

ĐẠI HỌC THÁI NGUYÊN  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ  
THÔNG TIN VÀ TRUYỀN THÔNG

---

**BÁO CÁO TỔNG KẾT**  
**ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP ĐẠI HỌC**

**XỬ LÝ SONG SONG TRONG PHƯƠNG PHÁP RBF-FD**  
**GIẢI BÀI TOÁN TRUYỀN NHIỆT**

**Mã số: ĐH2015-TN07-03**

**Chủ nhiệm đề tài: ThS. Ngô Mạnh Tường**

**Thái Nguyên, tháng 3 năm 2019**

ĐẠI HỌC THÁI NGUYÊN  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ  
THÔNG TIN VÀ TRUYỀN THÔNG

---

**BÁO CÁO TỔNG KẾT  
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP ĐẠI HỌC**

**XỬ LÝ SONG SONG TRONG PHƯƠNG PHÁP RBF-FD  
GIẢI BÀI TOÁN TRUYỀN NHIỆT**

**Mã số: ĐH2015-TN07-03**

Xác nhận của tổ chức chủ trì

Chủ nhiệm đề tài

Thái Nguyên, tháng 3 năm 2019

## DANH SÁCH THÀNH VIÊN THAM GIA ĐỀ TÀI

TT	Họ tên	Đơn vị công tác và lĩnh vực chuyên môn	Nhiệm vụ
1	Đặng Thị Oanh	Đơn vị công tác: Bộ môn Khoa học máy tính - Trường ĐH Công nghệ thông tin và Truyền thông Chuyên môn: Công nghệ thông tin	Cố vấn chuyên môn
2	Trịnh Minh Đức	Đơn vị công tác: Bộ môn Mạng và An toàn thông tin - Trường ĐH Công nghệ thông tin và Truyền thông Chuyên môn: Công nghệ thông tin	Cài đặt thuật toán
3	Dương Thị Nhung	Đơn vị công tác: Phòng KH-CN&HTQT-Trường ĐH Công nghệ thông tin và Truyền thông Chuyên môn: Công nghệ thông tin	Thư ký hành chính

## ĐƠN VỊ PHỐI HỢP

TT	Tên đơn vị	Nội dung phối hợp	Họ và tên người đại diện	Ghi chú
1	Trường ĐH Giessen, Germany	Thảo luận chuyên môn, viết chung bài báo quốc tế	GS. Oleg Davydov	

# Mục lục

PHẦN MỞ ĐẦU .....	1
Chương 1. Kiến thức cơ sở .....	3
1.1. Bài toán truyền nhiệt dừng .....	3
1.2. Nội suy dữ liệu phân tán .....	6
1.3. Hàm cơ sở bán kính .....	7
1.4. Ma trận xác định dương, hàm xác định dương .....	8
1.5. Nội suy hàm cơ sở bán kính .....	9
1.6. Một số kiến thức cơ bản về lý thuyết xử lý song song .....	10
Chương 2. Tính toán song song cho phương pháp RBF-FD trong không gian hai chiều .....	15
2.1. Véc tơ trọng số trong công thức vi phân số .....	15
2.2. Rời rạc bài toán truyền nhiệt dừng .....	17
2.3. Thuật toán chọn tâm .....	18
2.4. Song song hóa việc chọn tâm và tính véc tơ trọng số .....	21
2.5. Thử nghiệm số .....	24
2.6. Kết luận .....	27
Chương 3. Phương pháp không lưới RBF-FD trong không gian ba chiều .....	29
3.1. Phương pháp không lưới RBF-FD .....	29
3.2. Thuật toán chọn tâm dựa trên các góc khối .....	31
3.3. Thử nghiệm số .....	34
3.4. Kết luận .....	41
KẾT LUẬN .....	43
Tài liệu tham khảo .....	44

