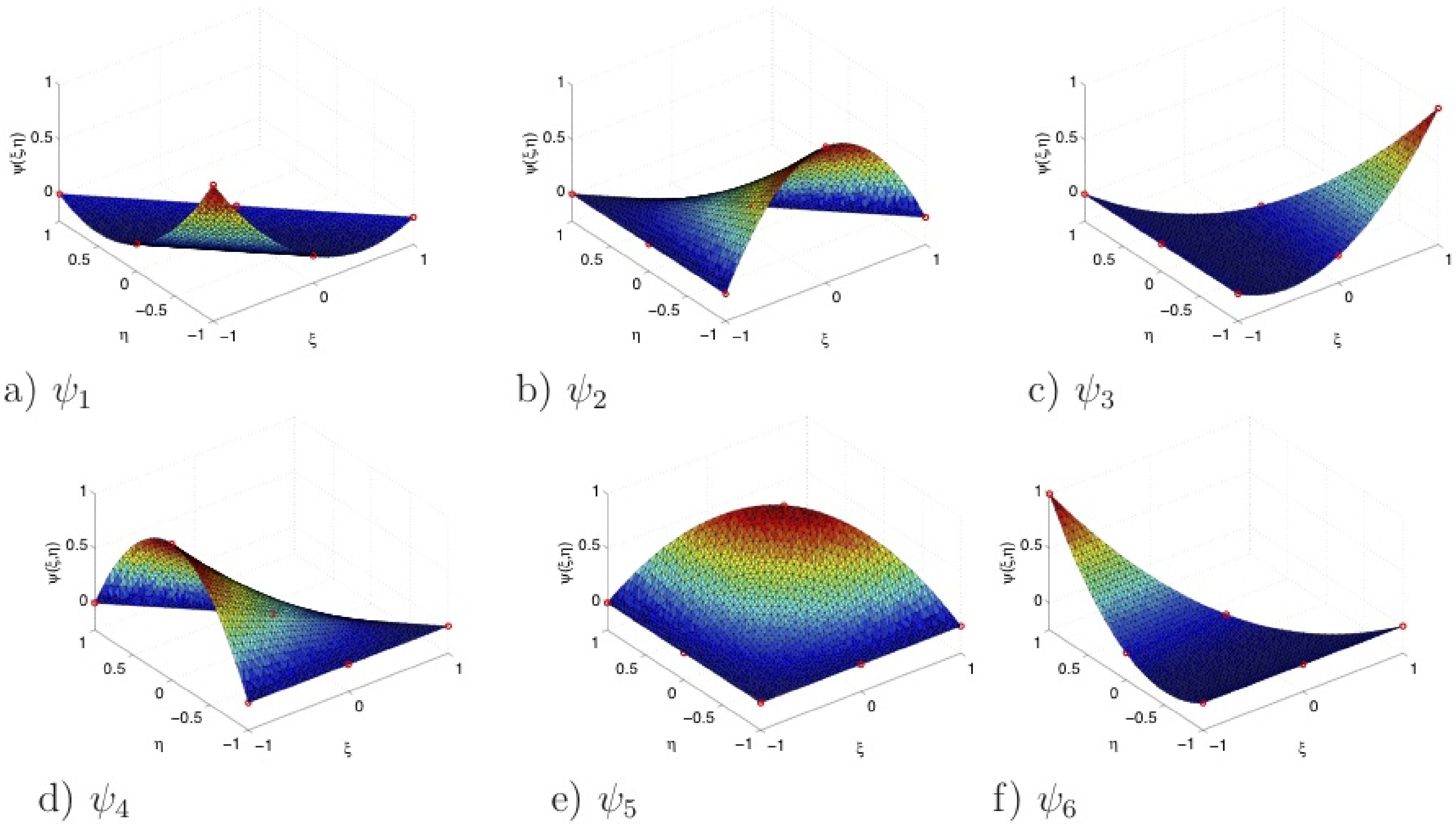
Triangular basis functions



a) ψ_1 b) ψ_2 c) ψ_3

Triangular basis functions of order N=1 at 3 interpolation points.

Triangular basis functions

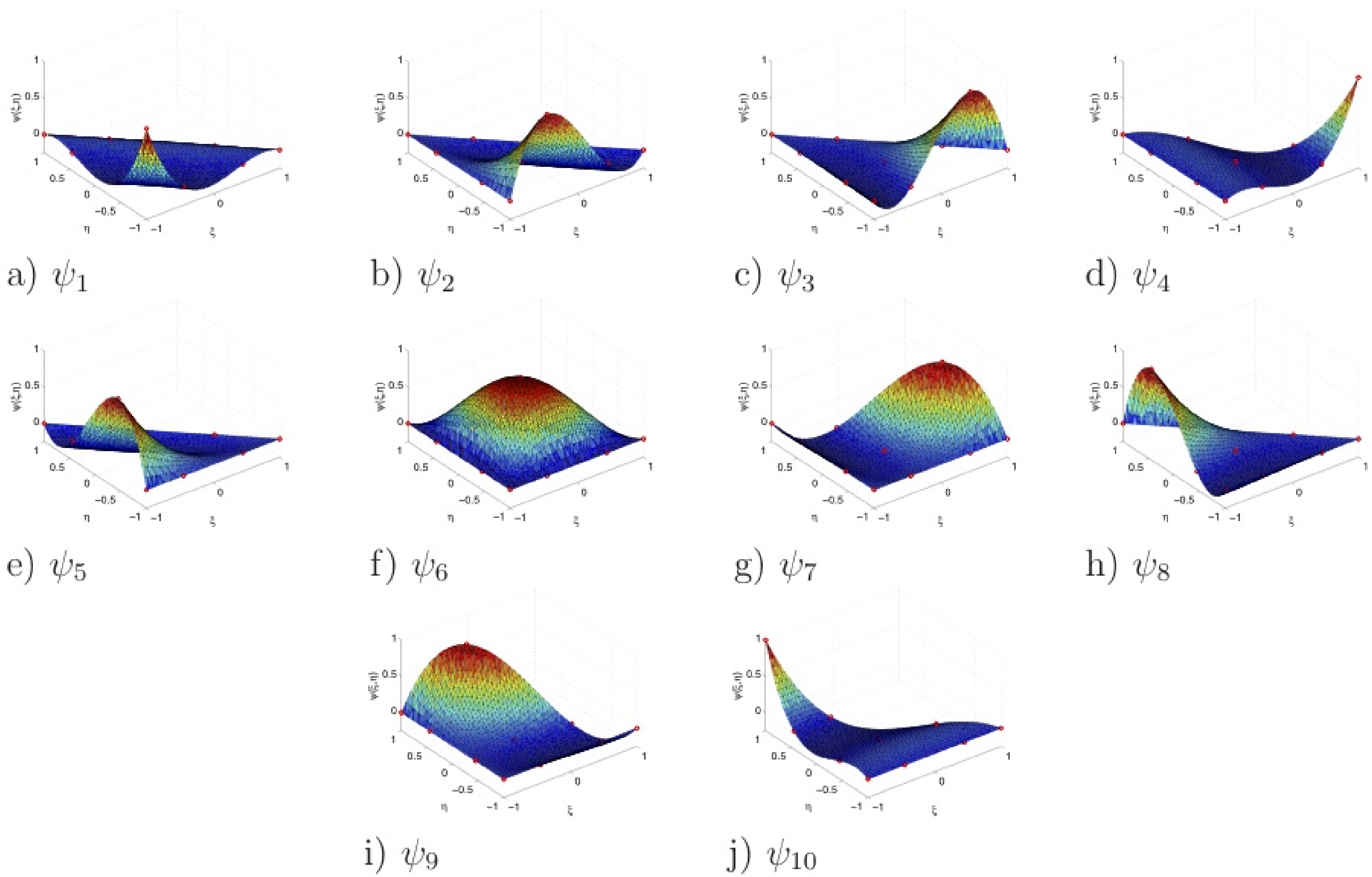


a) ψ_1 b) ψ_2 c) ψ_3

d) ψ_4 e) ψ_5 f) ψ_6

Triangular basis functions of order N=2 at 6 interpolation points.

Triangular basis functions



Triangular basis functions of order N=3 at 10 interpolation points.

Discontinuous Galerkin Method

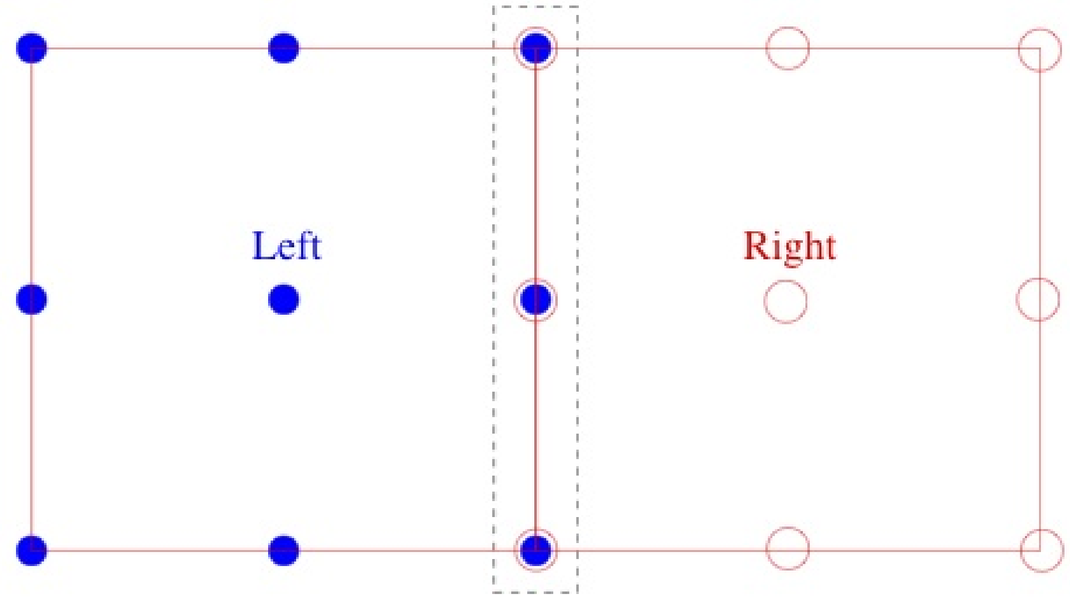
Applying DG to the Shallow water equations to obtain the weak form

$$\int_{\Omega_e} \left(\frac{\partial q_N^{(e)}}{\partial t} - F_N^{(e)} \cdot \nabla - S_N^{(e)} \right) \psi_i (x) \, dx$$
$$= - \sum_{l=1}^{3} \int_{\Gamma_e} \psi_i (x) \, n^{(e,l)} \cdot F_N^{(*,l)} dx$$

Integrating the above equation by parts,

$$\int_{\Omega_e} \psi_i (x) \left(\frac{\partial q_N^{(e)}}{\partial t} + \nabla \cdot F_N^{(e)} - S_N^{(e)} \right) dx$$
$$= \sum_{l=1}^{3} \int_{\Gamma_e} \psi_i (x) \, n^{(e,l)} \cdot \left(F_N^{(e)} - F_N^{(*,l)} \right) dx$$

Discontinuous Galerkin Method



Rusanov Numerical Flux

$$F_N^{(*,l)} = \frac{1}{2}\left[F_N\left(q_N^{(e)}\right) + F_N\left(q_N^{(l)}\right) - |\lambda^{(l)}|\left(q_N^{(l)} - q_N^{(e)}\right) n^{(e,l)}\right]$$

