



**ĐẠI HỌC
BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

2D SHALLOW WATER EQUATIONS

Model the propagation of Tsunamis

ONE LOVE. ONE FUTURE.



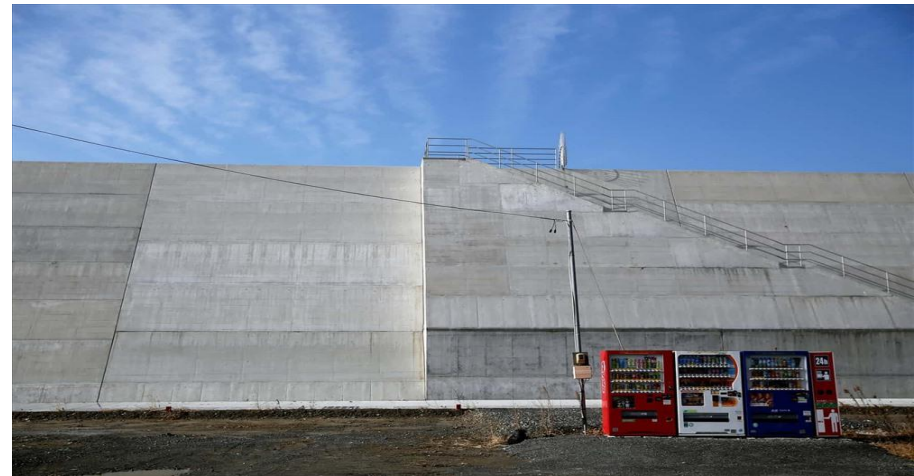
HUST

I. ĐẶT VẤN ĐỀ

Đặt vấn đề

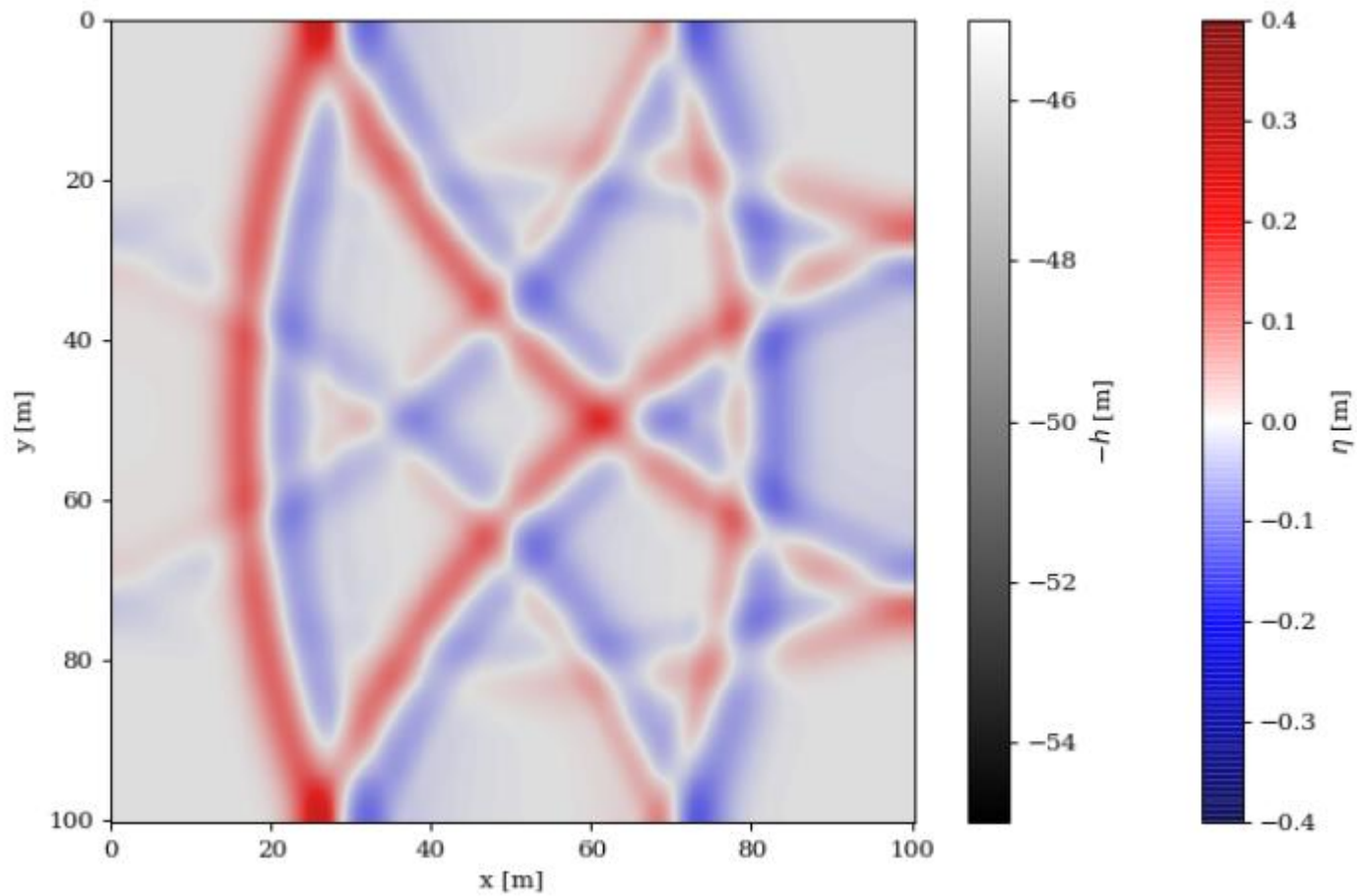
- ❖ Sóng thần là một loạt các đợt sóng tạo nên khi một thể tích lớn của nước ở đại dương bị chuyển dịch chớp nhoáng trên một quy mô lớn.
 - ❖ Hậu quả tai hại của sóng thần có thể ở mức cực lớn. Nó tàn phá, cuốn trôi nhà cửa, xe cộ, cơ sở vật chất và nhấn chìm hàng trăm ngàn người vài giờ trong nước.
 - ❖ Sóng thần không thể được dự đoán một cách hoàn toàn chính xác, nhưng có những dấu hiệu có thể báo trước một đợt sóng thần sắp xảy ra, và nhiều hệ thống đang được phát triển và được sử dụng để giảm thiểu những thiệt hại do sóng thần gây ra.
- ➔ Chúng ta sẽ thử mô phỏng một cơn sóng thần và xem cách chúng lan truyền trên đại dương như thế nào!**