# Danh sách bảng

1.1	Một số hàm cơ sở bán kính, trong đó $r = \ x - x_k\ $ . . . . .	8
1.2	Một số hàm cơ sở bán kính với tham số hình dạng $\delta > 0$ . . . . .	8
2.1	Các sai số <i>rms</i> so với nghiệm chính xác của FEM và phương pháp RBF-FD, thời gian chạy của phương pháp RBF-FD trên các bộ xử lý của Bài toán 1. . . . .	25
2.2	Các sai số <i>rms</i> so với nghiệm chính xác của FEM và phương pháp RBF-FD, thời gian chạy của phương pháp RBF-FD trên các bộ xử lý của Bài toán 2. . . . .	27
3.1	8 - góc khối . . . . .	31
3.2	16 - góc khối. . . . .	33
3.3	Các sai số RRMS so với nghiệm chính xác của Bài toán 3. . . . .	36
3.4	Các sai số RRMS so với nghiệm chính xác của Bài toán 4. . . . .	37
3.5	Các sai số RRMS so với nghiệm của lần kế tiếp trên cùng một phương pháp ( $E_n$ ) và so với nghiệm đạt được bởi FEM ( $E_r$ ) của Bài toán 5. . . . .	41

# Danh sách hình vẽ

- 2.1 Lưu đồ song song hóa thuật toán hỗ trợ chọn tâm ODP và tính véc tơ trọng số. . . 23
- 2.2 Kết quả thử nghiệm số của Bài toán 1: (a) Miền  $\Omega$  của phương pháp RBF-FD. (b) Phân luồng dữ liệu chạy song song trên 4 bộ xử lý của phương pháp RBF-FD. (c) Các sai số *rms* của FEM và phương pháp RBF-FD. (d) Thời gian chạy của phương pháp RBF-FD trên các bộ xử lý. . . . . 26
- 2.3 Kết quả thử nghiệm số của Bài toán 2: (a) Miền  $\Omega$  của phương pháp RBF-FD. (b) Phân luồng dữ liệu chạy song song trên 4 bộ xử lý của phương pháp RBF-FD. (c) Các sai số *rms* của FEM và phương pháp RBF-FD. (d) Thời gian chạy của phương pháp RBF-FD trên các bộ xử lý. . . . . 28
- 3.1 Kết quả thử nghiệm số của Bài toán 3. Hình (a) minh họa sai số RRMS trên các tâm trong miền. Hình (b) biểu diễn mật độ của ma trận thưa của hệ (3.3) ứng với phương pháp RBF-FD và của ma trận cứng ứng với FEM. . . . . 36
- 3.2 Kết quả thử nghiệm số của Bài toán 4: (a) Các sai số RRMS trên các nút trong miền. (b) Mật độ của ma trận hệ số. . . . . 37
- 3.3 (a) Miền  $\Omega$  được tạo bởi lệnh `pdegplot` trong MATLAB PDE Toolbox. (b)–(d) Đồ thị các sai số RRMS và mật độ của ma trận hệ số sử dụng các thuật toán RBF-FD 1, 2, 3 và FEM như trong Bảng 3.5. (e)–(f) Đồ thị các sai số RRMS và mật độ của ma trận hệ số ứng với phương pháp RBF-FD chọn các tập tâm hỗ trợ tính trọng số có 16, 18 hoặc 20 điểm gần  $\zeta$  nhất và FEM. . . . . 39
- 3.4 Đường đồng mức các nghiệm của FEM (bên trái) và RBF-FD 1 (bên phải) với trường hợp miền rời rạc có 65652 điểm trong. (a)–(b) đường đồng mức theo biến  $z$ ; (c)–(d) đường đồng mức theo biến  $y$ ; (e)–(f) đường đồng mức theo biến  $x$  và bản màu trên mặt phẳng. . . . . 40

ĐẠI HỌC THÁI NGUYÊN

Đơn vị: Trường Đại học Công nghệ thông tin và Truyền thông

## THÔNG TIN KẾT QUẢ NGHIÊN CỨU

### 1. Thông tin chung

- Tên đề tài: Xử lý song song trong phương pháp RBF-FD giải bài toán truyền nhiệt.
- Mã số: ĐH2015-TN07-03.
- Chủ nhiệm đề tài: ThS. Ngô Mạnh Tường.
- Tổ chức chủ trì: Trường Đại học Công nghệ thông tin và Truyền thông - Đại học Thái Nguyên
- Thời gian thực hiện: Từ tháng 1 năm 2015 đến tháng 12 năm 2016 (gia hạn đến tháng 12 năm 2018).

### 2. Mục tiêu

- Xây dựng được thuật toán song song nhằm mục đích song song hóa quá trình xác định véc tơ trọng số trong phương pháp RBF-FD;
- Đánh giá hiệu quả của giải pháp song song hóa bằng lý thuyết và thử nghiệm;
- Thử nghiệm số trên tập dữ liệu phân tán lớn.