Where,

$$\lambda^{(l)} = max\left(|U^{(e)}| + \sqrt{\phi^{(e)}}, |U^{(l)}| + \sqrt{\phi^{(l)}}\right)$$

with,

$$U^{(e,l)} = u^{(e,l)} \cdot n^{(l)}$$

Matrix form of semi−discrete equations

Using the polynomial approximation $q_N = \sum_{i=1}^{M_N} \psi_i q_i$

$$\int_{\Omega_e} \psi_i \psi_j dx \frac{\partial q_j^{(e)}}{\partial t} + \int_{\Omega_e} \psi_i \nabla \psi_j dx \cdot F_j^{(e)} - \int_{\Omega_e} \psi_i \psi_j dx S_j^{(e)}$$
$$= \sum_{l=1}^{3} \int_{\Gamma_e} \psi_i \psi_j n^{(e,l)} dx \cdot \left(F^{(e)} - F^{(*,l)} \right)_j$$

Defining element matrices as

$$M_{ij}^{(e)} = \int_{\Omega_e} \psi_i \psi_j dx, \; M_{ij}^{(e,l)} = \int_{\Gamma_e} \psi_i \psi_j n^{(e,l)} dx, \; D_{ij}^{(e)} = \int_{\Omega_e} \psi_i \nabla \psi_j dx$$

Matrix form of semi−discrete equations

Eliminating mass matrix on LHS

$$\hat{D}^{(e)} = \left(M^{(e)}\right)^{-1} D^{(e)}, \hat{M}^{(e,l)} = \left(M^{(e)}\right)^{-1} M^{(e,l)}$$

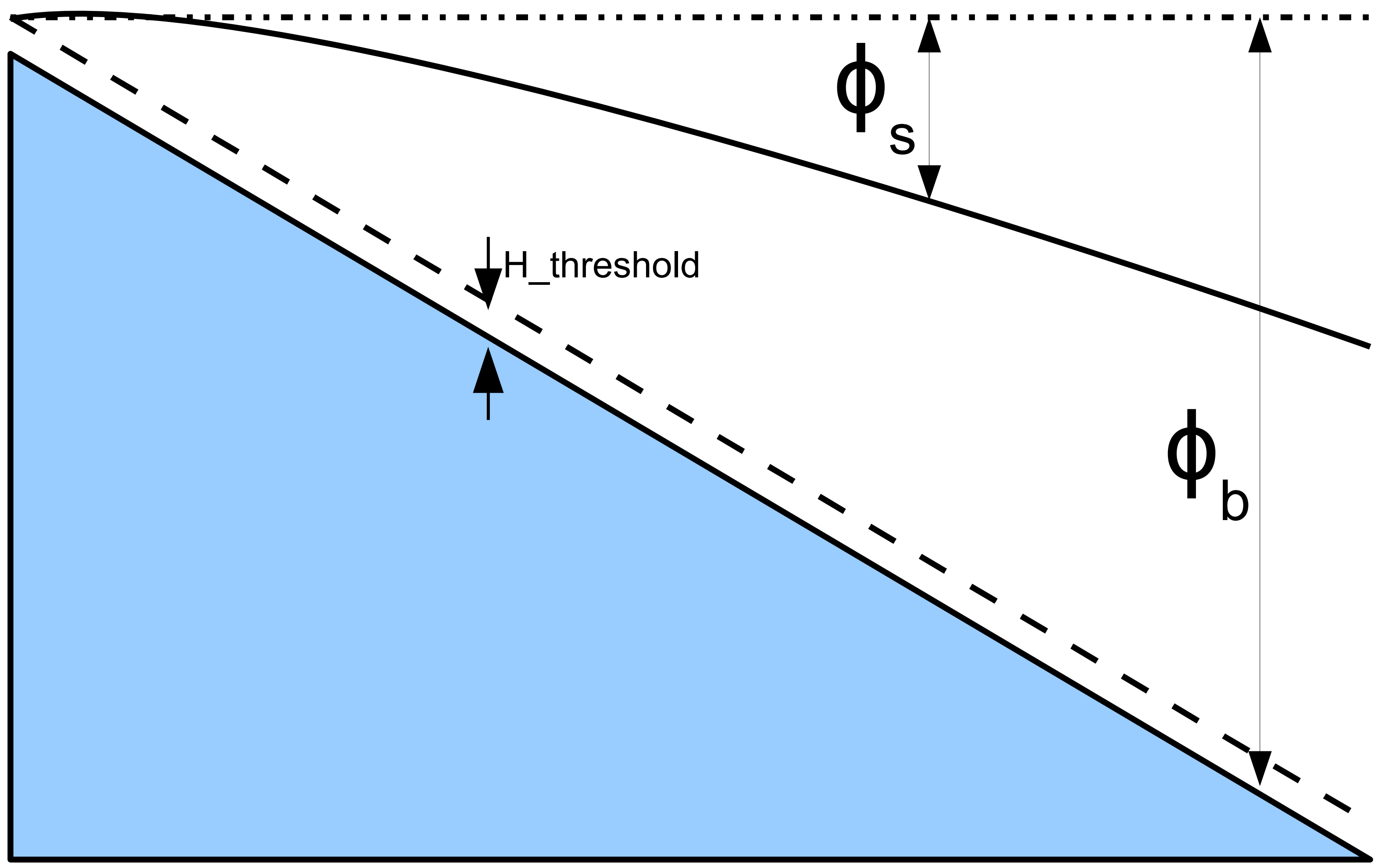
$$\frac{\partial q_i^{(e)}}{\partial t} + \left(\hat{D}_{ij}^{(e)}\right)^T F_j^{(e)} - S_i^{(e)} = \sum_{l=1}^{3} \left(\hat{M}_{ij}^{(e,l)}\right)^T \left(F^{(e)} - F^{(*,l)}\right)_j$$

Inundation Modeling

Coastal ocean modeling

- The shoreline is represented as a moving boundary condition where $\phi = \phi_s + \phi_b = 0$
- Moving front is described as $x = x_b + \int v_b dt$.
- Where x_b is initial position and v the velocity of the front.
- Approaches used to model the wetting and drying of land.
  - Fixed grid methods. – Easier to implement. Additional algorithms required to maintain depth positivity.
  - Moving grid methods. – traditionally perceived as cumbersome. (Lynch and Gray 1978)

Wetting and Drying Algorithm



Wetting and Drying Algorithm – based on Gourgue et al 2009

Conservation of Mass

$$\frac{\partial \phi_s}{\partial t} = -F(U)$$

where $\phi = \phi_s + \phi_b$, and operator $F(U) \equiv F(\phi_s, \bar{u})$

- Step 1 – limit ϕ to a threshold value.

$$\phi_s^* = max\left(\phi_s^n, H_{threshold} - \phi_b\right)$$

- Step 2

$$\frac{\phi_s^{**}}{\Delta t} = -F(\phi_s^*, \bar{u})$$

- Step 3 – ensure free surface does not move to dry areas.

$$\frac{\phi_s^{n+1}}{\Delta t} = -F^*(\phi_s^*, \bar{u})$$

Wetting and Drying Algorithm – based on Gourgue et al 2009

Conservation of Mass in matrix form

$$\frac{\partial \phi_i^{**(e)}}{\partial t} = -\left(\hat{D}_{ij}^{(e)}\right)^T F_j^{(e)} + \sum_{l=1}^{3} \left(\hat{M}_{ij}^{(e,l)}\right)^T \left(F^{(e)} - F^{(*,l)}\right)_j$$

Let,

$$F_j^s\left(\phi_s^*, \bar{u}\right) = -\left(\hat{D}_{ij}^{(e)}\right)^T F_j^{(e)}$$

$$F_j^c\left(\phi_s^*, \bar{u}\right) = \sum_{l=1}^{3} \left(\hat{M}_{ij}^{(e,l)}\right)^T \left(F^{(e)} - F^{(*,l)}\right)_j$$

Wetting and Drying Algorithm – based on Gourgue et al 2009

$$\frac{\phi_s^{n+1}}{\Delta t} = F_j^{c*}\left(\phi_s^*, \bar{u}\right) + F_j^{s*}\left(\phi_s^*, \bar{u}\right)$$

Where,

$$F_j^{c*} = \begin{cases} 0 \ if \ F_c^j < 0 \ \& \ \phi^n < H_{threshold} \\ F_c^j \ otherwise \end{cases}$$

$$F_j^{s*} = \begin{cases} 0 \ if \ there \ is \ a \ node \ i \ \in \ \Omega_e \ with \ F_s^j < 0 \ \& \ \phi < H_{threshold} \\ F_s^j \ otherwise \end{cases}$$

*limited to using linear elements.

Steady state test

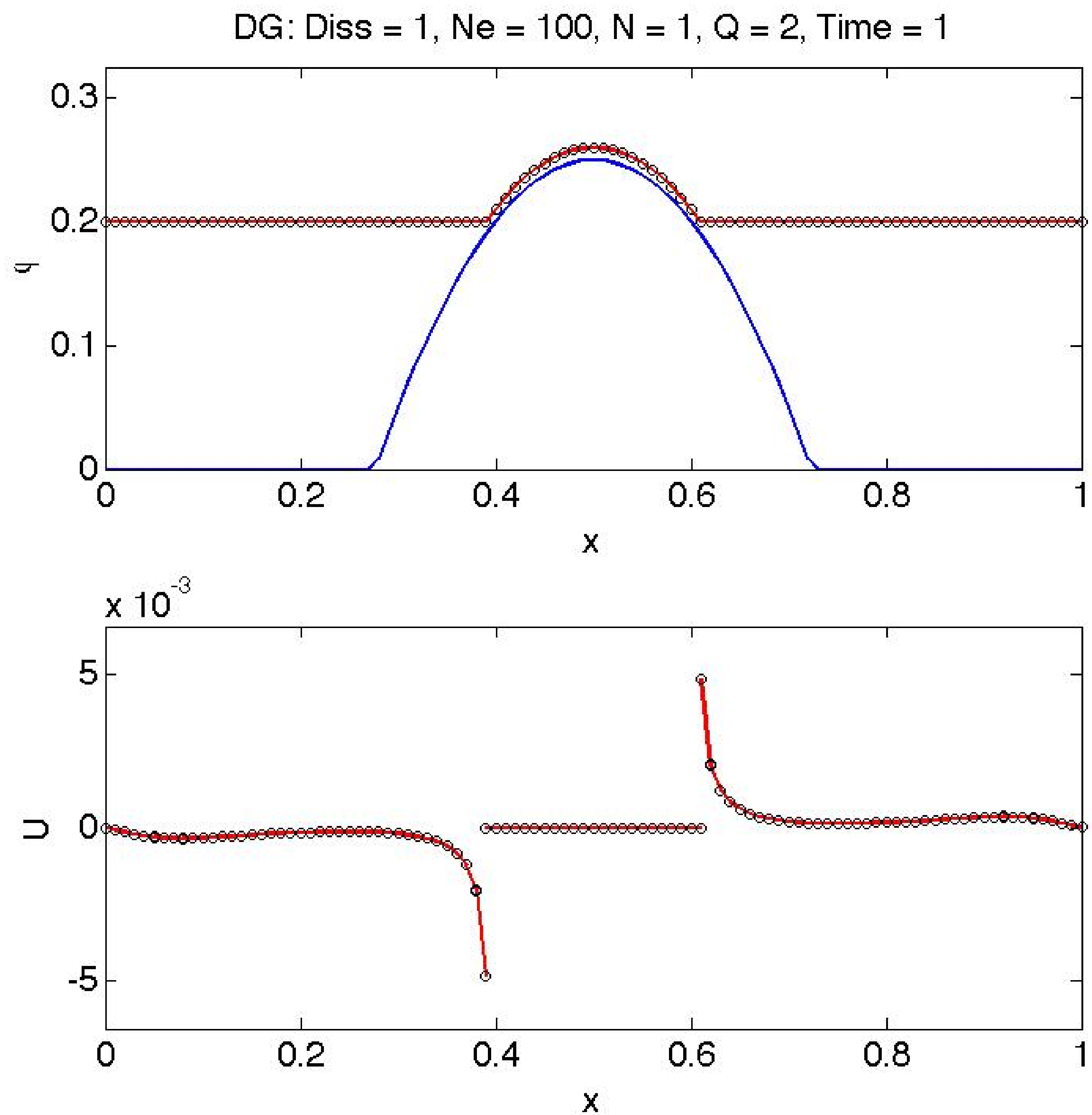
Is the model well balanced ?
Bottom topography is defined as,

$$h_b(x) = max\left(0, 0.25 - 5\left(x - 0.5^2\right)\right), 0 \leq x \;\; eq \; 1$$