Một vài hình ảnh liên quan đến sóng thần



- Tạo một mặt phẳng tọa độ 2D của các điểm trên bề mặt đại dương
- Với điều kiện đầu của một cơn sóng thần xuất hiện trên đại dương là độ cao mặt nước và lưu lượng theo 2 hướng x và y , chúng ta tính toán sau một khoảng thời gian độ cao tại các điểm mặt nước thay đổi thế nào, và mô hình hóa chúng.
- Sử dụng phương trình nước nông 2D (2D Shallow Water Equations) để tính toán các thông số tại các điểm trên bề mặt nước.

Hình ảnh mô phỏng

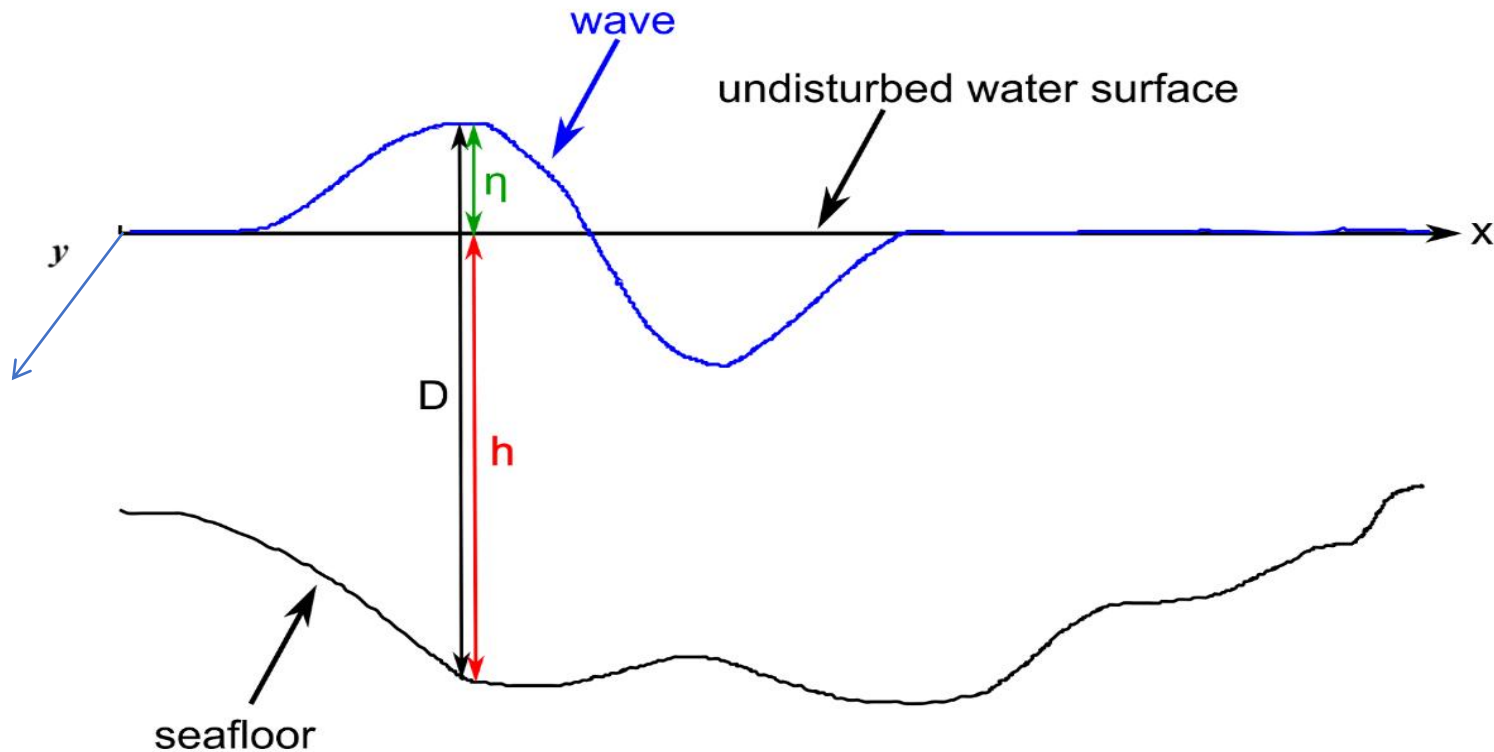




HUST

II. CƠ SỞ LÝ THUYẾT

Cơ sở lý thuyết



Tại mỗi điểm (x,y) trên bề mặt đại dương, h là chiều cao từ đáy biển lên mặt nước tĩnh, η là biên độ của sóng, $D = h + \eta$ là chiều cao của cột nước tại điểm đó

❖ Phương trình nước nông 2D

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0$$

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left(\frac{M^2}{D} \right) + \frac{\partial}{\partial y} \left(\frac{MN}{D} \right) + gD \frac{\partial \eta}{\partial x} + \frac{g\alpha^2 M \sqrt{M^2 + N^2}}{D^{7/3}} = 0$$

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left(\frac{MN}{D} \right) + \frac{\partial}{\partial y} \left(\frac{N^2}{D} \right) + gD \frac{\partial \eta}{\partial y} + \frac{g\alpha^2 N \sqrt{M^2 + N^2}}{D^{7/3}} = 0$$

(1) Phương trình bảo toàn khối lượng, (2) Phương trình bảo toàn động lượng theo hướng x , (3) Phương trình bảo toàn động lượng theo hướng y