### 3. Tính mới và sáng tạo

- Đề xuất 3 thuật toán mới;
- Thử nghiệm số trên các thuật toán mới để so sánh với các thuật toán đã công bố.

### 4. Kết quả nghiên cứu

- Đề xuất thuật toán tính toán song song cho phương pháp RBF-FD trong việc tìm tập các tâm hỗ trợ tính toán véc tơ trọng số và tính toán véc tơ trọng số;

- Đề xuất 2 thuật toán chọn tâm hỗ trợ phương pháp không lưới RBF-FD trong không gian 3 chiều.

## 5. Sản phẩm

### Sản phẩm khoa học

- 01 bài báo đăng trên tạp chí quốc tế thuộc danh mục scopus;
  - 02 bài báo đăng trên tạp chí chuyên ngành;
  - 01 bài báo đăng trên kỷ yếu hội thảo Quốc gia.
1. Oleg Davydov, Dang Thi Oanh and Ngo Mạnh Tuong (2019), " Octant-Based Stencil Selection for Meshless Finite Difference Methods in 3D", *Vietnam journal of mathematics*, đã có chấp nhận đăng.
  2. Đặng Thị Oanh, Ngô Mạnh Tường (2019), "Song song hóa việc chọn tâm và tính véc tơ trọng số cho phương pháp không lưới RBF-FD giải phương trình Poisson", *Tạp chí Khoa học và Công nghệ - Đại học Thái Nguyên*, tập 195, số 02, tr. 69-74.
  3. Đặng Thị Oanh, Ngô Mạnh Tường, Trịnh Minh Đức (2017), " Nghiên cứu sự ảnh hưởng của điều kiện tách biệt trong thuật toán sinh tâm thích nghi cho phương pháp không lưới giải bài toán elliptic ", *Kỷ yếu Hội nghị Khoa học Công nghệ quốc gia lần thứ X (Fair 2017) về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin*, tr. 667-673.
  4. Ngô Mạnh Tường (2015), "Phương pháp không lưới RBF-FD sử dụng nội suy Hermite RBF giải phương trình Poisson", *Tạp chí Khoa học và Công nghệ - Đại học Thái Nguyên*, tập 132, số 02, tr. 171-175.

**Sản phẩm đào tạo:** Hướng dẫn 02 luận văn thạc sĩ bảo vệ thành công:

1. Lâm Văn Trì, *Nghiên cứu sự ảnh hưởng của bộ tâm nội suy đến độ chính xác của xấp xỉ đạo hàm dựa trên nội suy hàm cơ sở bán kính*, Luận văn tốt nghiệp năm 2016, Quyết định số 812/QĐ-ĐHCNTT&TT ngày 18/10/2016, tại Trường Đại học Công nghệ Thông tin và Truyền thông - Đại học Thái Nguyên.



2. Vũ Huy Hoàng Đô, *Sự ảnh hưởng của bộ tâm được chọn trong phương pháp không lưới RBF-FD*, Luận văn tốt nghiệp năm 2016, Quyết định số 49/QĐ-ĐHCNTT&TT ngày 18/01/2016, tại Trường Đại học Công nghệ Thông tin và Truyền thông - Đại học Thái Nguyên.

**6. Phương thức chuyển giao, địa chỉ ứng dụng, tác động và lợi ích mang lại của kết quả nghiên cứu:**

Các bài báo và các công trình công bố trong các hội thảo sẽ là tài liệu cho những nghiên cứu sâu hơn trong chủ đề này. Kết quả nghiên cứu của đề tài cũng là nguồn tài liệu tham khảo hữu ích cho công tác giảng dạy cho sinh viên tại trường Đại học Công nghệ thông tin và Truyền thông - Đại học Thái Nguyên.

*Ngày ..... tháng ..... năm 2019*

**Tổ chức chủ trì**

**Chủ nhiệm đề tài ...**

## INFORMATION ON RESEARCH RESULTS

### 1. General information:

- Project title: Parallel Processing For RBF-FD Method to Solve Heat Transfer Problem
- Code number: ĐH2015-TN07-03.
- Coordinator: M.Sc. Ngo Manh Tuong.
- Implementing institution: TNU - University of Information and Communication Technology .
- Duration: From January 2015 to December 2016 (extended until December 2018).

### 2. Objective(s):

- Construct a parallel algorithm for goal to parallel of process determining weight vector for RBF-FD method;
- Evaluate the effectiveness of parallelization solutions by theory and experiment;
- Experiment on large scattered data.

