

## KHOA HỌC - CÔNG NGHỆ

# NGHIÊN CỨU MÔ PHỎNG 3D QUÁ TRÌNH CHUYỂN ĐỘNG CỦA MÀNG DẦU TRÊN BIỂN DỰA TRÊN THUẬT TOÁN TẠO LƯỚI ĐA GIÁC TỪ MỘT BỀ MẶT ĐẲNG THỂ

## RESEARCH ON 3D SIMULATION OF THE OIL FILM TRANSPORTATION AT SEA BASED ON POLYGON MESH GENERATION ALGORITHM BY AN ISO-SURFACE

ĐỖ VĂN CƯỜNG\*<sup>1</sup>, TRẦN GIA NINH<sup>1</sup>, ĐỖ TRUNG KIÊN<sup>2</sup>

<sup>1</sup>Khoa Hàng hải, Trường Đại học Hàng hải Việt Nam

<sup>2</sup>Phòng Quan hệ Quốc tế, Trường Đại học Hàng hải Việt Nam

\*Email liên hệ: dovancuong@vamaru.edu.vn

### Tóm tắt

Có hai phương pháp thực hiện kết xuất hình dạng của chất lỏng: Phương pháp hình học và phương pháp vật lý. Phương pháp hình học cho kết quả tính toán nhanh và yêu cầu phần cứng máy tính không cao. Phương pháp vật lý có thể mô phỏng một cách chính xác và tỉ mỉ chuyển động của chất lỏng nhưng lại đòi hỏi cấu hình phần cứng khá cao và khối lượng tính toán lớn. Trong bài báo này, nhóm tác giả đã áp dụng thuật toán hình học tạo lưới đa giác từ một giá trị đẳng thể để tái tạo và kết xuất chuyển động của màng dầu trên biển. Đây là nghiên cứu đầu tiên ở trong nước và trên thế giới áp dụng thuật toán Marching Square vào mô phỏng màng dầu trên biển. Ưu điểm của phương pháp này là có thể mô phỏng một thể tích lớn dầu tràn trên biển với tốc độ nhanh, khối lượng tính toán ít mà vẫn đảm bảo yếu tố thời gian thực (real-time).

**Từ khóa:** Chất lỏng, lưới đa giác, tràn dầu, thời gian thực.

### Abstract

There are two methods to render fluid: The geometric method and the physical method. The advantage of the geometric method is fast calculation results and not very high of hardware requirements. On the other hand, the physical method can be proposed an accurately and meticulously method to simulate the movement of oil film at sea but require of higher computer hardware and computation at the same oil spill volume. In this paper, the authors have applied a geometric algorithm to generate a polygon mesh from an iso-value to render and simulate the

motion of the oil film at sea. This is the first study to apply Marching Square algorithm to simulate oil film in Vietnam and in the world. The algorithm can be able to simulate a large volume of oil spilled at a fast speed, reduce the computation but still satisfy the real-time factors.

**Keywords:** Fluid, polygon mesh, oil spill, real-time.

### 1. Giới thiệu

Dầu bản chất cũng là một loại chất lưu được đặc trưng bởi độ nhớt, nhiệt độ, sức căng bề mặt, áp suất,... Trong lĩnh vực đồ họa máy tính, việc mô phỏng chất lưu có rất nhiều ứng dụng. Một chương trình mô phỏng chất lưu tốt sẽ có tầm quan trọng trong rất nhiều lĩnh vực như mô phỏng chính xác chuyển động của đám khói, lửa, nước và dầu để ứng dụng trong các ngành giải trí, phim ảnh, đào tạo, huấn luyện và trong các ngành khoa học, kỹ thuật khác. Động lực học chất lưu đóng vai trò là nền tảng tiêu chuẩn của các phương trình toán học để mô phỏng chất lưu như ngày nay.

Ngày nay, với sự phát triển của khoa học công nghệ trong đó có ngành công nghệ thông tin, việc mô phỏng chất lỏng đã không còn là vấn đề quá khó khăn và phức tạp. Tuy nhiên, ở một góc độ nào đó thì quá trình mô phỏng vẫn chỉ là sự tiệm cận với môi trường thật ngoài tự nhiên mà thôi. Đó là do sự hạn chế của phần cứng máy tính mà cụ thể là CPU (Central Processing Unit) và GPU (Graphics Processing Unit) cũng không thể nào đáp ứng được hết tất cả các yêu cầu của hàng tỷ phép tính trong mô phỏng động lực học chất lưu.

