

NGHIÊN CỨU PHƯƠNG PHÁP TẠO LƯỚI TÍNH TOÁN VÀ TỐI ƯU HÓA CHẤT LƯỢNG LƯỚI SỬ DỤNG DISTMESH

A STUDY ON GENERATION AND OPTIMIZATION TECHNIQUES USING DISTMESH

Ngô Văn Quyết, Phạm Thị Thanh Hương*

Khoa Cơ khí Động lực, Trường Cơ khí, Đại học Bách khoa Hà Nội

TÓM TẮT

Nghiên cứu này, giới thiệu phương pháp sinh lưới không cấu trúc và tối ưu hóa bằng DistMesh. Phương pháp này sử dụng thuật toán tối ưu lực phục hồi để cải thiện chất lượng lưới qua từng bước và chiếu các nút biên dựa trên định nghĩa hình học ẩn. DistMesh tự động tính toán hàm kích thước lưới dựa trên độ cong và đặc trưng hình học, đồng thời kết hợp phương pháp giới hạn gradient tối ưu. Phương pháp này có thể áp dụng cho mọi chiều không gian và có ứng dụng rộng rãi, bao gồm lưới di động, động lực học chất lỏng, tối ưu hóa hình dạng, biến dạng cấu trúc, và thích nghi số học. Kết quả nghiên cứu cho thấy, DistMesh tạo ra các lưới có chất lượng cao và hiệu quả tính toán vượt trội so với các phương pháp tạo lưới truyền thống khác.

Từ khóa: DistMesh; Lưới chất lượng cao; Hình học ẩn.

ABSTRACT

This study introduces unstructured mesh generation techniques using DistMesh. The method iteratively improves the mesh by solving force equilibrium equations and projecting boundary nodes according to the definition of implicit geometry, applicable in any spatial dimension. The algorithm computes the mesh size function automatically based on curvature and geometric features, combined with optimal gradient limiting methods. Applications of the algorithm include mobile meshing, fluid dynamics, shape optimization, structural deformation, and adaptive numerics. Results demonstrate that this method produces high-quality meshes and superior computational efficiency compared to other mesh generators.

Keywords: DistMesh; High quality mesh; Implicit geometry.



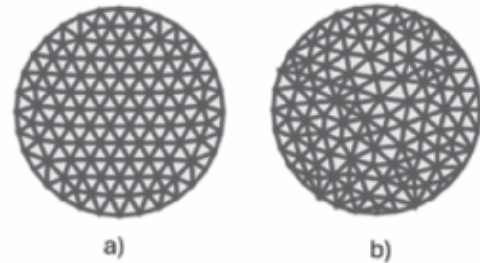
1. TỔNG QUAN

Quá trình tạo lưới đóng vai trò rất quan trọng trong nhiều ứng dụng, từ mô phỏng tính toán đến đồ họa máy tính [1]. Một lưới không cấu trúc cơ bản bao gồm việc chọn các điểm lưới và phân chia chúng thành các tam giác. Để xử lý các chi tiết phức tạp của hình học, lưới cần có các phần tử nhỏ. Tuy nhiên, trong một số trường hợp, việc sử dụng các phần tử lớn hơn cũng có thể hữu ích để giảm độ phức tạp của lưới. Trong các ứng dụng số, việc sử dụng các phần tử chất lượng cao, như các tam giác gần đều, là cực kỳ quan trọng để đảm bảo tính chính xác của quá trình phân tán và tính ổn định của hệ phương trình tuyến tính.

Một trong những thuật toán tạo lưới phổ biến nhất là phương pháp tinh chỉnh Delaunay[2]. Thuật toán này bắt đầu bằng cách chia một tập hợp các điểm ban đầu thành các tam giác và sau đó tinh chỉnh lưới cho đến khi chất lượng của các phần tử đạt được mức độ mong muốn. Thường thì thuật toán này hoạt động từ phía dưới lên, bắt đầu từ việc phân chia các điểm, sau đó là các đường cong, bề mặt và cuối cùng là thể tích. Phương pháp tiếp cận mặt [3] cũng là một phương pháp phổ biến, nơi mà quá trình tạo lưới bắt đầu từ biên và tạo ra các phần tử lưới theo một mặt tiếp cận vào trong miền.