### 3. Creativeness and innovativeness:

- Propose 3 new algorithms;
- Numerical test on the new algorithms to compare with published algorithms.

### 4. Research results:

- Propose a parallel computing algorithm for the RBF-FD method in finding stencil support selection and computing the weight vector;
- Propose two stencil support selection algorithms for RBF-FD method in 3D.

### 5. Products:

#### Scientific product

- A paper in scopus international journal;

- Two papers in national journal;
  - A paper in proceedings of national conference.
1. Oleg Davydov, Dang Thi Oanh and Ngo Manh Tuong (2019), Octant-Based Stencil Selection for Meshless Finite Difference Methods in 3D", *Vietnam journal of mathematics*, accepted to publication.
  2. Dang Thi Oanh, Ngo Manh Tuong (2019), "Parallelization in choose the centers and compute the weight vectors for the meshless RBF-FD to solve poisson equation", *TNU-Journal of science and technology*, Vol. 195, No. 02, p. 69 - 74.
  3. Dang Thi Oanh, Ngo Manh Tuong, Trinh Minh Duc (2017), "Research the influence of separate conditions in adaptive refinement algorithm for meshless method to solve elliptic problem, *Proceedings of national conference (Fair 2017)*, pp.667-673.
  4. Ngo Manh Tuong (2015), "The RBF-FD meshless method based on RBF Hermite interpolation for solving Poisson equation", *TNU-Journal of science and technology*, Vol. 132, No. 02, p. 171 - 175.

**Product training:** Graduate study guide 02 Master's theses.

1. Lam Van Tri, *Research the influence of the center interpolation to the accuracy of approximating the derivative based on the radial basis interpolation function*, Graduation thesis in 2016, number 812/QĐ-ĐHCNTT&TT,18/10/2016, at TNU - University of Information and Communication Technology.
2. Vu Huy Hoang Do, *The influence of stencils selection to meshless RBF-FD method*, Graduation thesis in 2016, number 49/QĐ-ĐHCNTT&TT,18/01/2016, at TNU - University of Information and Communication Technology.

## **6. Transfer alternatives, application institutions, impacts and benefits of research results:**

Articles and work published in seminars will be the materials for further research in the topic. The research results of the project are also useful references for teaching students at Thai Nguyen University of Information & Communication Technology.

## PHẦN MỞ ĐẦU

### 1. Tính cấp thiết của vấn đề nghiên cứu

Phương pháp RBF - FD [6, 10, 11, 12, 13, 14, 17, 18, 20, 22, 23] là phương pháp không lưới sử dụng nội suy hàm RBF (Radial Basis Function) với cách tiếp cận địa phương và dựa trên sự rời rạc hóa giống như phương pháp FD (finite different). Phương pháp này sử dụng các tập điểm rời rạc trong miền xác định để tính nghiệm tại các điểm này. Một lợi thế của kỹ thuật rời rạc không lưới là chỉ cần dựa trên tập điểm độc lập phân bố bất kỳ. Do đó, chi phí dành cho sinh lưới, duy trì lưới và cập nhật lưới được loại trừ.

Các nghiên cứu về phương pháp RBF-FD và ứng dụng giải phương trình đạo hàm riêng, đã được các nhà khoa học trong và ngoài nước quan tâm. Các kết quả trong hướng nghiên cứu này đã đạt được là: Xây dựng được một số cách tính véc tơ trọng số dựa trên ý tưởng của phương pháp FD và phương pháp phần tử hữu hạn FEM (finite element method), xây dựng thuật toán chọn tham số hình dạng tối ưu, xây dựng thuật toán chọn tâm nội suy, xây dựng thuật toán làm mịn thích nghi. Các kết quả thử nghiệm trong [10, 12, 18, 20, 23] cho thấy nghiệm xấp xỉ tìm được bởi phương pháp RBF-FD trên các bài toán mẫu chính xác hơn nhiều so với nghiệm xấp xỉ tìm được bởi phương pháp FD và FEM. Tuy nhiên, các nghiên cứu trước đây mới chỉ quan tâm đến độ chính xác của lời giải số.