Trong cách tiếp cận việc kết xuất chất lỏng, có hai phương pháp được sử dụng là phương pháp hình học và phương pháp vật lý. Phương pháp mô phỏng hình

học dựa trên nguyên lý xây dựng lưới đa giác (tam giác hoặc tứ giác) để kết nối các phần tử rời rạc lại với nhau, sau đó sử dụng sức mạnh của GPU để tính toán màu sắc, độ tương phản, ánh sáng nhằm kết xuất đưa ra hiển thị trên màn hình máy tính. M. Muller và cộng sự [1] đã sử dụng phương pháp lưới không gian màn hình (Screen Space Mesh) để mô phỏng nước. Ông sử dụng thuật toán Marching Square tạo lưới 2D, sau đó làm mịn bề mặt đa giác và kết xuất hình ảnh ra màn hình. S. Almeida và cộng sự [2] đã mở rộng lưới không gian màn hình cho phương pháp SPH (Smoothed Particle Hydrodynamics) dựa trên ngôn ngữ Shader Labs để kết xuất ra màn hình. Phương pháp mô phỏng vật lý không xây dựng lưới hình học mà dựa trên phương trình Navier-Stokes (N-S) để mô phỏng trực tiếp quá trình chuyển động của chất lỏng, phương trình này được đặt tên theo Claude-Louis Navier và George Gabriel Stokes. Đây là phương trình dùng để mô phỏng chuyển động của các dòng chảy chất lưu không nén được trong tự nhiên bằng cách xây dựng một trường vector, trong đó mỗi một chất điểm được đặc trưng bởi một đại lượng vector có hướng biểu thị cho vận tốc, nhiệt độ, áp suất của chất lưu. Từ sơ đồ hướng chuyển động của vector, người ta sẽ xây dựng được luồng dịch chuyển của chất lưu. J. Stam [3] đã sử dụng mô hình vật lý dựa trên phương trình N-S để mô phỏng chuyển động của chất lưu ổn định, tuy nhiên kết quả mô phỏng cho thấy sự chuyển động của dòng chảy giảm đi khá nhanh so với thực tế. K. Kakuda và cộng sự [4] sử dụng công nghệ GPGPU (General Purpose Computing on Graphics Processing Units) bằng CUDA (Compute Unified Device Architecture) để giải quyết phương trình N-S mô phỏng dòng chảy của chất lỏng, điều này cũng tăng đáng kể hiệu suất của quá trình tính toán.

Trong bài báo này, nhóm tác giả sẽ xây dựng mô hình mô phỏng 3D quá trình chuyển động của màng dầu trên biển dựa trên thuật toán Marching Square trong không gian 2 chiều bằng phần mềm Unity. Do chương trình được áp dụng trong hệ thống mô phỏng ứng phó sự cố tràn dầu trên biển nên có đặc điểm là cần cho kết quả nhanh, ít dữ liệu đầu vào, chính vì vậy chỉ khảo sát 2 quá trình lan rộng và trôi nổi của màng dầu là hoàn toàn phù hợp. Phương pháp này cho ra kết quả nhanh, không cần nhiều dữ liệu đầu vào và có thể mô phỏng diện tích lớn của màng dầu trên biển mà vẫn đảm bảo yếu tố thời gian thực. Ngoài ra nó cũng biểu diễn được quá trình phân tách màng dầu thành những phần nhỏ hơn một cách dễ dàng, điều mà các nghiên cứu trước đây chưa xây dựng được.