Initial condition

$$h_s + h_b = max\,(0.2, b)$$
$$\phi U = 0 \quad \text{over entire domain}$$

Steady state test case



Balzano [1998] Test Case 1

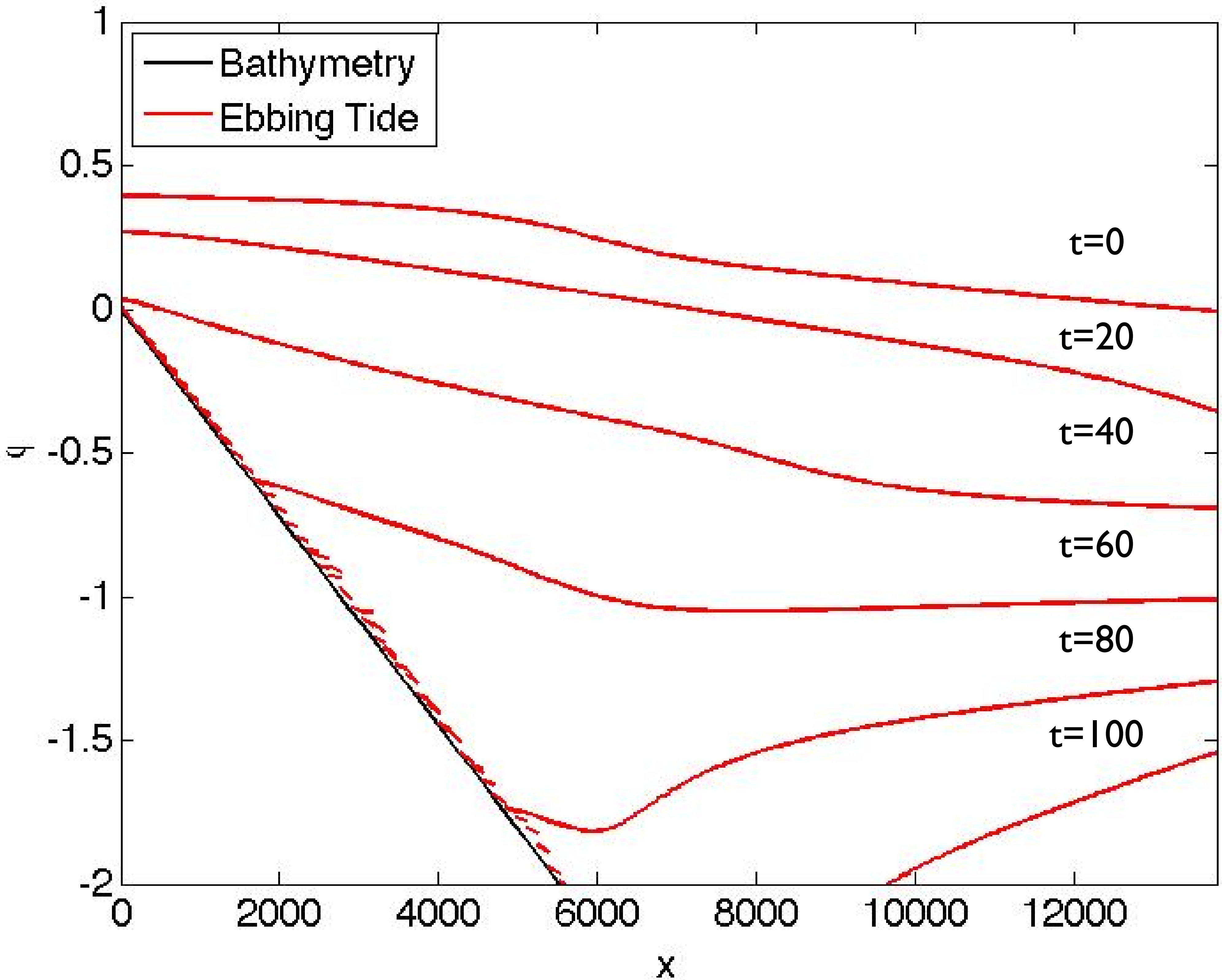
Bottom topography is defined as,

$$h_b(x) = \frac{x}{2760}$$

Domain size is 13,800 meters. Sinusoidal forcing at the open end is given by,

$$\phi_s = g * (2sin\left(\frac{2\pi t}{43200}\right))$$

Balzano [1998] Test Case 1



Balzano [1998] Test Case 2

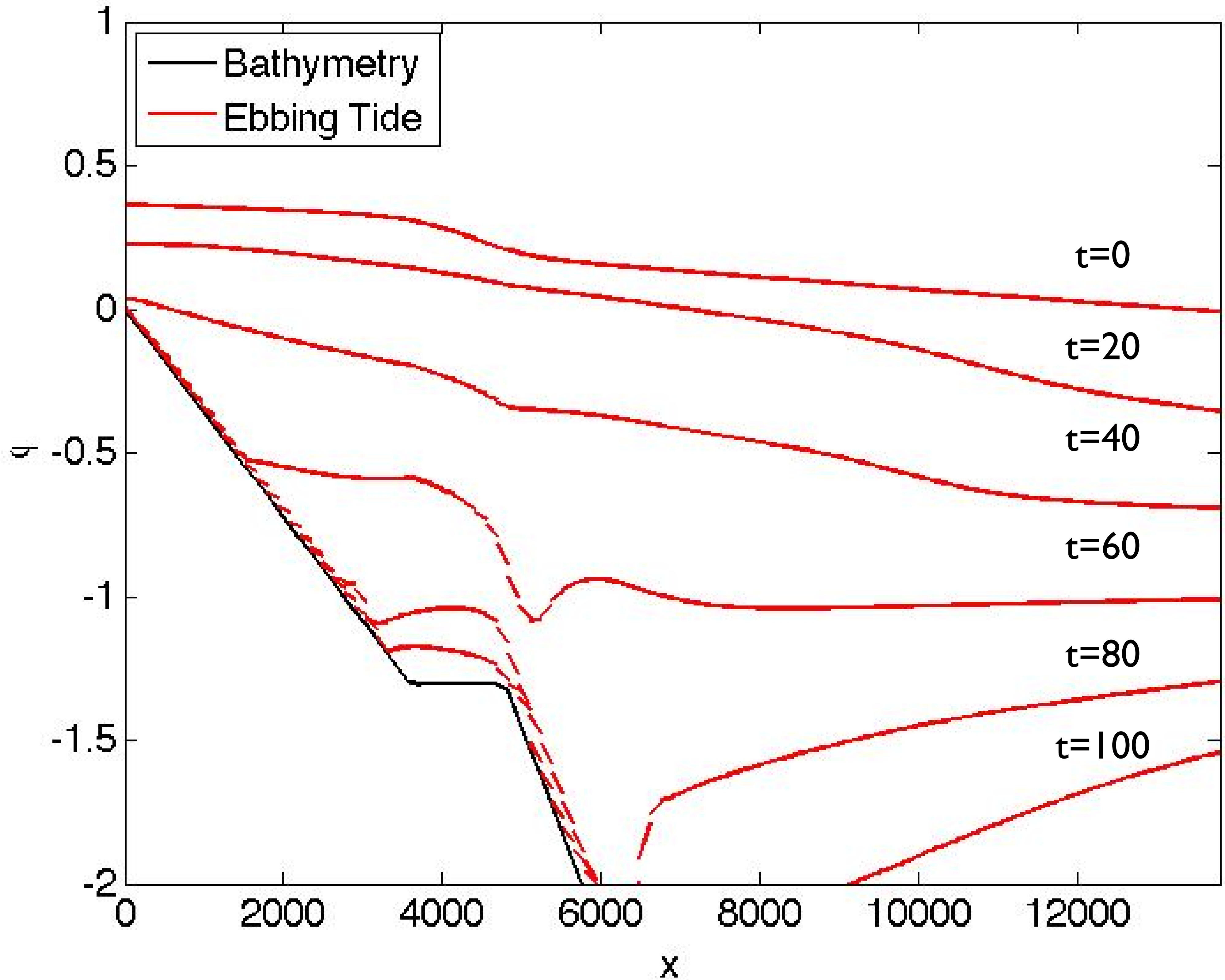
Bottom topography is defined as,

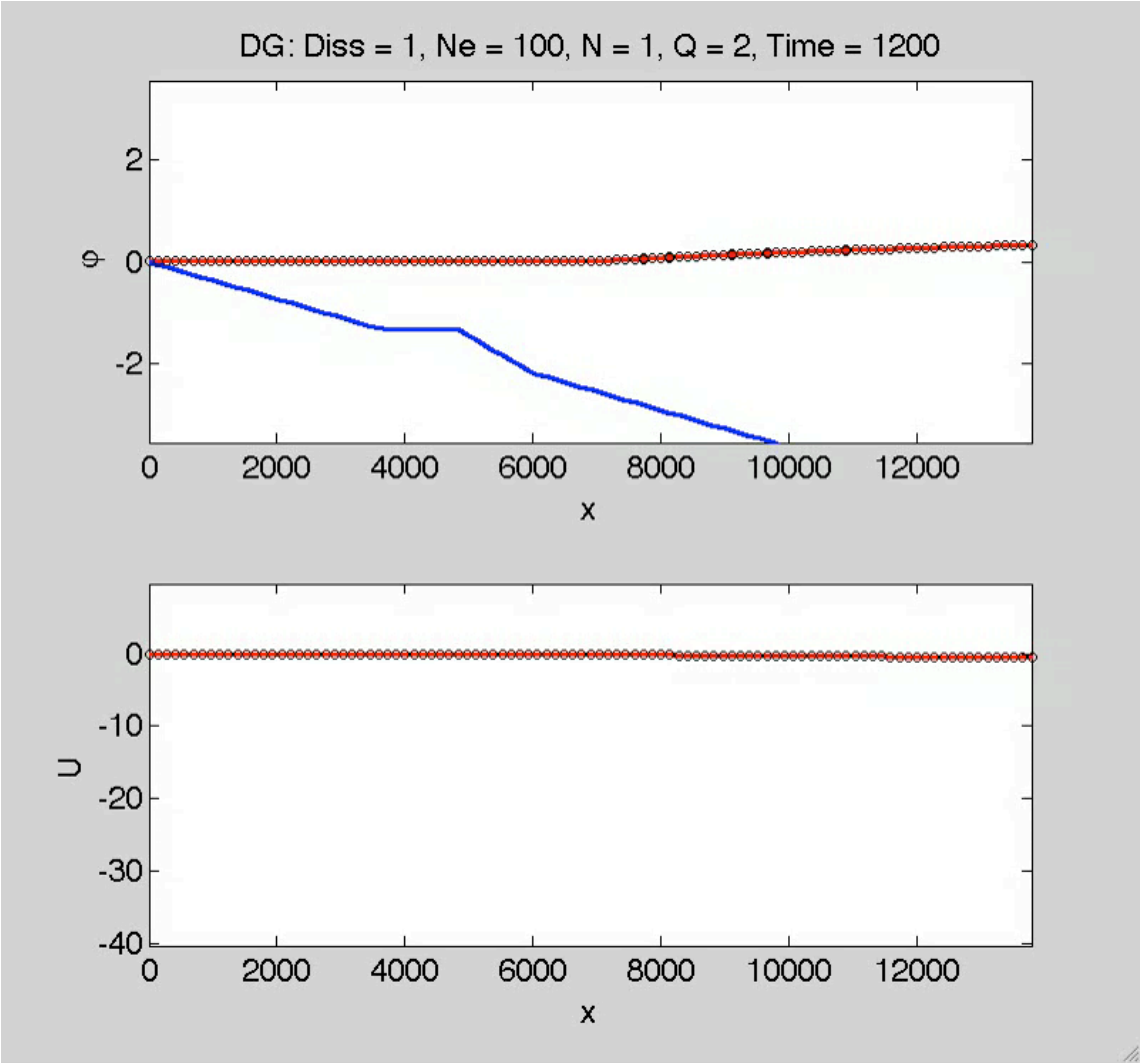
$$h_b(x) = \begin{cases} \frac{x}{2760} & \text{if x} \leq \text{3600m, or if x} \geq \text{6000m} \\ \frac{30}{23} & \text{if 3600m} \leq \text{x} \leq \text{4800m} \\ \frac{x}{920} - \frac{100}{23} & \text{if 4800m} \leq \text{x} \leq \text{6000m} \end{cases}$$