- M và N là các lưu lượng nước theo hướng x và y, được xác định với u, v tương ứng là vận tốc ngang theo hướng x, y

$$M = \int_{-h}^{\eta} u dz = u(h + n) = uD$$

$$N = \int_{-h}^{\eta} v dz = v(h + n) = vD$$

- Đại lượng $\frac{g\alpha^2 N \sqrt{M^2 + N^2}}{D^{7/3}}$ và $\frac{g\alpha^2 M \sqrt{M^2 + N^2}}{D^{7/3}}$ mô tả ảnh hưởng của đáy biển đến biên độ sóng.
- α biểu thị độ nhám Manning
- g biểu thị gia tốc trọng trường

❖ Công thức tổng quát cho phân bố Gaussian trong 2D

$$f(x, y) = A \cdot \exp \left(- \left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2} \right) \right)$$

- Mục đích: mô hình được sóng trong mặt phẳng 2D
- A là biên độ của sóng
- σ_x và σ_y là độ rộng phân bố theo 2 hướng x và y

❖ Sử dụng phương pháp sai phân hữu hạn để giải phương trình nước nông 2D

- Chia miền tính toán thành lưới điểm $m \times n$. Tại mỗi điểm (i, j) ta thực hiện tính toán $\eta(i, j)$, $M(i, j)$, $N(i, j)$ sau các khoảng thời gian Δt

- Tính $\eta(i, j)$

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0$$

* *Xấp xỉ đạo hàm theo không gian*

- Đạo hàm riêng của M theo x

$$\frac{\partial M}{\partial x}(i, j) \approx \frac{M[i, j + 1] - M[i, j - 1]}{2 \cdot dx}$$

Đây là công thức sai phân trung tâm cho đạo hàm bậc nhất, sử dụng các giá trị tại hai điểm lưới lân cận $j + 1$ và $j - 1$

- Đạo hàm riêng của N theo y

$$\frac{\partial N}{\partial y}(i, j) \approx \frac{N[i + 1, j] - N[i - 1, j]}{2 \cdot dy}$$

Tương tự, đây là công thức sai phân trung tâm trên lưới theo phương y .