## 2. Mô hình thuật toán biểu diễn sự lan truyền cơ học của màng dầu trên biển

### 2.1. Mô hình thuật toán biểu diễn quá trình lan rộng của màng dầu

Từ những năm 60 của thế kỷ 20, Fay [5] đã đưa ra các tính toán của mình cho hình dạng vệt dầu loang trên biển, ông coi nó chỉ đơn thuần là một hình tròn có bán kính  $r$  mở rộng về diện tích. Vào năm 1984, trong một nghiên cứu thực nghiệm tràn dầu tại vùng biển Ả-Rập, Lehr [6] đã chỉ ra rằng kết quả nghiên cứu tràn dầu của Fay là chưa chính xác và khó thực hiện trong thực tế, vì không tính đến ảnh hưởng của môi trường tới sự lan tràn dầu. Bằng các thử nghiệm thực tế, Lehr đã đưa ra được công thức toán học tính toán sự giãn nở diện tích màng dầu trên biển có tính đến ảnh hưởng của gió và dòng chảy, khi đó màng dầu có dạng gần giống hình Elip với bán trục lớn và bán trục nhỏ. Tuy nhiên, diện tích của màng dầu chỉ đạt đến một giới hạn nhất định thì sẽ không tăng lên mà chuyển sang các giai đoạn tiếp theo như phân tán, kết tủa và biến đổi vật lý khác. Lý thuyết của Lehr được minh họa qua công thức (1) và (2):

$$A = 2270 \left[ \frac{\Delta\rho}{\rho_0} \right]^{\frac{2}{3}} v^{\frac{2}{3}} t^{\frac{1}{2}} + 40 \left[ \frac{\Delta\rho}{\rho_0} \right]^{\frac{1}{3}} v^{\frac{1}{3}} W^{\frac{4}{3}} t \quad (1)$$

$$A_{\max} = 10^5 v^{\frac{3}{4}} \quad (2)$$

Với,  $A$  là diện tích màng dầu ( $m^2$ );  $A_{\max}$  là diện tích lớn nhất màng dầu ( $m^2$ );  $V$  là thể tích dầu bị tràn ( $barrel$ );  $W$  là tốc độ gió ( $knot$ );  $t$  là thời gian ( $phút$ );  $\rho_0$  là tỷ trọng dầu;  $\Delta\rho$  là biến thiên tỷ trọng nước biển-dầu;  $Q$  là bán trục nhỏ;  $R$  là bán trục lớn.

### 2.2. Mô hình thuật toán biểu diễn quá trình trôi dạt của màng dầu

Để cụ thể hóa các khẳng định của Lehr, M. Fingas [7] đã đưa ra được công thức thực nghiệm thể hiện sự phụ thuộc của hướng dịch chuyển màng dầu trên biển vào gió và dòng chảy. Sự trôi dạt của màng dầu được tính toán dựa trên tính toán chuyển động của từng hạt dầu riêng biệt. Tốc độ di chuyển của hạt dầu thứ  $i$  bất kỳ được xác định bởi đạo hàm của trục  $X$  theo thời gian  $t$ :

$$\frac{dX_i}{dt} = V_{drift}(x_i, y_i, t) + V_{diff}(x_i, y_i, t) \quad (3)$$

Với,  $X_i(x_i, y_i)$  là vị trí của hạt thứ  $i$ ;  $V_{diff}$  là tốc độ phân tán;  $t$  là thời gian ( $s$ ).

$$V_{drift}(x_i, y_i) = a_w V_w(x_i, y_i) + a_c V_c(x_i, y_i) \quad (4)$$

Với,  $V_w$  là tốc độ gió tại 10m trên mặt nước biển;  $V_c$  là tốc độ dòng chảy;  $a_w = 0,03$  là hệ số gió;  $a_c = 1,0$  là hệ số dòng chảy:

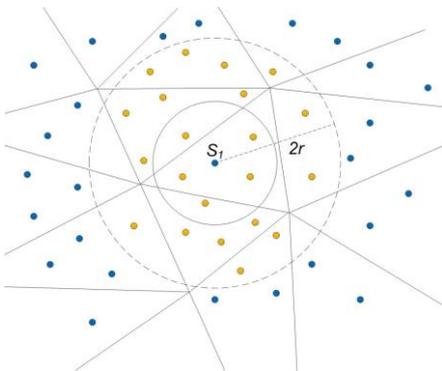
$$V_{diff}(x_i, y_i) = V_{drift} R_n e^{i\alpha_n} \quad (5)$$

Với,  $R_n [-1, 1]$  là hệ số;  $\alpha_n [0, \pi]$  là góc lệch;  $V_{drift}$  là vận tốc đạt.