Domain size is 13,800 meters. Sinusoidal forcing at the open end is given by,

$$\phi_s = g * (2sin\left(\frac{2\pi t}{43200}\right))$$

Balzano [1998] Test Case 2





DG: Diss = 1, Ne = 100, N = 1, Q = 2, Time = 1200
φ
2
0
-2
0
2000
4000
6000
8000
10000
12000
x
U
0
-10
-20
-30
-40
0
2000
4000
6000
8000
10000
12000
x

Balzano [1998] Test Case 3

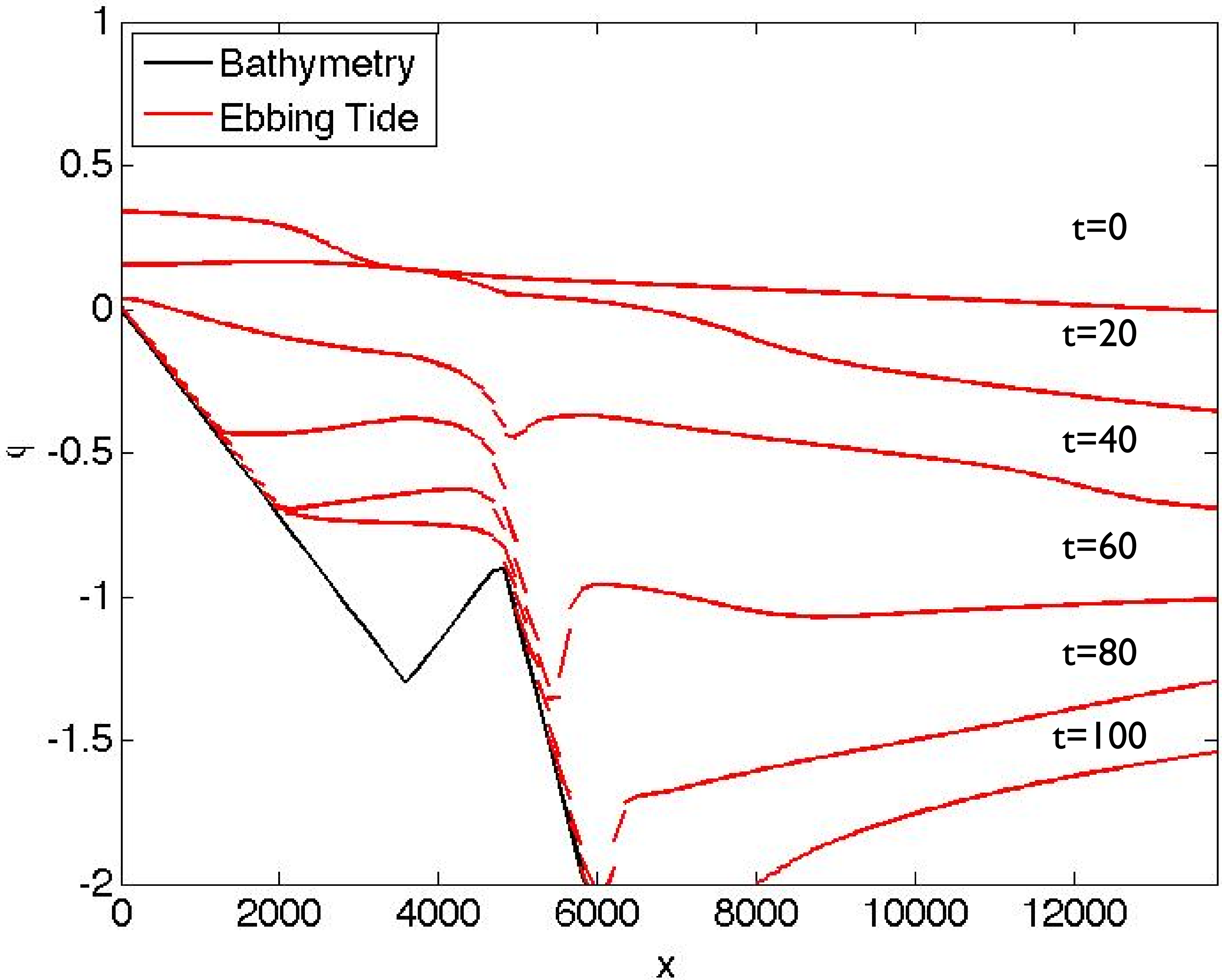
Bottom topography is defined as,

$$h_b(x) = \begin{cases} \frac{x}{2760} & \text{if x} \leq \text{3600m, or if x} \geq \text{6000m} \\ \frac{-x}{2760} + \frac{60}{23} & \text{if 3600m} \leq \text{x} \leq \text{4800m} \\ \frac{x}{920} - \frac{100}{23} & \text{if 4800m} \leq \text{x} \leq \text{6000m} \end{cases}$$

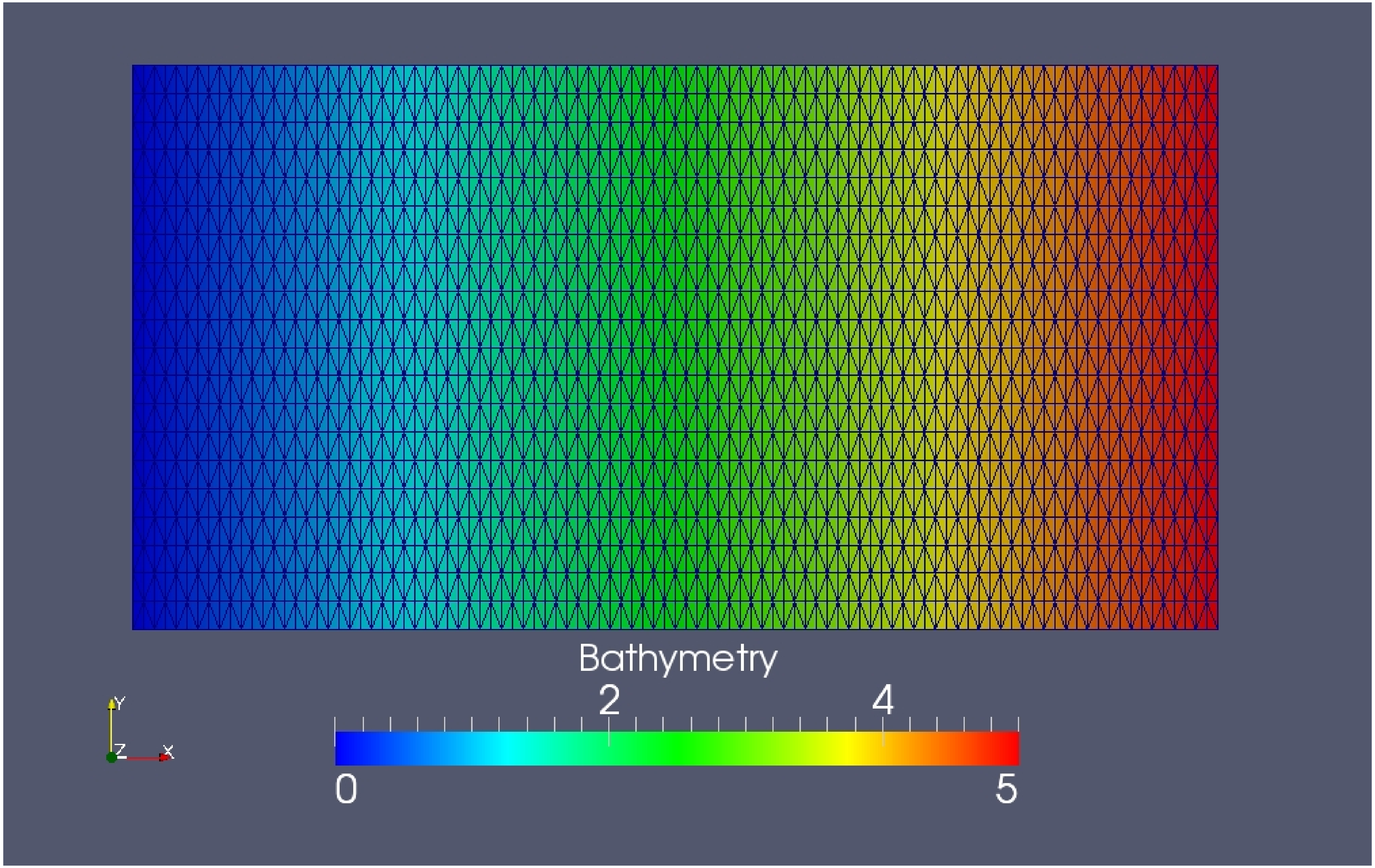
Domain size is 13,800 meters. Sinusoidal forcing at the open end is given by,