- * Xấp xỉ đạo hàm theo thời gian

- Đạo hàm theo thời gian của η

$$\frac{\partial \eta}{\partial t}(i, j) \approx \frac{\eta^{n+1}[i, j] - \eta^n[i, j]}{\Delta t}$$

$$\text{Từ đó: } \eta^{n+1}[i, j] = \eta^n[i, j] - \Delta t \cdot \left(\frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} \right)$$

→ Đây là cách cập nhật giá trị của η dựa trên thông tin về sự phân kỳ của lưu lượng dòng chảy M, N tại thời điểm hiện tại

➤ Tính $M(i, j)$

Phương trình bảo toàn động lượng theo phương x

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left(\frac{M^2}{D} \right) + \frac{\partial}{\partial y} \left(\frac{MN}{D} \right) + gD \frac{\partial \eta}{\partial x} + \frac{g\alpha^2 M \sqrt{M^2 + N^2}}{D^{7/3}} = 0$$

Sử dụng công thức sai phân trung tâm:

- $$A = \frac{\partial}{\partial x} \left(\frac{M[i, j]^2}{D[i, j]} \right) \approx \frac{\frac{M[i, j+1]^2}{D[i, j+1]} - \frac{M[i, j-1]^2}{D[i, j-1]}}{2 \cdot dx}$$
- $$B = \frac{\partial}{\partial y} \left(\frac{M[i, j] \cdot N[i, j]}{D[i, j]} \right) \approx \frac{\frac{M[i+1, j] \cdot N[i+1, j]}{D[i+1, j]} - \frac{M[i-1, j] \cdot N[i-1, j]}{D[i-1, j]}}{2 \cdot dy}$$
- $$C = gD[i, j] \cdot \frac{\partial \eta[i, j]}{\partial x[i, j]} \approx gD[i, j] \cdot \frac{\eta[i, j+1] - \eta[i, j-1]}{2 \cdot dx}$$

- $X = \frac{g\alpha^2 M \sqrt{M^2 + N^2}}{D^{7/3}}$
- $\frac{\partial M[i,j]}{\partial t} \approx A + B + C + X$

Từ đó: $M^{n+1}[i,j] = M^n[i,j] - \Delta t \cdot (A + B + C + X)$

➤ Tính $N(i,j)$

Phương trình bảo toàn động lượng theo phương y

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left(\frac{MN}{D} \right) + \frac{\partial}{\partial y} \left(\frac{N^2}{D} \right) + gD \frac{\partial \eta}{\partial y} + \frac{g\alpha^2 N \sqrt{M^2 + N^2}}{D^{7/3}} = 0$$

Sử dụng công thức sai phân trung tâm:

- $A = \frac{\partial}{\partial y} \left(\frac{N[i,j]^2}{D[i,j]} \right) \approx \frac{\frac{N[i+1,j]^2}{D[i+1,j]} - \frac{N[i-1,j]^2}{D[i-1,j]}}{2 \cdot dy}$
- $B = \frac{\partial}{\partial x} \left(\frac{M[i,j] \cdot N[i,j]}{D[i,j]} \right) \approx \frac{\frac{M[i,j+1] \cdot N[i,j+1]}{D[i,j+1]} - \frac{M[i,j-1] \cdot N[i,j-1]}{D[i,j-1]}}{2 \cdot dx}$
- $C = gD[i,j] \cdot \frac{\partial \eta[i,j]}{\partial y[i,j]} \approx gD[i,j] \cdot \frac{\eta[i,j+1] - \eta[i,j-1]}{2 \cdot dy}$
- $X = \frac{g\alpha^2 N \sqrt{M^2 + N^2}}{D^{7/3}}$
- $\frac{\partial N[i,j]}{\partial t} \approx A + B + C + X$

Từ đó: $N^{n+1}[i,j] = N^n[i,j] - \Delta t \cdot (A + B + C + X)$



HUST

III. CHƯƠNG TRÌNH

❖ Giải thích thuật toán

- Tạo mô hình đại dương với L_x và L_y là chiều dài theo phương x và y , rời rạc mô hình với nx và ny điểm lưới theo phương x và y .
- Thời gian t được chia thành các khoảng dt ; dx và dy là khoảng cách của các điểm tương ứng trên lưới điểm theo x , y
- η là trường biên độ, M và N là các trường lưu lượng theo phương x và y , h là trường độ cao mực nước tĩnh
- Các hằng số gia tốc g và độ nhám α .
- Sử dụng ngôn ngữ lập trình Python