Trong phương pháp RBF-FD [10, 12, 18, 20, 23], với mỗi tâm nằm trong miền ta cần xác định một véc tơ trọng số, véc tơ này được xác định khi xác định được bộ tâm nội suy nằm trong lân cận của nó và véc tơ trọng số là nghiệm của hệ phương trình với ma trận hệ số là ma trận thưa. Công đoạn xác định véc tơ trọng số là tốn kém thời gian và hoàn toàn độc lập nhau, chính vì vậy vấn đề song song hóa trong phương pháp RBF-FD là khả thi và cấp thiết để làm giảm đáng kể chi phí về mặt thời gian.

### 2. Mục tiêu, đối tượng, phạm vi, cách tiếp cận, phương pháp nghiên cứu

- Mục tiêu
  - Xây dựng được thuật toán song song nhằm mục đích song song hóa quá trình xác định véc tơ trọng số trong phương pháp RBF-FD;
  - Đánh giá hiệu quả của giải pháp song song hóa bằng lý thuyết và thử nghiệm;

– Thử nghiệm số trên tập dữ liệu phân tán lớn.

- Đối tượng nghiên cứu: Đề tài tập trung vào việc song song hoá quá trình chọn tâm và tính véc tơ trọng số của phương pháp RBF-FD.
- Phạm vi nghiên cứu: Bài toán toán truyền nhiệt dừng.
- Cách tiếp cận:

Do các vấn đề có liên quan là những vấn đề trên thế giới đang nghiên cứu và phát triển nên công việc đầu tiên cần phải tiến hành là thu thập và hệ thống hóa các tài liệu chuyên môn liên quan đến những kết quả gần đây nhất về phương pháp RBF-FD và các kỹ thuật xử lý song song.

Các bước tiến hành sẽ tuân theo quy trình: nghiên cứu, phân tích, thiết kế giải pháp, cài đặt. Việc hợp tác với các tập thể và cá nhân trong và ngoài nước sẽ được đặc biệt quan tâm, tận dụng tối đa chất xám và công nghệ mới, nâng cao chất lượng và hiệu quả thực hiện đề tài.

- Phương pháp nghiên cứu: Phương pháp nghiên cứu của đề tài được lựa chọn là lý thuyết kết hợp với thực nghiệm. Quá trình được tiến hành dựa trên việc tìm hiểu tài liệu, cài đặt thử nghiệm sau đó đánh giá kết quả thực nghiệm và cải tiến nhằm nâng cao chất lượng.

# Chương 1

## Kiến thức cơ sở

Trong chương này chúng tôi sẽ trình bày các kiến thức cơ sở liên quan đến đề tài, bao gồm: 1.1 Giới thiệu về bài toán truyền nhiệt dừng; 1.2 Khái niệm bài toán nội suy dữ liệu phân tán; 1.3 Khái niệm hàm cơ sở bán kính; 1.4 Ma trận xác định dương, hàm xác định dương; 1.5 Nội suy với hàm cơ sở bán kính và cuối cùng là 1.6 Một số kiến thức cơ bản về lý thuyết xử lý song song.

### 1.1. Bài toán truyền nhiệt dừng

#### Bài toán truyền nhiệt một chiều:

Phương trình truyền nhiệt một chiều [2] hay phương trình mô tả hiện tượng truyền nhiệt trong thanh vật chất đồng chất có dạng

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}, \quad a < x < b, \quad t > 0, \quad k = \text{const} > 0, \quad (1.1)$$

trong đó  $u = u(t, x)$  là nhiệt độ của thanh vật chất ở vị trí  $x$  tại thời điểm  $t$ ,  $\frac{\partial u}{\partial t}$  là mức độ thay đổi của nhiệt độ tại một điểm nào đó theo thời gian,  $\frac{\partial^2 u}{\partial x^2}$  là đạo hàm bậc 2 (lưu chuyển nhiệt) của hàm nhiệt độ,  $k$  là hằng số truyền nhiệt (phụ thuộc vào vật liệu, độ dẫn nhiệt, mật độ và dung tích nhiệt),  $a, b$  là các hằng số biểu diễn độ dài của thanh vật chất.