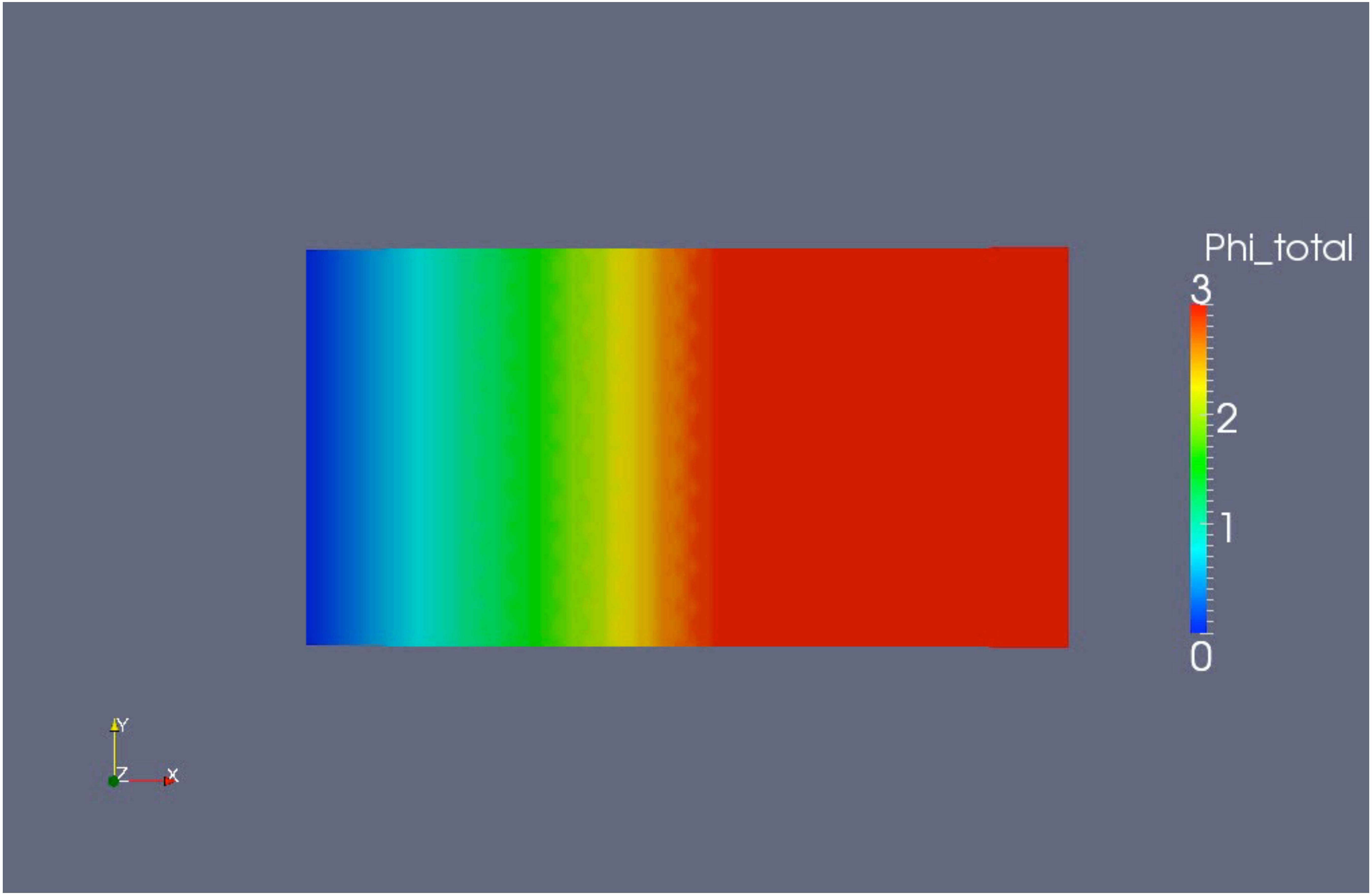
$$\phi_s = g * (2sin\left(\frac{2\pi t}{43200}\right))$$