### 3. Ứng dụng thuật toán đĩa Poisson biểu diễn tập hữu hạn các hạt dầu trên biển

Thuật toán phân chia điểm Poisson được nhắc đến lần đầu tiên vào năm 1960 để thực hiện phân bố cây trong rừng. Trong đồ họa máy tính, thuật toán này dùng để kết xuất hình ảnh, tạo hình và xử lý hình học cho đối tượng. Nguyên lý hoạt động của thuật toán là vẽ đối tượng trong không gian  $n$  chiều dựa trên việc tạo các tập điểm trên bề mặt đối tượng sao cho chúng cách nhau một khoảng cách đã được giới hạn trước. Ưu điểm của thuật toán là có thể phân bố các điểm trong không gian mà không bị chồng lấn hoặc quá thưa thớt. So với phân bố đều và phân bố ngẫu nhiên, thuật toán đĩa Poisson có thể điều chỉnh được mật độ xuất hiện các điểm trên bề mặt đối tượng tại những chỗ cần thể hiện chi tiết, do đó hình ảnh của đối tượng tại những vị trí này sẽ có độ phân giải sắc nét hơn. Ví dụ như trong vẽ ảnh, tại những vị trí như mắt, mũi, tóc, mũ của đối tượng cần được vẽ rõ nét thì mật độ điểm tại đó sẽ được tăng lên tương ứng.

Trong một nghiên cứu trước [8], nhóm tác giả đã áp dụng thuật toán phân chia đĩa Poisson được lập trình song song đa luồng chạy trên GPU để tăng hiệu suất tính toán và tăng số lượng hạt dầu được tạo ra. Cách thức thực hiện như sau: Đầu tiên thuật toán tạo ra một tập điểm trên một bề mặt, sau đó mỗi một luồng sẽ tự động nhặt một điểm trên tập vừa tạo ra để kiểm tra điều kiện khoảng cách  $d \geq 2r$  sẽ được lưu lại và



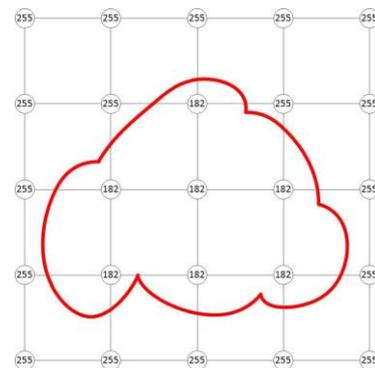
Hình 1. Phân chia điểm đa luồng trên GPU

hiển thị trên màn hình. Công việc được thực hiện đồng thời trên nhiều luồng của GPU để tăng đáng kể hiệu suất. Hình 1 thể hiện nguyên lý tính toán của thuật toán phân chia đĩa Poisson.

### 4. Thuật toán tạo lưới đa giác từ một bề mặt đẳng thế để mô phỏng màng dầu

#### 4.1. Thuật toán Marching Cubes và Marching Square

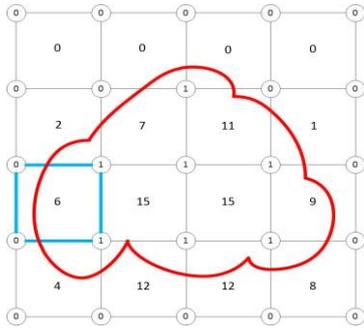
Thuật toán lấy mẫu đĩa Poisson đã tạo ra một bước đột phá trong việc mô tả các đặc điểm của sự cố tràn dầu trên biển, do đó cung cấp khả năng dễ dàng hiển thị chuyển động của nó thông qua việc áp dụng các công thức toán học cho từng hạt dầu, tuy nhiên hình dạng dầu vẫn rời rạc hàng nghìn hạt chưa hoàn toàn thỏa mãn mô phỏng chất lỏng. Trong bài báo này, nhóm tác giả sử dụng thuật toán Marching Cubes để biểu diễn sự liên tục của màng dầu. Thuật toán đã được áp dụng cho nhiều lĩnh vực, bao gồm trò chơi điện tử, hình ảnh trực quan y tế như hình ảnh quét máy tính (CT), hình ảnh cộng hưởng từ (MRI) và biểu diễn các mô hình 3D phi đa giác. Thuật toán Marching Cubes sử dụng một bề mặt đẳng thế để xác định đỉnh nào nằm bên trong hoặc bên ngoài mặt phẳng đó, sau đó một bề mặt sẽ được tạo ra từ các điểm cắt, được phát triển bởi Lorensen và Cline và ban đầu được trình bày tại hội nghị SIGGRAPH năm 1987 [9]. Kỹ thuật này cho phép chuyển đổi mô hình 3D phi đa giác (chẳng hạn như mô hình được biểu thị bằng *voxels*) thành mô hình đa giác. Để đơn giản ta đưa thuật toán về không gian 2 chiều gọi là Marching Square. Nguyên lý của thuật toán Marching Square như sau:



Hình 2. Tính toán chỉ số của vertex và cell

- Bước 1: Chia miền cần vẽ thành các *cell* và gán cho mỗi đỉnh (*vertex*) của *cell* một chỉ số (*index*), thuật toán sẽ tiến hành so sánh các chỉ số của *vertex* và giá trị *isovalue* mặc định, nếu chỉ số của *vertex* lớn hơn *isovalue* thì chuyển chỉ số của *vertex* sang hệ nhị phân là 0, ngược lại chuyển chỉ số của *vertex* là 1, trong

Hình 2 và Hình 3, lựa chọn giá trị  $isovalue = 200$ .



Hình 3. Tính toán chỉ số của vertex và cell

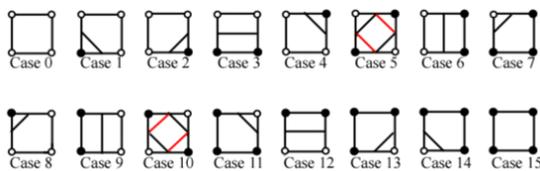
- Bước 2: Tại mỗi *cell* của miền tính toán, đi theo chiều kim đồng hồ để xác định mã nhị phân của *cell*, sau đó sử dụng toán tử thao tác *bit OR* (*bitwise OR*) và phép dịch trái *bit* để tìm ra chỉ số của *cell* dưới dạng hệ thập phân. Tại Hình 3, ví dụ lựa chọn ô được tô đậm xanh để tính, giá trị *bit* nhị phân là 0110 tương đương với số 6 trong hệ thập phân. Tương tự chúng ta cũng sẽ tính được chỉ số hệ thập phân với các *cell* còn lại.

- Bước 3: Với mỗi *cell* có dạng hình vuông 4 cạnh, mỗi cạnh sẽ có 2 trường hợp đường bao (*contour line*) cắt qua, như vậy sẽ có  $2^4 = 16$  trường hợp có thể xảy ra trong thuật toán *Marching Square* như trong Hình 4.

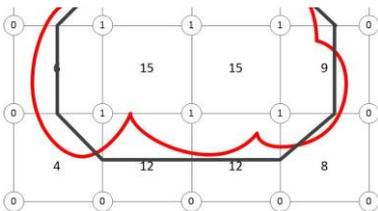
- Bước 4: Từ chỉ số của *cell* tính được ở bước 2 tra vào bảng 16 trường hợp ở bước 3 để vẽ các cạnh của đa giác, đường tô đậm màu đen trong Hình 5.

- Bước 5: Áp dụng phép nội suy tuyến tính giữa các giá trị dữ liệu trường ban đầu để tìm vị trí chính xác của đường bao dọc theo các cạnh của *cell*, đường màu đỏ trong Hình 5.

Như vậy, trải qua 5 bước ở trên, chúng ta có thể vẽ ra được bề mặt của đa giác từ một trường vô hướng 2 chiều.



Hình 4. Các trường hợp đường bao cắt các cell

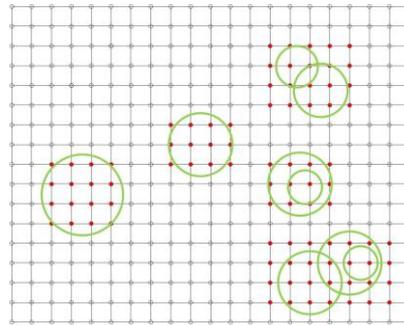


Hình 5. Vẽ sơ bộ hình dạng của đa giác

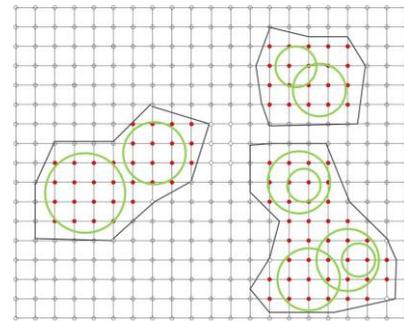
### 4.2. Ứng dụng thuật toán *Marching Cubes* để mô phỏng chuyển động màng dầu trên biển