❖ Giải thích thuật toán

- Tạo 3 hàm `update_eta_2D`, `update_M_2D`, `update_N_2D` để cập nhật các trường tương ứng η , M , N với các dữ liệu vào cần thiết.
- Tạo hàm `Shallow_water_2D` với cơ chế hoạt động: sau mỗi khoảng thời gian dt , cập nhật các trường η , M , N , sau đó cập nhật $D = \eta + h$ là trường mô tả độ dày của cột sóng.
- Xác định các điều kiện biên
- Trong quá trình giải quyết bài toán nước nông, chúng ta hình dung chiều cao sóng và độ sâu bằng cách sử dụng hàm `imshow` và ghi kết quả dưới dạng hình ảnh TIFF (được mô tả trong hàm `Shallow_water_2D`). Các hình ảnh TIFF sau đó có thể được biên dịch thành phim tua nhanh thời gian.

- Nhập các thư viện cần thiết

```
1  # xuất thu vien
2  import numpy
3  from matplotlib import pyplot
4  from scipy.ndimage import gaussian_filter
5  import os
6  from numba import jit
```

- Cấu hình các tham số cho Matplotlib

```
8  # tham so matplotlib
9  pyplot.rcParams['font.family'] = 'serif'
10 pyplot.rcParams['font.size'] = 16
```

- Tạo thư mục đầu ra **image_out**

```
12 # image_out
13 output_dir = "C:/image_out"
14 os.makedirs(output_dir, exist_ok=True)
```

- Cập nhật trường η

```
16 # truong eta
17 @jit(nopython=True)
18 def update_eta_2D(eta, M, N, dx, dy, dt, nx, ny):
19     for i in range(1, nx - 1):
20         for j in range(1, ny - 1):
21             dMdx = (M[j, i + 1] - M[j, i - 1]) / (2. * dx)
22             dNdy = (N[j + 1, i] - N[j - 1, i]) / (2. * dy)
23             eta[j, i] = eta[j, i] - dt * (dMdx + dNdy)
24     eta[0, :] = eta[1, :]
25     eta[-1, :] = eta[-2, :]
26     eta[:, 0] = eta[:, 1]
27     eta[:, -1] = eta[:, -2]
28     return eta
```

- Cập nhật trường M

```
30 # truong m
31 @jit(nopython=True)
32 def update_M_2D(eta, M, N, D, g, h, alpha, dx, dy, dt, nx, ny):
33     arg1 = M ** 2 / D
34     arg2 = M * N / D
35     fric = g * alpha ** 2 * M * numpy.sqrt(M ** 2 + N ** 2) / D ** (7. / 3.)
36     for i in range(1, nx - 1):
37         for j in range(1, ny - 1):
38             darg1dx = (arg1[j, i + 1] - arg1[j, i - 1]) / (2. * dx)
39             darg2dy = (arg2[j + 1, i] - arg2[j - 1, i]) / (2. * dy)
40             detadx = (eta[j, i + 1] - eta[j, i - 1]) / (2. * dx)
41             M[j, i] = M[j, i] - dt * (darg1dx + darg2dy + g * D[j, i] * detadx + fric[j, i])
42     return M
```

- Cập nhật trường N

```
44 # truong n
45 @jit(nopython=True)
46 def update_N_2D(eta, M, N, D, g, h, alpha, dx, dy, dt, nx, ny):
47     arg1 = M * N / D
48     arg2 = N ** 2 / D
49     fric = g * alpha ** 2 * N * numpy.sqrt(M ** 2 + N ** 2) / D ** (7. / 3.)
50     for i in range(1, nx - 1):
51         for j in range(1, ny - 1):
52             darg1dx = (arg1[j, i + 1] - arg1[j, i - 1]) / (2. * dx)
53             darg2dy = (arg2[j + 1, i] - arg2[j - 1, i]) / (2. * dy)
54             detady = (eta[j + 1, i] - eta[j - 1, i]) / (2. * dy)
55             N[j, i] = N[j, i] - dt * (darg1dx + darg2dy + g * D[j, i] * detady + fric[j, i])
56     return N
```

❖ Hàm giải phương trình nước nông

- Khởi tạo các giá trị và biến cần thiết để tính toán

```
59  def Shallow_water_2D(eta0, M0, N0, h, g, alpha, nt, dx, dy, dt, X, Y):
60      eta = eta0.copy()
61      M = M0.copy()
62      N = N0.copy()
63      D = eta + h
64      ny, nx = eta.shape
65      x = numpy.linspace(0, nx * dx, num=nx)
66      y = numpy.linspace(0, ny * dy, num=ny)
--
```