Trong trường hợp  $k = k(x, t, u)$ , tức là  $k$  phụ thuộc vào  $x, t, u$ , khi đó phương trình (1.1) có dạng

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ k(x, t, u) \frac{\partial}{\partial x} \right], \quad a < x < b, \quad t > 0.$$

Khi thanh vật chất còn có một nguồn nhiệt (sinh nhiệt hay hấp thụ nhiệt) đặc trưng bởi hàm  $f(x, t)$  thì ta có phương trình

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ k(x, t, u) \frac{\partial}{\partial x} \right] + f(x, t, u), \quad a < x < b, \quad t > 0.$$

Nếu  $k$  và  $f$  không phụ thuộc vào  $u$  thì ta có phương trình truyền nhiệt tuyến tính

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ k(x, t) \frac{\partial}{\partial x} \right] - q(x, t)u + f(x, t), \quad a < x < b, \quad t > 0.$$

Nếu đến một thời điểm nhất định sự phân bố nhiệt trên thanh vật chất ổn định, không thay đổi theo thời gian thì được gọi là hiện tượng truyền nhiệt đã dừng. Khi đó nhiệt độ không thay đổi theo thời gian nên  $\frac{\partial u}{\partial t} = 0$  và ta có phương trình truyền nhiệt dừng

$$\frac{d^2 u}{dx^2} = 0, \quad a < x < b$$

hay

$$\frac{d}{dx} \left[ k(x, u) \frac{du}{dx} \right] = f(x, u), \quad a < x < b$$

hay

$$\frac{d}{dx} \left[ k(x, u) \frac{du}{dx} \right] - q(x)u = f(x), \quad a < x < b.$$

### Bài toán truyền nhiệt hai chiều:

Phương trình truyền nhiệt trong môi trường phẳng đồng chất có dạng

$$\frac{\partial u}{\partial t} = k \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (x, y) \in \Omega, \quad k = \text{const.} \quad (1.2)$$

trong đó  $\Omega$  là bản mỏng vật chất có biên là đường cong khép kín  $\Gamma$  nằm trong mặt phẳng  $Oxy$ ,  $u = u(t, x, y)$  là nhiệt độ của thanh vật chất ở vị trí  $(x, y)$  tại thời điểm  $t$ ,  $\frac{\partial u}{\partial t}$  là mức độ thay đổi của nhiệt độ tại một điểm nào đó theo thời gian,  $\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}$  là đạo hàm bậc 2 (lưu chuyển nhiệt) của hàm nhiệt độ của nhiệt độ theo hướng  $x$  và  $y$ .

Trong trường hợp tổng quát phương trình (1.2) có dạng

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ f_1(x, y, t, u) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, t, u) \frac{\partial u}{\partial y} \right] + f(x, y, t, u), \quad (x, y) \in \Omega, \quad t > 0 \quad (1.3)$$

hay khi  $f_1, f_2, f$  không phụ thuộc vào  $u$  thì phương trình (1.3) có dạng

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ f_1(x, y, t) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, t) \frac{\partial u}{\partial y} \right] - q(x, y, t)u + f(x, y, t), \quad (x, y) \in \Omega, \quad t > 0. \quad (1.4)$$

Các phương trình (1.2), (1.3), (1.4) được gọi là các phương trình truyền nhiệt 2 chiều [2].

Nếu đến một thời điểm nhất định sự phân bố nhiệt trên bản mỏng vật chất ổn định, không thay đổi theo thời gian thì được gọi là hiện tượng truyền nhiệt dừng. Khi đó nhiệt độ không thay đổi theo thời gian nên  $\frac{\partial u}{\partial t} = 0$  và phương trình truyền nhiệt dừng có dạng

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (x, y) \in \Omega \quad (1.5)$$

hay

$$\begin{aligned} \frac{\partial}{\partial x} \left[ f_1(x, y, u) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, u) \frac{\partial u}{\partial y} \right] &= f(x, y, u), \quad (x, y) \in \Omega, \\ \frac{\partial}{\partial x} \left[ f_1(x, y) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y) \frac{\partial u}{\partial y} \right] - q(x, y) u &= f(x, y), \quad (x, y) \in \Omega. \end{aligned}$$

Phương trình (1.5) với vế phải khác 0 khi đó ta có phương trình Poisson hai chiều

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad (x, y) \in \Omega. \quad (1.6)$$

Bài toán tìm hàm  $u = u(x, y)$  thỏa mãn phương trình (1.6) với điều kiện biên

$$u(x, y) = g(x, y), \quad (x, y) \in \Gamma \quad (1.7)$$

được gọi là bài toán biên Dirichlet đối với phương trình Poisson.

Ý nghĩa vật lý của bài toán là mô tả sự phân bố nhiệt đã ổn định trong mặt phẳng  $\Omega$  khi phân bố nhiệt độ tại biên  $\Gamma$  của  $\Omega$  ổn định là  $g(x, y)$ .