Balzano [1998] Test Case 3

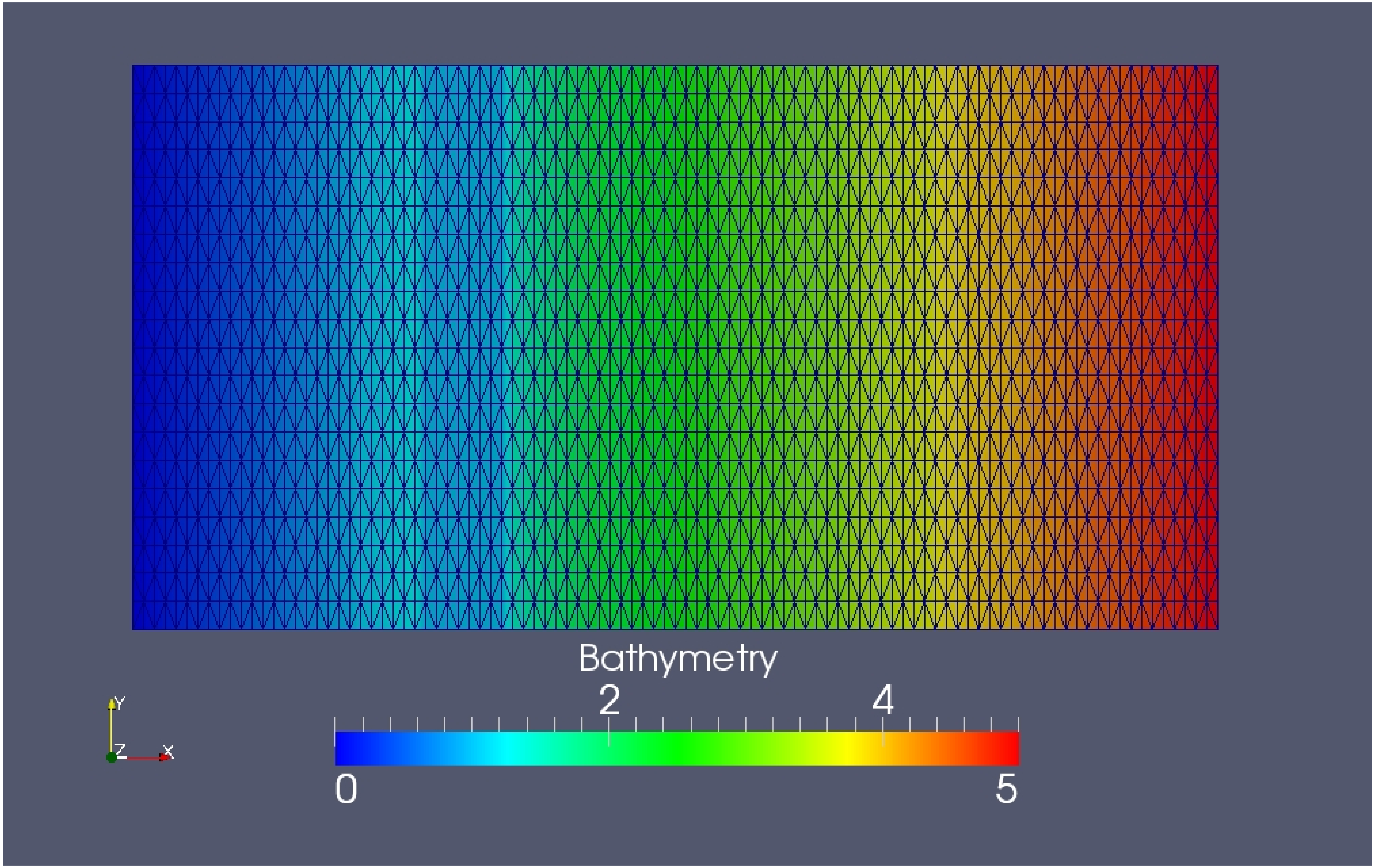


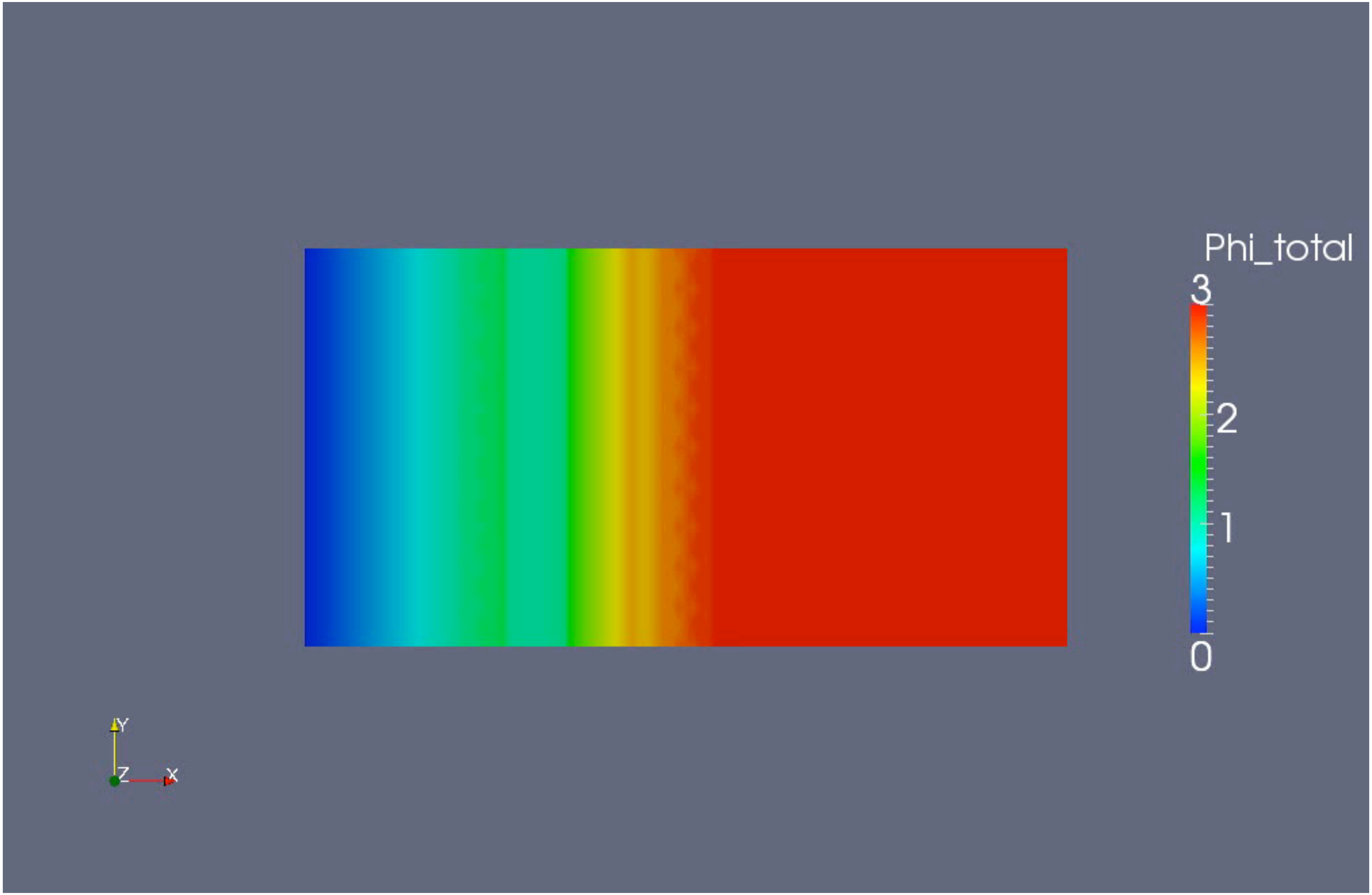
Balzano [1998] 2D Test Case 1




Phi_total
3
2
1
0
Y
Z
X

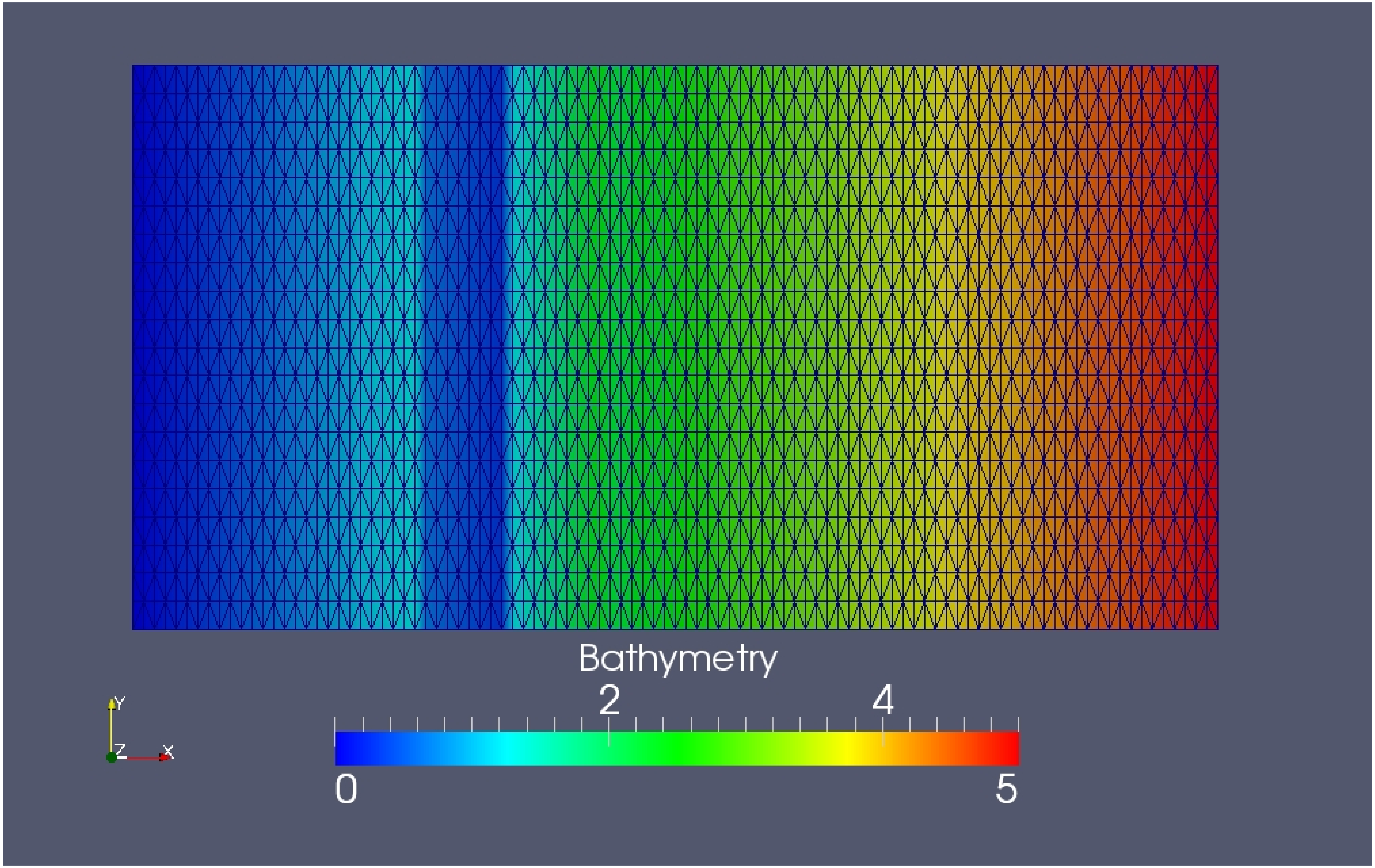
Balzano [1998] 2D Test Case 2

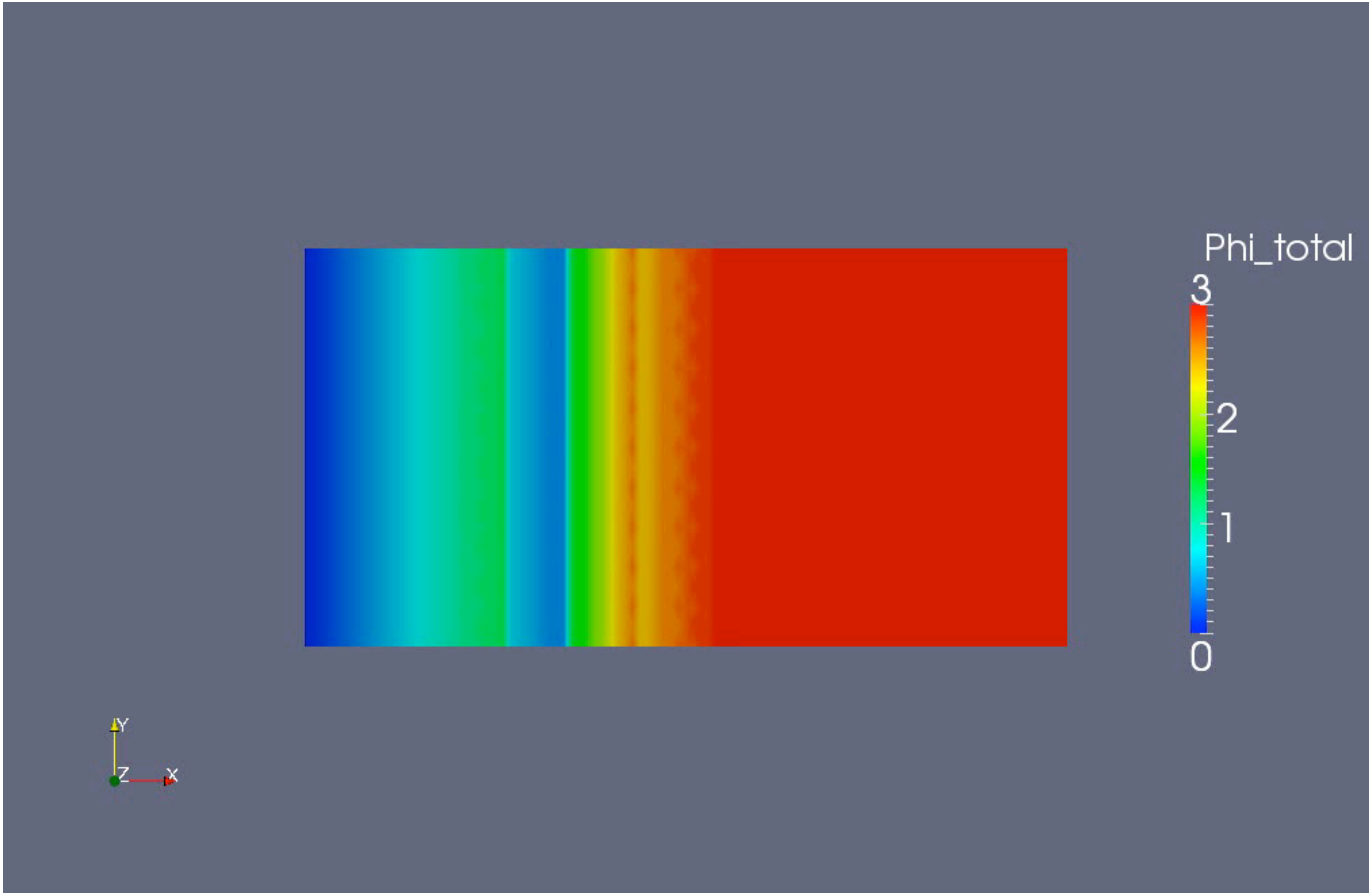




Phi_total
3
2
1
0
Y
Z
X

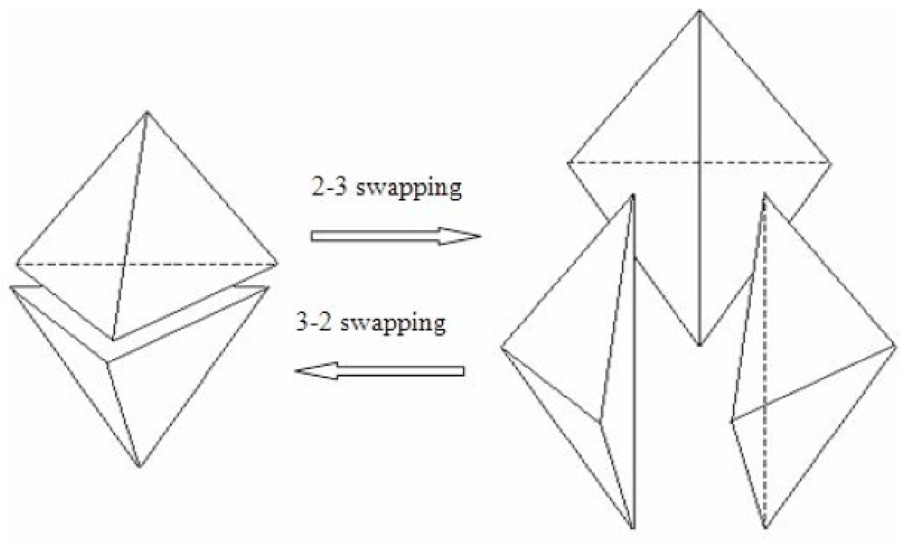
Balzano [1998] Test Case 3



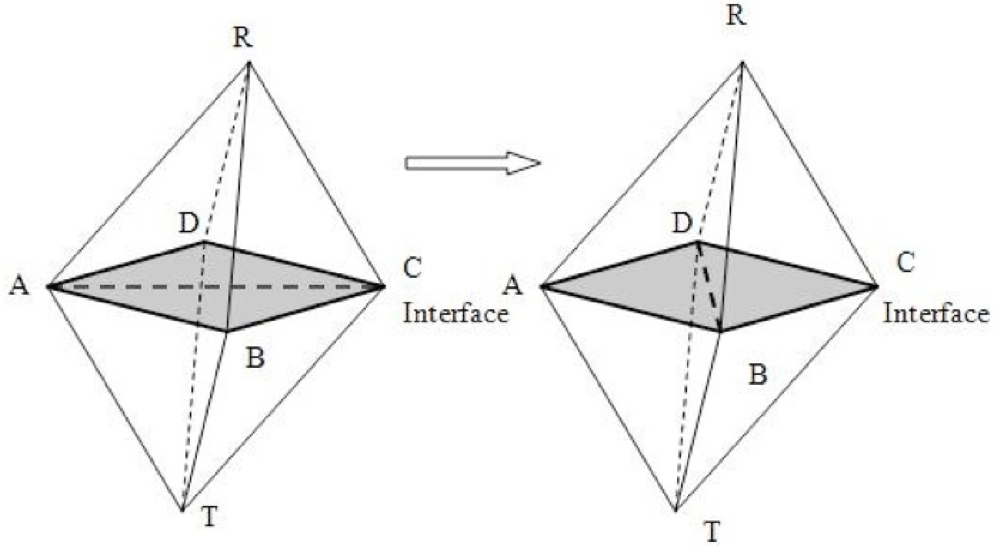

Phi_total
3
2
1
0
Y