Do hạt dầu ở nghiên cứu trước [8] được coi có dạng hình cầu, đầu tiên thuật toán sẽ kiểm tra xem những đỉnh nào của *cell* nằm bên trong hình tròn. Trong Hình 6, những đỉnh có màu đỏ là điểm được hiển thị trên màn hình, cũng chính là vùng tính toán của thuật toán. Phương trình thuật toán các điểm có tọa độ  $(x_0, y_0)$  nằm trong hình tròn bán kính  $r$  thỏa mãn:

$$(x - x_0)^2 + (y - y_0)^2 \leq r^2 \quad (6)$$

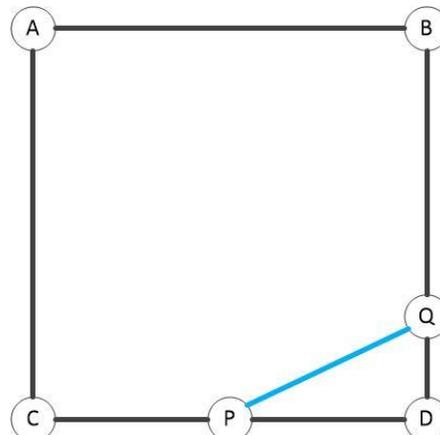


Hình 6. Kiểm tra các đỉnh nằm trong hình tròn



Hình 7. Vẽ sơ bộ hình dạng của màng dầu

Sau khi xác định được vùng tính toán là các đỉnh màu đỏ, chúng ta cũng áp dụng lần lượt các bước như



Hình 8. Nội suy tuyến tính giao điểm của đường bao

ở Mục 4.1 để vẽ các khối hạt dầu chuyển động uyển chuyển và va chạm vào nhau tạo thành các hình dạng đặc trưng của màng dầu như trong Hình 7.

Để sự chuyển động của màng dầu không bị gấp khúc, chúng ta áp dụng phương pháp nội suy tuyến tính để làm mịn đường bao và giảm hiệu ứng răng cưa. Thuật toán Marching Square tạo ra một xấp xỉ cho một đường bao của trường vô hướng 2 chiều. Nói cách khác, nếu chúng ta có hàm 2D, hàm này sẽ tìm giá trị gần đúng của một đường trong đó tất cả các điểm trên đường có cùng giá trị hàm thỏa mãn điều kiện  $f(x,y)=1$ . Xét trường hợp thứ 2 trong 16 trường hợp của thuật toán như Hình 8, đặt ký hiệu cho các đỉnh của cell là A, B, C, D, đường bao sẽ cắt cell tại P và Q.

Bài toán đặt ra là tìm tọa độ của  $Q(x,y)$ , đối với điểm  $P(x,y)$  thì làm tương tự. Ta đã biết:

$$Q_x = B_x \quad (7)$$

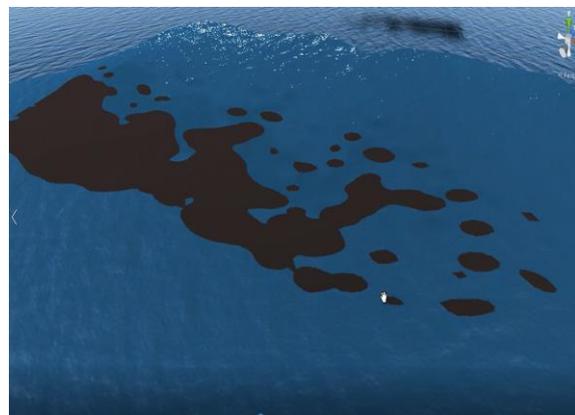
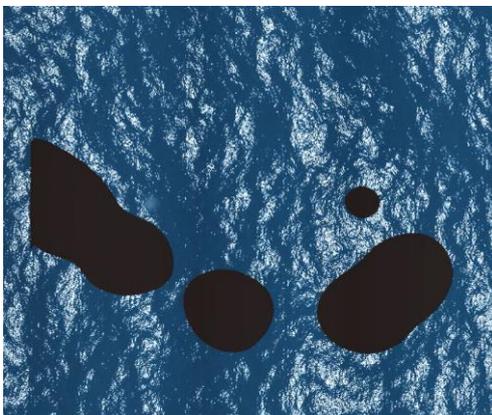
Mặc dù  $f(x, y)$  không phải là tuyến tính, phép nội suy tuyến tính vẫn cho kết quả tốt để làm mịn các đường bao, do đó có thể tính gần đúng như sau:

$$\frac{Q_y - B_y}{D_y - B_y} \approx \frac{f(Q_x, Q_y) - f(B_x, B_y)}{f(D_x, D_y) - f(B_x, B_y)} \quad (8)$$

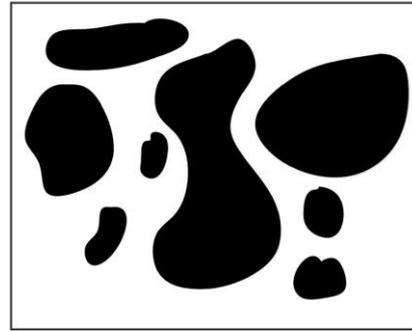
Vì chúng ta đang tìm  $Q$  để nó nằm trên đường bao trong đó  $f(x,y) = 1$ , nên chúng ta muốn  $f(Q_x, Q_y) \approx 1$  thì công thức (8) được viết lại như sau:

$$Q_y = B_y + (D_y - B_y) \left( \frac{1 - f(B_x, B_y)}{f(D_x, D_y) - f(B_x, B_y)} \right) \quad (9)$$

Các giá trị ở vế phải của (9) đều đã biết, vì vậy dễ dàng tính được giá trị của  $Q_y$ . Tương tự như vậy chúng ta cũng có thể thiết lập công thức để tính được tọa độ của  $P(x,y)$ . Sau khi nội suy tuyến tính và tăng độ phân giải thì hình dạng của màng dầu được thể hiện như mô tả trong Hình 9.



Hình 10. Mô phỏng vệt dầu tại các thời điểm khác nhau



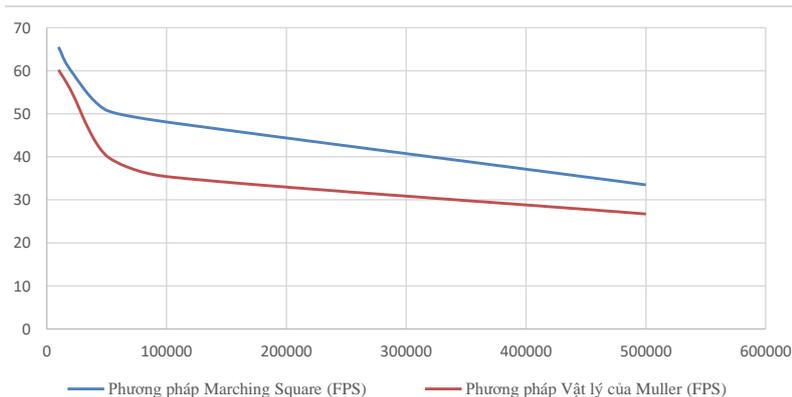
Hình 9. Hình dạng vệt dầu sau khi nội suy tuyến tính

## 5. Thực hiện mô phỏng vệt dầu loang trên biển

Để thực hiện mô phỏng vệt dầu loang với hơn 500 nghìn hạt dầu, nhóm tác giả đã sử dụng Card đồ họa NVIDIA GTX 1050Ti được gắn trên Mainboard Intel Core(TM) i7-7700HQ xung nhịp 2.80 GHz. Card GTX 1050Ti được gắn 768 nhân CUDA - kiến trúc Pascal, tên mã GP107, bộ nhớ dung lượng 4Gb GDDR5, độ rộng băng thông 128 bit. Đây là một loại Card màn hình thuộc phân khúc tầm trung nhưng vẫn đáp ứng được render các hình ảnh của vệt dầu loang trên biển. Hình 10 thể hiện kết quả mô phỏng sự chuyển động của màng dầu trên biển bằng phần mềm Unity với các thông số đầu vào của dầu như sau:

- Lượng dầu tràn:  $1500m^3$ ;
- Khối lượng riêng của dầu:  $\rho_0 = 890 kg/m^3$ ;
- Khối lượng riêng nước biển:  $\rho_w = 1023 kg/m^3$
- Vận tốc gió ở độ cao 10 m so với mặt nước biển:  $W_{10} = 0.5 m/s$ ;

Hình bên trái là khi mới bắt đầu xảy ra sự cố tràn dầu, vệt dầu có diện tích bao phủ nhỏ và chủ yếu có dạng Elip. Hình bên phải là khi vệt dầu đã loang và phát triển về mặt thể tích, do đó diện tích bao phủ sẽ lớn hơn. Các hình ảnh được chụp ở các góc quay khác



Hình 11. So sánh chỉ số FPS (Frame per Second)

nhau. Chương trình mô phỏng đã thể hiện được quá trình phân tách của màng dầu thành những phần nhỏ hơn và sự chuyển động nhịp nhàng giữa màng dầu với sóng biển như ngoài thực tế.

Ngoài ra, để thể hiện tính ưu việt của chương trình, nhóm tác giả đã so sánh chỉ số FPS (Frame per Second) với phương pháp mô phỏng vật lý của Muller, kết quả cho thấy phương pháp Marching Square luôn cho FPS cao hơn với cùng số lượng hạt dầu được mô phỏng (xem Hình 11).

## 6. Kết luận

Mô phỏng chất lỏng nói chung và màng dầu nói riêng có tầm quan trọng rất lớn trong việc đào tạo, huấn luyện công tác ứng phó tràn dầu trên biển, góp phần giúp tiết kiệm chi phí và nguồn lực con người. Tuy nhiên, với việc vừa mô phỏng bề mặt biển và màng dầu sẽ gây ra khối lượng tính toán và kết xuất đồ họa là vô cùng lớn. Phương pháp mô phỏng hình học dựa trên thuật toán tạo lưới đa giác từ một bề mặt đẳng thế cho kết quả nhanh và có thể mô phỏng một diện tích dầu lớn. Đồng thời chỉ số FPS cũng tỏ ra vượt trội so với phương pháp mô phỏng vật lý. Chương trình mô phỏng hoàn toàn có thể được tích hợp vào phòng mô phỏng buồng lái tại Trường Đại học Hàng hải Việt Nam để thực hiện diễn tập ứng phó sự cố tràn dầu trên biển.

## Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong đề tài mã số: DT22-23.02.

## TÀI LIỆU THAM KHẢO

[1] M. Müller, S. Schirm, and S. Duthaler (2007), *Screen Space Meshes*, in Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, pp.9-15.

- [2] C. Brito, S. Almeida et al. (2017), *Screen Space Rendering Solution for Multiphase SPH Simulation*, in Proceedings of the 19<sup>th</sup> Symposium on Virtual and Augmented Reality, pp.309-318.
- [3] J. Stam (1999), *Stable Fluids*. In SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pp.121-128.
- [4] K. Kakuda, T. Nagashima et al. (2012), *Particle-based Fluid Flow Simulations on GPGPU Using CUDA*. Computer Modeling in Engineering and Sciences, Vol.88, No.1, pp.17-28.
- [5] Fay J. A (1969), *The spread of oil slicks on a calm sea*. Oil on the Sea, New York: Ed. Plenum Press, pp.53-63.
- [6] Lehr W J, Cekirge H M, Fraga R J, et al (1984), *Empirical studies of the spreading of oil spills*. Oil and Petrochemical Pollution, Vol.2, pp.7-12.
- [7] M. Fingas (2015), *Chapter 8 - Introduction to spill modeling*, pp.187-200.
- [8] Đỗ Văn Cường, Trần Gia Ninh, Nguyễn Lê Kim Phúc (2022), *Mô phỏng quá trình chuyển động và phân tách của màng dầu trên biển dựa trên thuật toán phân chia điểm Poisson*. Tạp chí Khoa học Công nghệ Hàng hải, Số 70 (04/2022), tr.09-14.
- [9] W. Lorensen and H. Cline (1987). *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*. In Proc. of the SIGGRAPH'87, pp.163-169.

Ngày nhận bài:	18/12/2022
Ngày nhận bản sửa:	27/12/2022
Ngày duyệt đăng:	30/12/2022