DistMesh là một công cụ mạnh mẽ cho việc tạo lưới không cấu trúc, đặc biệt là khi cần tối ưu hóa chất lượng của lưới, công cụ này được viết hoàn toàn bằng ngôn ngữ MATLAB. Ở hình 1 có thể thấy rõ được các phần tử lưới được sinh ra từ DistMesh có chất lượng đồng đều và cao hơn so với phương pháp Delaunay thông thường. Trong nghiên cứu này, chúng tôi sẽ tập trung vào mô phỏng số để so sánh kết quả giữa các lưới khác nhau được sinh ra từ

DistMesh và từ phương pháp Delaunay trong Star-CCM+.



Hình 1. Mô tả các phương pháp tạo lưới khác nhau cho hình tròn, a) Lưới được sinh ra từ DistMesh và b) Lưới được sinh ra bằng phương pháp Delaunay.

2. PHƯƠNG PHÁP SINH LƯỚI VÀ TỐI ƯU HÓA

Thuật toán tạo lưới này dựa trên một phép tương tự cơ học giữa một lưới tam giác và cấu trúc giàn 2-D, hay cấu trúc lò xo [4], [5]. Thuật toán bắt đầu bằng cách sử dụng thuật toán Delaunay để tạo lưới tam giác cho tập hợp các điểm trong mặt phẳng x, y . Đối với mô hình vật lý này, các cạnh của tam giác (các đường nối giữa các điểm) được coi như các thanh của giàn, và các điểm được coi như các khớp nối. Mỗi thanh có một mối quan hệ giữa lực và biến dạng $f(l, l_0)$, phụ thuộc vào chiều dài hiện tại l và chiều dài ban đầu không bị kéo giãn l_0 của nó. Các ngoại lực tác động lên cấu trúc được áp dụng tại các điểm biên. Tại mỗi nút biên, có một phản lực vuông góc với biên, có độ lớn đủ để ngăn nút di chuyển ra ngoài.

Nhiệm vụ của thuật toán là tìm vị trí của các khớp sao cho toàn bộ cấu trúc đạt trạng thái cân bằng lực tĩnh. Khi $h(x, y) = 1$ (hàm này để xác định độ dài cân bằng của các cạnh), mục tiêu là chiều dài của tất cả các thanh tại trạng thái cân bằng sẽ gần như bằng nhau, tạo ra một lưới tam giác đều và có hình dạng tốt.

Để giải quyết trạng thái cân bằng lực, chúng ta cần thu thập tọa độ x và y của tất cả các điểm lưới N vào một mảng $N \times 2$, gọi là p :

$$p = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}$$

Vector lực $F(p)$ tại mỗi điểm lưới sẽ được chia ra thành các thành phần theo phương ngang và dọc, ký hiệu là $F_x(p)$ và $F_y(p)$. Các thành phần này được tính dựa trên tổng lực từ tất cả các thanh kết nối. Mỗi thanh tạo ra một lực tỷ lệ với sự kéo giãn hoặc nén so với chiều dài ban đầu của nó. Điều kiện cân bằng là tổng các thành phần lực này tại mỗi điểm bằng không, đảm bảo không có chuyển động thuần và do đó lưới ở trạng thái ổn định.

$$F(p) = [F_{int,x}(p) \ F_{int,y}(p)] + [F_{ext,x}(p) \ F_{ext,y}(p)]$$

Ở đây, F_{int} chứa các lực bên trong từ các thanh và F_{ext} là các lực bên ngoài (phản lực từ các ranh giới).

Lưu ý rằng $F(p)$ phụ thuộc vào cấu trúc của các thanh nối giữa các khớp. Trong thuật toán, cấu trúc này được xác định bằng cách sử dụng phân chia Delaunay của các điểm trên lưới. Delaunay tạo ra các tam giác không giao nhau để lấp đầy vùng nằm phía trong của các điểm đầu vào bao quanh nó, sao cho mỗi cạnh chỉ được chia sẻ bởi tối đa hai tam giác và đường tròn ngoại tiếp của mỗi tam giác không chứa bất kỳ điểm đầu vào nào khác. Trên mặt phẳng, việc phân chia tam giác này được biết đến để tối đa hóa góc nhỏ nhất của tất cả các tam giác. Đáng chú ý, vector lực $F(p)$ không thể được coi là một hàm liên tục của p do cấu trúc (sự có hoặc không có các thanh nối) được thay đổi bởi Delaunay khi các điểm di chuyển.