- Hiển thị trực quan trường sóng η và địa hình đáy biển h trên cùng một biểu đồ.

```
68 fig, ax = pyplot.subplots(figsize=(10, 6))
69 cmap = 'seismic'
70 extent = [numpy.min(x), numpy.max(x), numpy.min(y), numpy.max(y)]
71 topo = ax.imshow(numpy.flipud(-h), cmap=pyplot.cm.gray, interpolation='nearest', extent=extent)
72 im = ax.imshow(numpy.flipud(eta), extent=extent, interpolation='spline36', cmap=cmap, alpha=.75, vmin=-0.4, vmax=0.4)
73 ax.set_xlabel('x [m]')
74 ax.set_ylabel('y [m]')
75 pyplot.colorbar(im, ax=ax, label=r'$\eta$ [m]')
76 pyplot.colorbar(topo, ax=ax, label=r'$-h$ [m]')
77 ax.invert_yaxis()
```

- Khởi tạo tham số lưu ảnh

```
79 nsnap = 50
80 snap_count = 0
```

- Với các thời điểm, cập nhật các trường η , M , N , D và lưu lại mô hình sau một số bước thời gian nhất định

```
82  ✓ for n in range(nt):
83      eta = update_eta_2D(eta, M, N, dx, dy, dt, nx, ny)
84      M = update_M_2D(eta, M, N, D, g, h, alpha, dx, dy, dt, nx, ny)
85      N = update_N_2D(eta, M, N, D, g, h, alpha, dx, dy, dt, nx, ny)
86      D = eta + h
87
88  ✓ if (n % nsnap) == 0:
89      im.set_data(numpy.flipud(eta))
90      fig.canvas.draw()
91      fig.canvas.flush_events()
92      name_snap = os.path.join(output_dir, f"Shallow_water_2D_{snap_count + 1000}.tiff")
93      pyplot.savefig(name_snap, format='tiff', bbox_inches='tight', dpi=125)
94      snap_count += 1
95
96  return eta, M, N
```



HUST

IV. ỨNG DỤNG

- ❖ Sau khi đã có hàm **shallow_water_2d**, chúng ta có thể mô phỏng được sự lan truyền của sóng thần trên đại dương bằng cách truyền các tham số đầu vào.
- ❖ Ví dụ:
 - Giả sử có 1 mô hình đại dương $L_x = 100\text{ m}$ và $L_y = 100\text{ m}$ theo các hướng x, y , với số lượng điểm lưới $n_x = 401$ và $n_y = 401$, độ sâu không đổi $h = 50\text{ m}$, gia tốc $g = 9.81$, độ nhám $\alpha = 0.025$
 - Trường eta là một Gaussian ở trung tâm mô hình, với biên độ 0.5 m và bán kính là 10 m .

- Trường thông lượng xả M, N được định nghĩa như sau:

$$\begin{aligned}M_0(x, y) &= 100\eta_0(x, y) \\ N_0(x, y) &= 0\end{aligned}$$

- Điều kiện biên cho các lưu lượng M, N ở mọi biên:

$$\begin{aligned}M(0, y) &= M(L_x, y) = M(x, 0) = M(x, L_y) \\ N(0, y) &= N(L_x, y) = N(x, 0) = N(x, L_y)\end{aligned}$$