Z
X

Adaptive Mesh Refinement

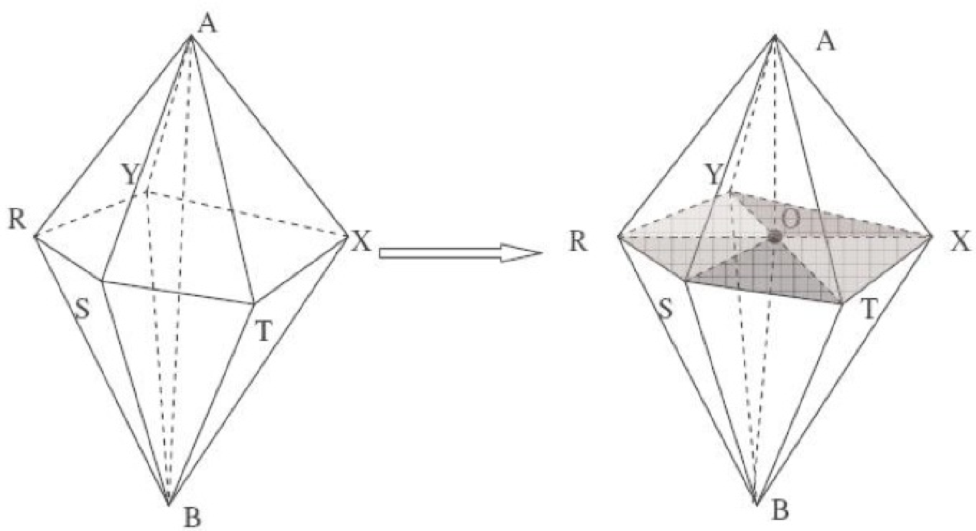
Adaptive Mesh Refinement



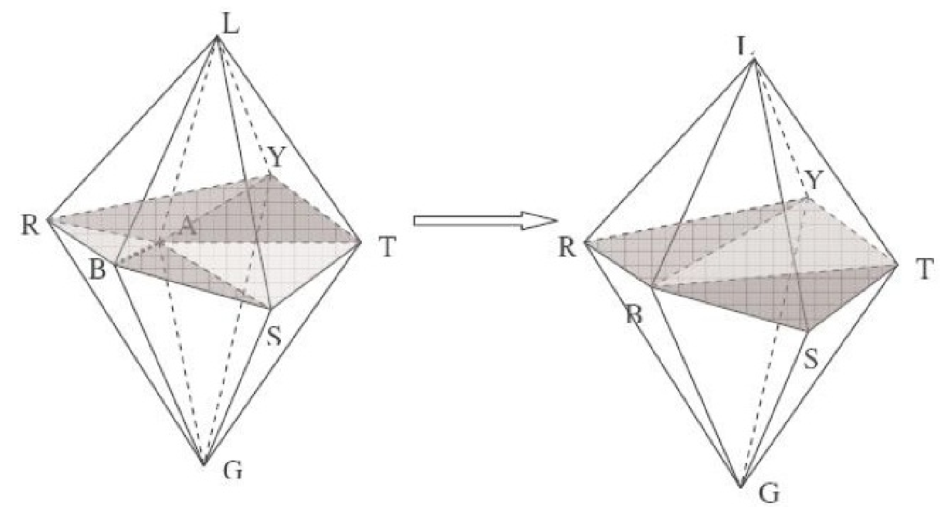
2–3 and 3–2 flipping



4–4 flipping

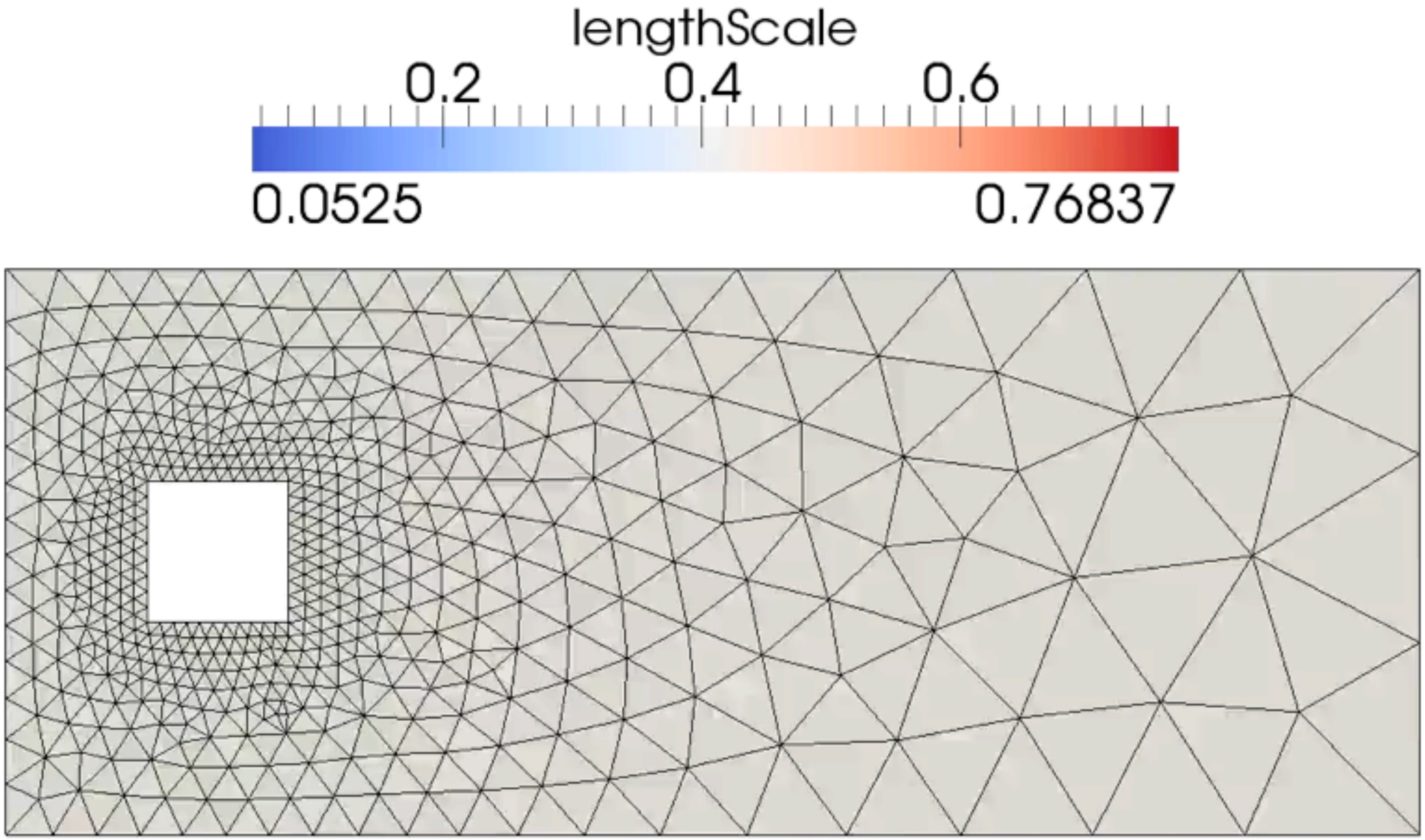


Edge bisection

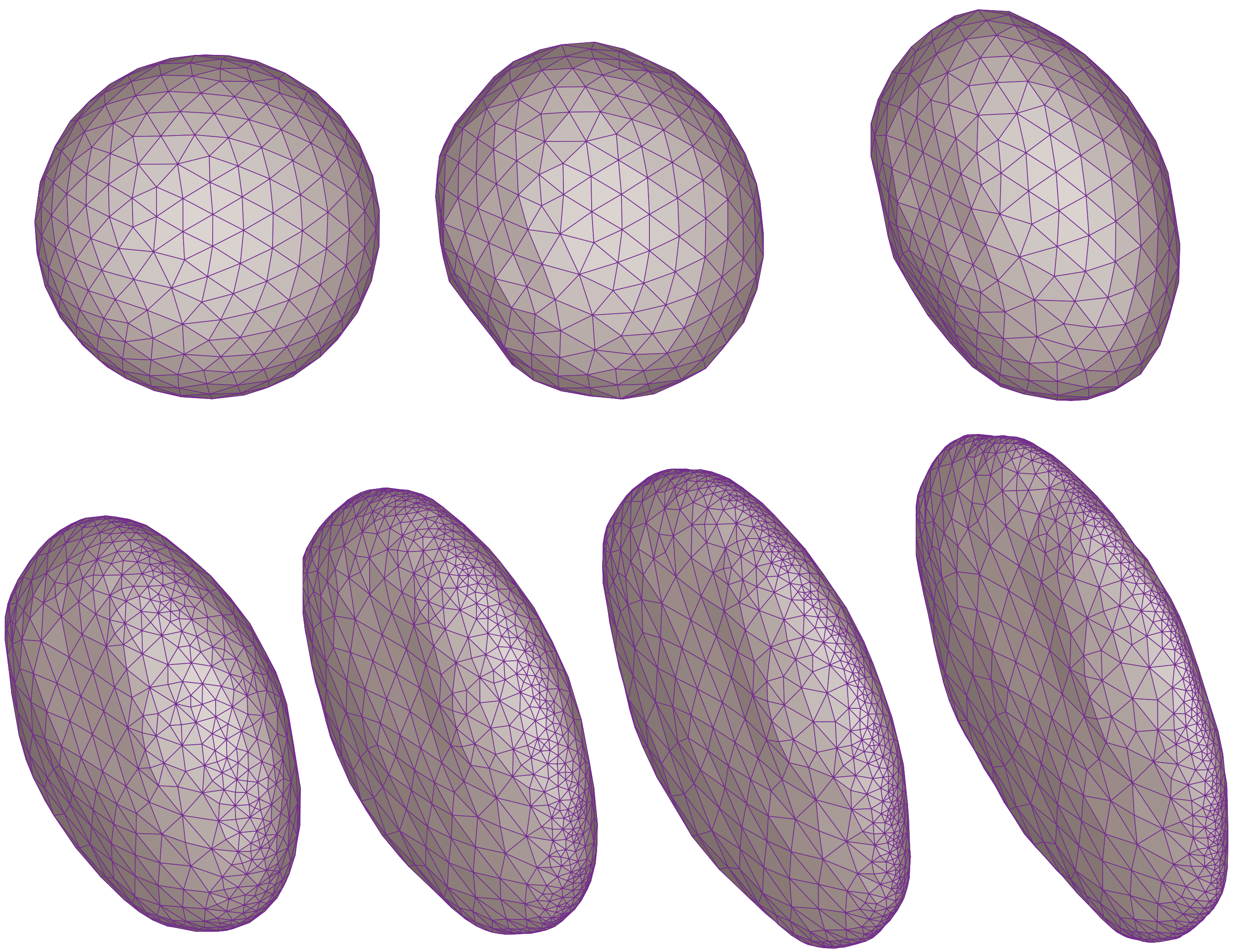


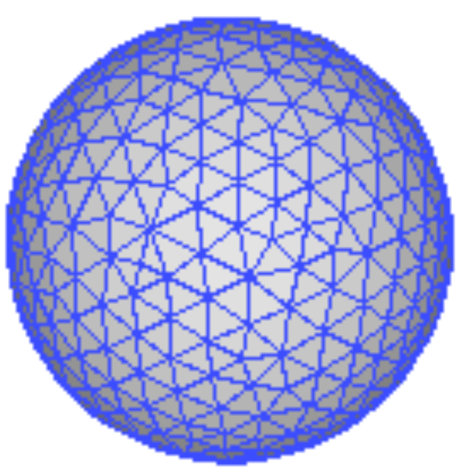
Edge collapse

2D Unstructured AMR – S. Menon (In Progress)



3D Unstructured AMR – Gopalakrishnan, Quan and Schmidt [2006]