Hệ phương trình $F(p) = 0$ cần được giải để tìm ra một tập hợp các vị trí cân bằng p . Đây là một vấn đề khá khó khăn, một phần do tính không liên tục trong hàm lực (sự thay đổi của hình học) và một phần do sự hiện diện của các ngoại lực tại biên.

Một cách tiếp cận đơn giản để giải hệ phương trình $F(p) = 0$ là đưa vào một yếu tố phụ thuộc vào thời gian ảo. Đối với một số $p(0) = p_0$, chúng ta có thể xem xét một hệ các phương trình vi phân (trong hệ không thứ nguyên).

$$\frac{dp}{dt} = F(p), \quad t \geq 0.$$

Nếu tìm được một nghiệm tĩnh, nó sẽ thỏa mãn hệ phương trình $F(p) = 0$. Hệ phương trình (2.3) được xấp xỉ bằng phương pháp Euler tiến (forward Euler method). Tại các thời điểm rời rạc (giả định) $t_n = n\Delta t$, nghiệm xấp xỉ $p_n \approx p(t_n)$ được cập nhật bằng cách:

$$p_{n+1} = p_n + \Delta t F(p_n),$$

Khi đánh giá hàm lực, các vị trí p_n được biết trước, cấu trúc giàn (phân chia tam giác của tập điểm hiện tại) cũng được xác định. Các lực hồi phục bên ngoài được xử lý theo cách sau: Tất cả các điểm vượt ra ngoài vùng trong quá trình cập nhật từ p_n đến p_{n+1} sẽ được di chuyển trở lại điểm biên gần nhất. Điều này tuân thủ yêu cầu rằng các lực tác động vuông góc với biên, các điểm chỉ có thể di chuyển dọc theo biên, nhưng không được đi ra ngoài.

Có nhiều lựa chọn khác nhau cho hàm lực $f(l, l_0)$ trong mỗi thanh và một số lựa chọn đã được nghiên cứu [6], [7]. Hàm $k(l_0 - l)$ mô phỏng lò xo tuyến tính thông thường. Trong triển khai của chúng tôi, phản ứng tuyến tính này được sử dụng cho các lực đẩy nhưng không cho phép lực kéo: 

$$f(l, l_0) = \begin{cases} k(l_0 - l) & \text{khi } l < l_0 \\ 0 & \text{khi } l \geq l_0 \end{cases}$$

Các hàm lực phi tuyến có thể tạo ra các lưới tốt hơn (ví dụ với $k(l_0+1)/2l_0$), nhưng hàm lực tuyến tính từng đoạn lại cho kết quả tốt (k được thêm vào để đảm bảo đơn vị chính xác; k được đặt bằng 1). Việc yêu cầu $f=0$ khi $l=l_0$ là hợp lý. Cách xử lý biên được đề xuất có nghĩa là không điểm nào bị ép buộc phải ở lại biên, chúng chỉ bị ngăn không cho vượt qua biên. Do đó, điều quan trọng là hầu hết các thanh phải sinh ra lực đẩy $f > 0$, để giúp các điểm lan tỏa khắp toàn bộ hình học. Điều này có nghĩa là $f(l, l_0)$ nên dương khi l gần bằng chiều dài mong muốn, điều này có thể đạt được bằng cách chọn l_0 lớn hơn một chút so với chiều dài thực sự mong muốn (một giá trị mặc định tốt trong 2-D là 20%, điều này dẫn đến $F_{scale} = 1.2$).

Đối với các lưới đồng đều, l_0 là hằng số. Tuy nhiên, có nhiều trường hợp việc có các kích thước khác nhau ở các vùng khác nhau sẽ có lợi. Ở những khu vực có hình học phức tạp, cần sử dụng các phần tử nhỏ để giải quyết vấn đề (tính thích ứng hình học). Phương pháp giải có thể yêu cầu các phần tử nhỏ gần các điểm kỳ dị để đạt độ chính xác toàn cục tốt (trình giải thích ứng). Một lưới đồng đều với các phần tử nhỏ này sẽ đòi hỏi rất nhiều nút. Trong triển khai này, phân bố chiều dài cạnh mong muốn được cung cấp bởi người dùng dưới dạng một hàm kích thước phần tử (size function) $h(x,y)$. Lưu ý rằng $h(x,y)$ không nhất thiết phải bằng kích thước thực tế; nó chỉ đưa ra phân bố tương đối trên toàn miền. Điều này tránh sự liên kết ngầm với số lượng nút, mà không cần người dùng phải chỉ định. Ví dụ, nếu $h(x,y) = 1 + x$ trong hình vuông đơn vị, thì chiều dài cạnh gần biên trái ($x = 0$) sẽ khoảng một nửa chiều dài cạnh gần biên phải ($x = 1$). Điều này đúng bất kể số lượng điểm và kích thước phần tử thực tế.