- Điều kiện biên cho η

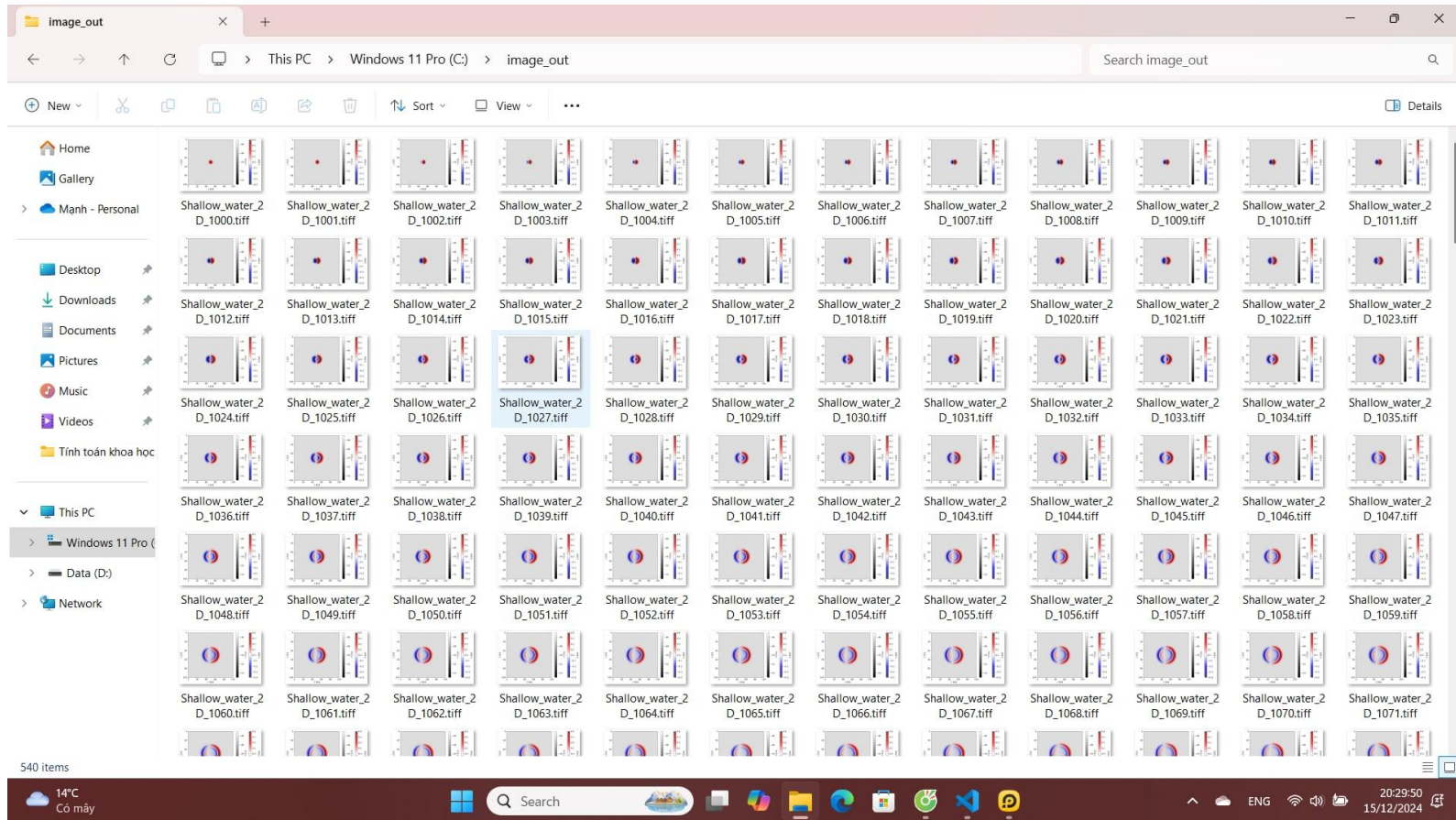
$$\frac{\partial \eta}{\partial x}(0, y) = \frac{\partial \eta}{\partial x}(L_x, y) = \frac{\partial \eta}{\partial y}(x, 0) = \frac{\partial \eta}{\partial y}(x, L_y) = 0$$

**Các điều kiện ranh giới giả định dẫn đến phản xạ ranh giới đáng kể có thể không thực tế đối với một vấn đề nhất định. Tuy nhiên, để giữ cho vấn đề đơn giản, chúng ta sẽ không cố giảm thiểu phản xạ ranh giới.*

- Tiếp theo, cùng với các hàm Shallow_water_2D, update_eta_2D, update_M_2D, update_N_2D, chạy đoạn code sau:

```
121 Lx = 100.0
122 Ly = 100.0
123 nx = 401
124 ny = 401
125 dx = Lx / (nx - 1)
126 dy = Ly / (ny - 1)
127 x = numpy.linspace(0.0, Lx, num=nx)
128 y = numpy.linspace(0.0, Ly, num=ny)
129 X, Y = numpy.meshgrid(x, y)
130 h = 50 * numpy.ones_like(X)
131 eta0 = 0.5 * numpy.exp(-((X - 50)**2 / 10) - ((Y - 50)**2 / 10))
132 M0 = 100. * eta0
133 N0 = 0. * M0
134 g = 9.81
135 alpha = 0.025
136 Tmax = 6.0
137 dt = 1 / 4500.0
138 nt = int(Tmax / dt)
139 eta, M, N = Shallow_water_2D(eta0, M0, N0, h, g, alpha, nt, dx, dy, dt, X, Y)
```