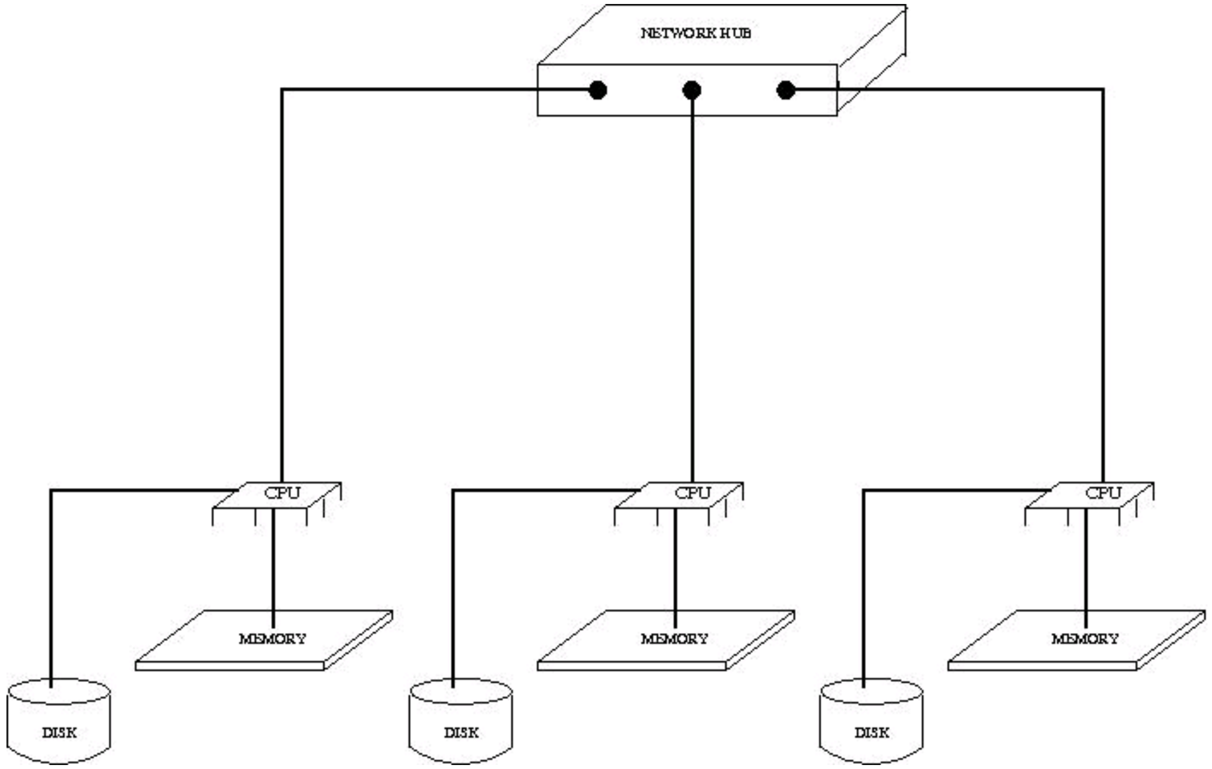




Distributed – Memory Machines

- Each node in the computer has a locally addressable memory space
- The computers are connected together via some high-speed network
  - Infiniband, Myrinet, Giganet, etc..

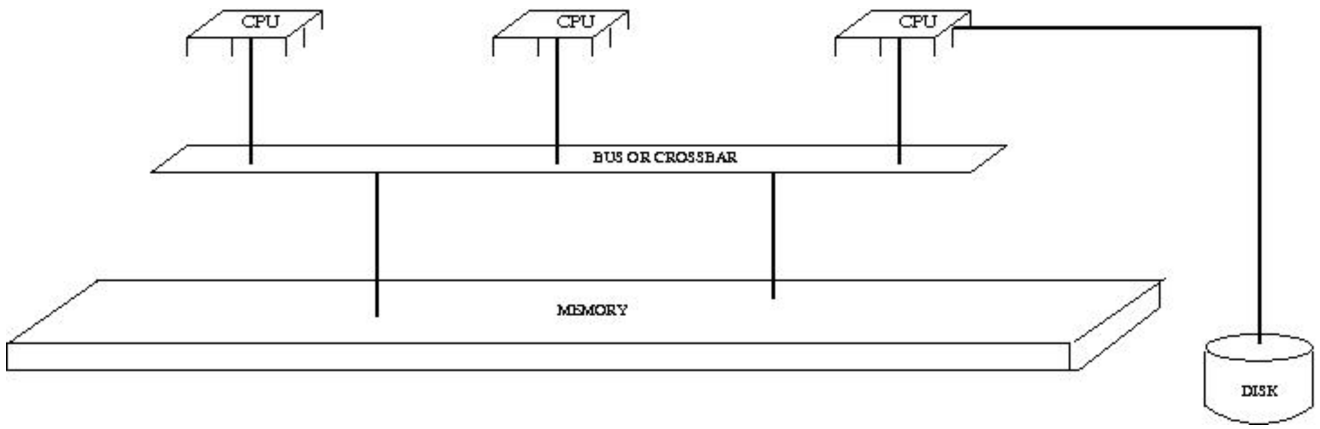
- Pros
  - Really large machines
  - Size limited only by gross physical considerations:
    - Room size
    - Cable lengths (10's of meters)
    - Power/cooling capacity
    - Money!
  - Cheaper to build and run
- Cons
  - Harder to program
    Data Locality



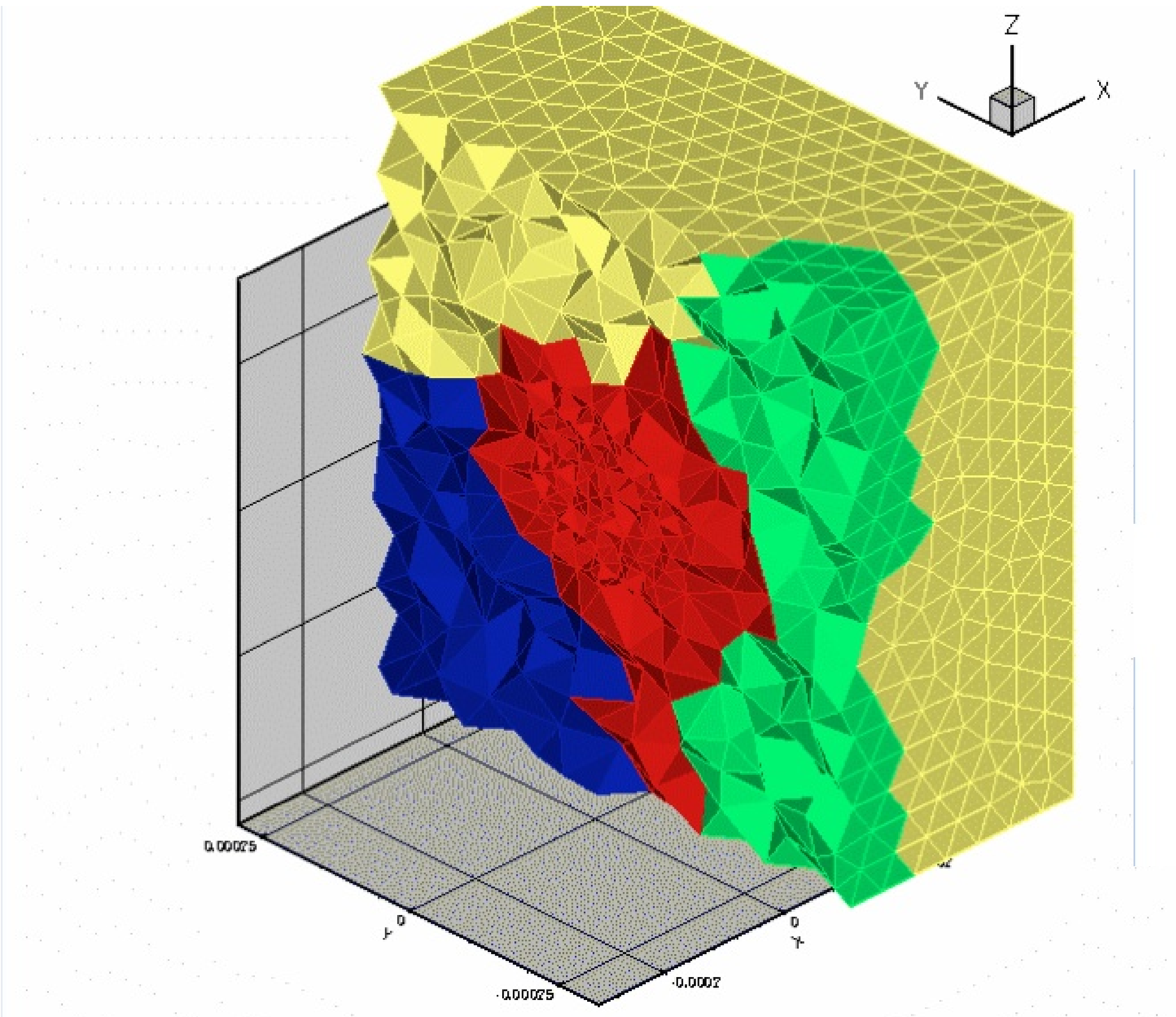
Shared-Memory Processing

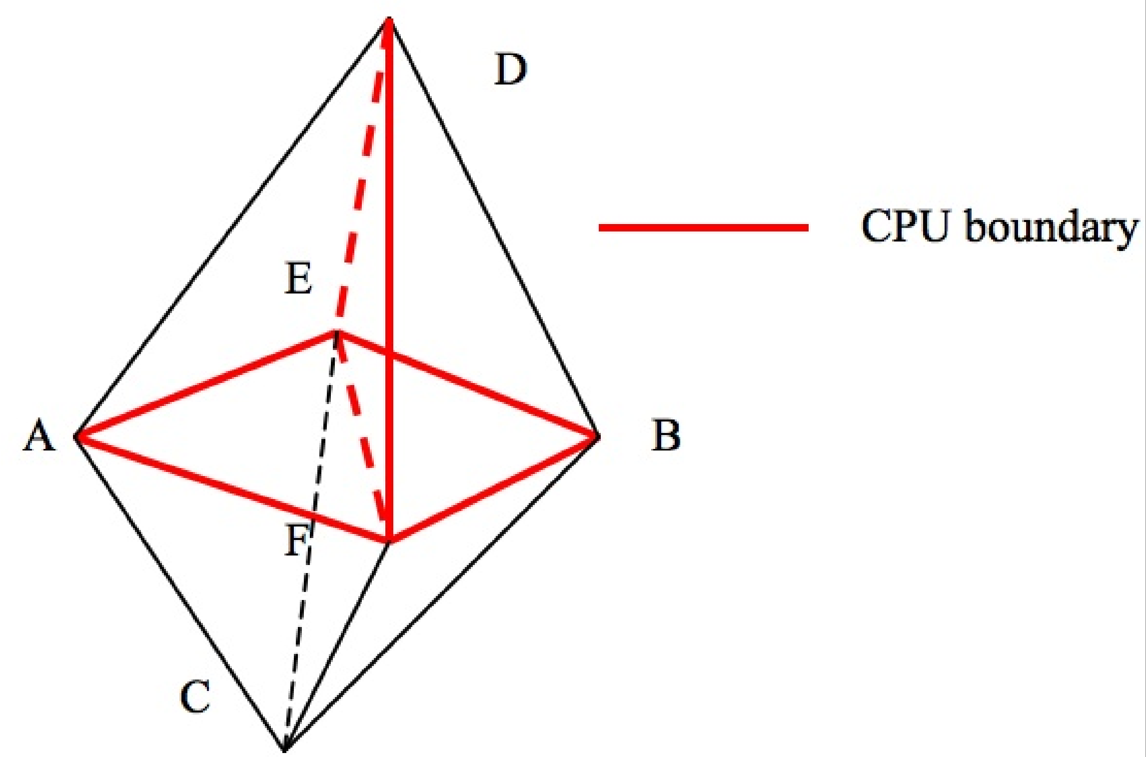
Each processor can access the entire data space

- Pro's
  - Easier to program
  - Amenable to automatic parallelism
  - Can be used to run large memory serial programs
- Con's
  - Expensive
  - Difficult to implement on the hardware level
  - Processor count limited by contention/coherency (currently around 512)
  - Watch out for "NU" part of "NUMA"



Parallel AMR





Conclusions and Future Directions

- DG provides a robust framework to implement SWE.
- Use of triangular DG method enables the modeling of complex coastlines.
- Adaptive mesh refinement helps in providing adequate resolution to resolve interesting features.
- Higher order Wetting and Drying Methods.
- Wind forcing data from mesoscale atmospheric codes - NUMA.
- Moving mesh technique may provide a useful alternative to model inundation as a moving boundary.
- Collaboration with Randy LeVeque and Kyle Mandli. Comparison and verification of test cases.