Để tìm hệ số tỷ lệ, tỷ lệ giữa diện tích lưới từ các chiều dài cạnh thực tế l_i và "kích thước mong muốn" (từ $h(x,y)$ tại trung điểm (x_i, y_i) của các thanh) được tính như sau:

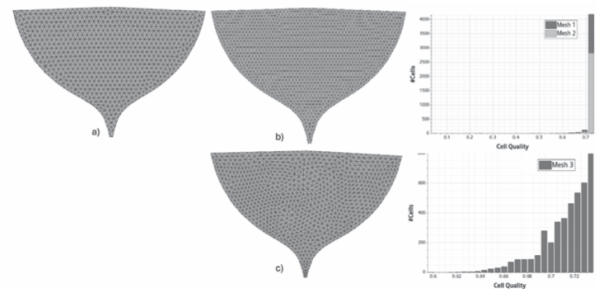
$$\text{Hệ số tỷ lệ} = \left(\frac{\sum l_i^2}{\sum h(x_i, y_i)^2} \right)^{1/2}$$

Điều này đảm bảo rằng phân bố kích thước phần tử được duy trì đúng theo mong muốn của người dùng mà không cần xác định chính xác số lượng nút.

3. MÔ PHỎNG ĐÁNH GIÁ LƯỚI

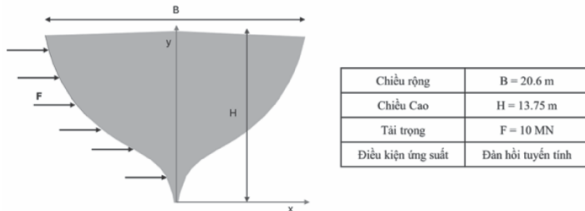
3.1. Mô hình mô phỏng

Mô hình được sử dụng để mô phỏng trong nghiên cứu này là mô hình mặt cắt ngang 2D tại sườn 0.5 của một tàu container 25,000 DWT với kích thước tiêu chuẩn. Chúng tôi tiến hành khảo sát trên ba loại lưới khác nhau: Lưới 1 (a) được tạo ra bằng DistMesh với kích thước phần tử xấp xỉ 0.5 m, lưới 2 (b) được tạo ra bằng DistMesh với kích thước phần tử mịn hơn là 0.2 m và lưới 3 (c) được tạo ra bằng phần mềm Simcenter STAR-CCM+ sử dụng phương pháp Delaunay với kích thước phần tử xấp xỉ 0.5 m, như được minh họa trong Hình 2. Mô hình được sử dụng để tính toán là mô hình Solid Stress, và quá trình mô phỏng được thực hiện trên phần mềm Simcenter STAR-CCM+.



Hình 2. Biểu đồ so sánh chất lượng phần tử của 2 phương pháp chia lưới

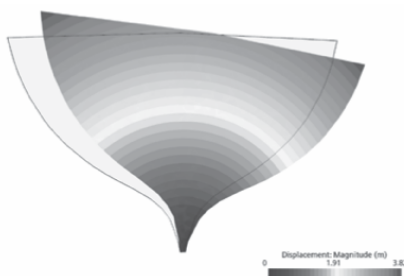
Trong mô hình vật lý, chúng tôi giả định một lực do sóng biển gây ra theo phương ngang tác động từ bên hông của con tàu. Trong trường hợp này, tất cả các sườn của tàu sẽ chịu lực phân bố này. Chúng tôi phân tích lực tác động lên một mặt cắt ngang của tàu như được mô tả trong Hình 3.