- Kết quả thu được là một thư mục gồm các hình ảnh TIFF





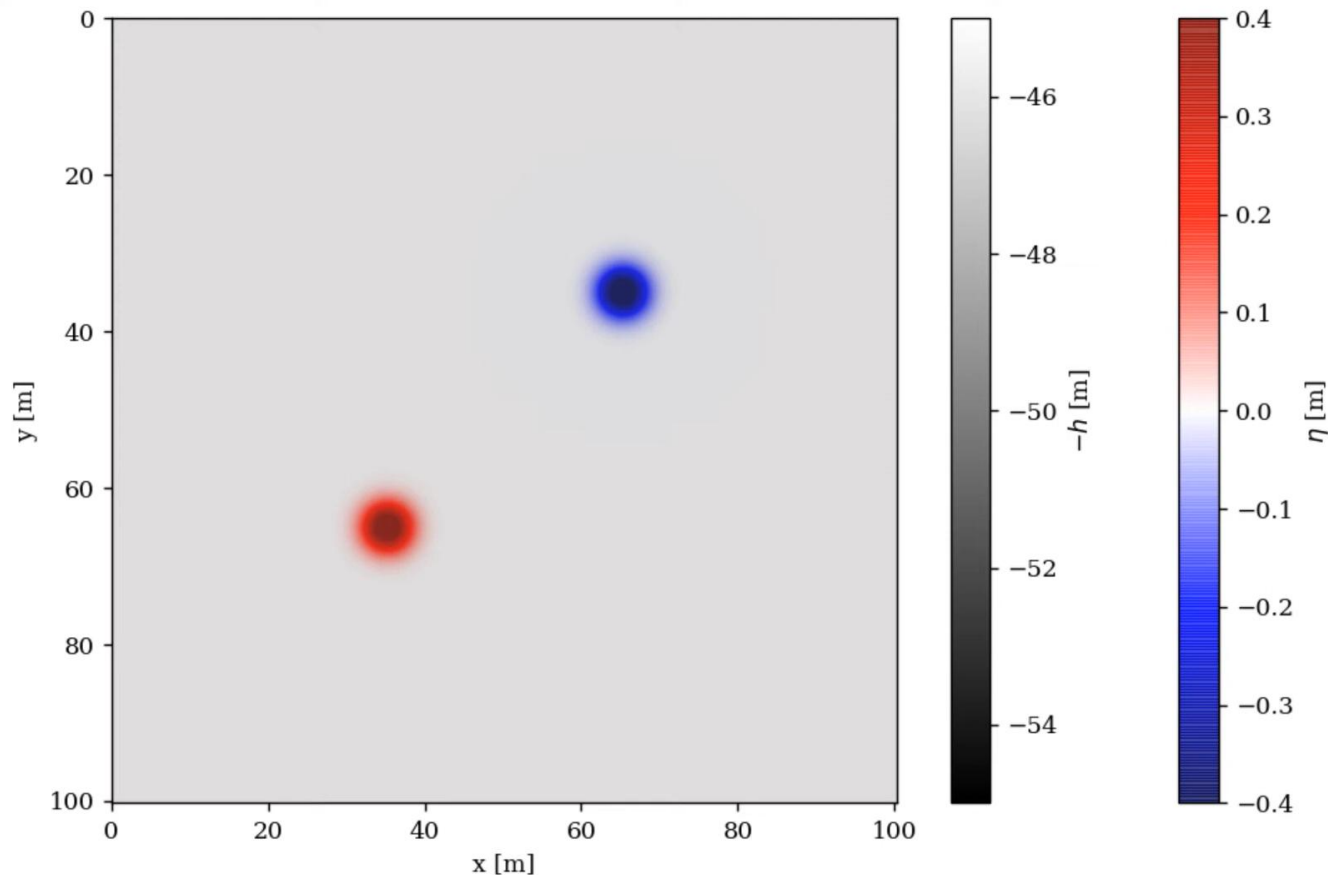
HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

Ứng dụng

- Tương tự, mô phỏng sự lan truyền và tương tác của hai cơn sóng thần với 2 nguồn tại $(x_1, y_1) = (35 \text{ m}, 35 \text{ m})$, $(x_2, y_2) = (60 \text{ m}, 60 \text{ m})$, các tham số khác giữ nguyên





HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

- Sóng thần trong đại dương có địa hình thay đổi ngẫu nhiên. Chúng ta thêm một số nhiễu loạn ngẫu nhiên vào mô hình đo độ sâu không đổi h_0 bằng cách sử dụng hàm `random.rand` từ thư viện NumPy. Để làm mịn nhiễu loạn ngẫu nhiên, chúng ta áp dụng hàm `gaussian_filter` từ SciPy thư viện:



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

KẾT LUẬN

- Như vậy, chúng ta có thể hình dung được một cơn sóng thần lan truyền trong đại dương với một số điều kiện đầu vào khác nhau.
- Các mô hình trên chỉ là lý tưởng hóa, thực tế có thể xảy ra sự sai khác lớn.

A large, stylized graphic of the HUST logo, composed of concentric circles of dots in a lighter shade of red, set against a solid red background.

HUST

THANK YOU !