Hình 3. Mô hình vật lý và bảng thông số

3.2. Phân tích kết quả

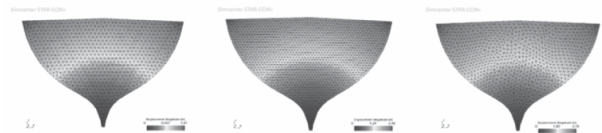
Kết quả bài mô phỏng so sánh giữa các lưới (hình 4, 5) cho thấy sự khác biệt đáng kể về các thông số quan trọng có thể quan sát trong bảng 1: Độ dịch chuyển lớn nhất theo phương x, ứng suất lớn nhất theo phương x, và tổng thời gian xử lý CPU của bộ giải. Có thể thấy rằng, lưới Delaunay cho kết quả độ dịch chuyển lớn nhất và ứng suất cao nhất, trong khi hai lưới DistMesh cho kết quả gần nhau hơn và thấp hơn về cả độ dịch chuyển và ứng suất. Tuy nhiên, về tổng thời gian xử lý CPU, DistMesh (Lưới 1) có thời gian xử lý nhanh nhất, cho thấy hiệu suất tính toán cao hơn so với lưới Delaunay cùng kích thước.



Hình 4. Mô tả vị trí của mô hình sau khi tác động lực

Bảng 1. Tổng hợp kết quả - đánh giá Độ Dịch Chuyển, CPU Time và Ứng Suất Tối Đa trong mô phỏng của 3 lưới khác nhau:

Phương pháp tạo lưới	Độ dịch chuyển lớn nhất theo phương x	$(\sigma_{xx})_{max}$	Tổng thời gian xử lý CPU của bộ giải
DistMesh (Lưới 1)	1.454102 (m)	2.532733e+09 (Pa)	4.982 (s)
DistMesh (Lưới 2)	1.519024 (m)	2.602335e+09 (Pa)	6.234 (s)
Delaunay (Star-CCM+)	2.038736 (m)	2.766248e+09 (Pa)	6.152 (s)



Hình 5. Kết quả mô phỏng lưới

4. KẾT LUẬN

Nghiên cứu này đã giới thiệu và áp dụng các kỹ thuật sinh lưới không cấu trúc sử dụng DistMesh để mô phỏng một mặt cắt ngang của tàu container 25,000 DWT. Chúng tôi đã tiến hành khảo sát trên ba loại lưới khác nhau và so sánh chất lượng phần tử cũng như hiệu quả tính toán của từng loại lưới. Kết quả cho thấy rằng, phương pháp DistMesh tạo ra các lưới có chất lượng cao và hiệu quả tính toán vượt trội hơn so với các phương pháp chia lưới truyền thống, đặc biệt là khi sử dụng lưới mịn hơn với kích thước phần tử 0.2 m.

Lưới tạo ra từ phần mềm Simcenter STAR-CCM+ với kích thước phần tử 0.5 m cũng cho thấy hiệu quả tốt, nhưng chưa đạt đến mức độ tinh tế và chất lượng như các lưới tạo ra bằng DistMesh. Phân tích lực tác động lên mặt cắt ngang của tàu cho thấy rằng lưới có chất lượng cao đảm bảo tính chính xác của kết quả mô phỏng và ổn định của hệ phương trình tuyến tính.

Nghiên cứu này, đóng góp vào việc mở rộng hiểu biết về việc áp dụng các phương pháp tạo lưới và tối ưu hóa trong việc mô phỏng các

vấn đề tính toán phức tạp. Điều này giúp cải thiện sự chính xác và hiệu quả của các phân tích kỹ thuật trong lĩnh vực cơ khí và hàng hải. ❖

Ngày nhận bài: **23/5/2024**

Ngày phản biện: **12/6/2024**

Tài liệu tham khảo:

- [1]. Persson, P.-O., & Strang, G. (2004). *A simple mesh generator in MATLAB*. SIAM Review, 46(2), 329-345..
- [2]. Shewchuk, J. R. (1996). *Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator*. Applied Computational Geometry: Towards Geometric Engineering, 1148, 203-222.
- [3]. Frey, P. J., & George, P. L. (2008). *Mesh Generation: Application to Finite Elements*. Wiley-ISTE.
- [4]. Baker, T. J. (1989). *Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation*. Engineering with Computers, 5(3), 161-175.
- [5]. Field, D. A. (1988). *Laplacian smoothing and Delaunay triangulations*. Communications in Applied Numerical Methods, 4(6), 709-712.
- [6]. Frank J. Bossen and Paul S. Heckbert. *A pliant method for anisotropic mesh generation*. In Proceedings of the 5th International Meshing Roundtable, pages 63–76. Sandia Nat. Lab., 1996.
- [7]. Kenji Shimada and David C. Gossard. *Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing*. In SMA '95: Proceedings of the Third Symposium on Solid Modeling and Applications, pages 409–419, 1995.