### **Bài toán truyền nhiệt ba chiều:**

Tương tự như trong không gian hai chiều, thay bản mỏng vật chất bằng một khối vật chất  $\Omega$  có mặt biên khép kín  $\Gamma$ , đặt trong không gian  $Oxyz$ . Khi đó *phương trình truyền nhiệt trong môi trường không gian đồng chất* là

$$\frac{\partial u}{\partial t} = k \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right), \quad (x, y, z) \in \Omega, \quad k = \text{const} \quad (1.8)$$

hay dạng tổng quát của phương trình (1.8) là

$$\begin{aligned} \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ f_1(x, y, z, t, u) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, z, t, u) \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial z} \left[ f_3(x, y, z, t, u) \frac{\partial u}{\partial z} \right] \\ + f(x, y, z, t, u), \quad (x, y, z) \in \Omega, \quad t > 0. \end{aligned} \quad (1.9)$$

Nếu  $f_1, f_2, f_3, f$  không phụ thuộc vào  $u$  thì phương trình (1.9) có dạng

$$\begin{aligned} \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left[ f_1(x, y, z, t) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, z, t) \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial z} \left[ f_3(x, y, z, t) \frac{\partial u}{\partial z} \right] \\ - q(x, y, z, t) u + f(x, y, z, t), \quad (x, y, z) \in \Omega, \quad t > 0. \end{aligned} \quad (1.10)$$

Các phương trình (1.8), (1.9), (1.10) được gọi là các *phương trình truyền nhiệt 3 chiều* [2].

*Phương trình truyền nhiệt dừng ba chiều* có dạng

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0, \quad (x, y, z) \in \Omega$$



hay

$$\begin{aligned} & \frac{\partial}{\partial x} \left[ f_1(x, y, z, u) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, z, u) \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial z} \left[ f_3(x, y, z, u) \frac{\partial u}{\partial z} \right] \\ & \qquad \qquad \qquad = f(x, y, z, u), \quad (x, y, z) \in \Omega, \\ & \frac{\partial}{\partial x} \left[ f_1(x, y, z) \frac{\partial u}{\partial x} \right] + \frac{\partial}{\partial y} \left[ f_2(x, y, z) \frac{\partial u}{\partial y} \right] + \frac{\partial}{\partial z} \left[ f_2(x, y, z) \frac{\partial u}{\partial z} \right] - q(x, y, z) u \\ & \qquad \qquad \qquad = f(x, y, z), \quad (x, y, z) \in \Omega. \end{aligned}$$

Phương trình Poisson với điều kiện biên Dirichlet trong không gian là bài toán tìm hàm  $u = u(x, y, z)$  thỏa mãn

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(x, y, z), \quad (x, y, z) \in \Omega, \quad (1.11)$$

$$u(x, y, z) = g(x, y, z), \quad (x, y, z) \in \Gamma. \quad (1.12)$$

## 1.2. Nội suy dữ liệu phân tán

Cho bộ dữ liệu  $(x_i, y_i), i = 1, 2, \dots, n, x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$ , trong đó  $x_i$  là các vị trí đo,  $y_i$  là các kết quả tại vị trí đo. Giả sử  $u_1, u_2, \dots, u_n$  là các hàm cơ sở của không gian tuyến tính các hàm  $d$  biến liên tục [22]. Ký hiệu

$$F = \text{span} \{u_1, u_2, \dots, u_n\} = \left\{ \sum_{k=1}^n c_k u_k, c_k \in \mathbb{R} \right\}.$$

Bài toán nội suy: Tìm hàm  $P_f \in F$  thỏa mãn

$$P_f(x_i) = y_i, \quad i = 1, 2, \dots, n. \quad (1.13)$$

Vì  $P_f \in F$  nên

$$P_f(x) = \sum_{k=1}^n c_k u_k(x), \quad x \in \mathbb{R}^d, \quad (1.14)$$

từ (1.13) và (1.14) ta có

$$Ac = y, \quad (1.15)$$

trong đó

$$A = \begin{pmatrix} u_1(x_1) & \cdots & u_n(x_1) \\ \cdots & \cdots & \cdots \\ u_1(x_n) & \cdots & u_n(x_n) \end{pmatrix}, \quad (1.16)$$

$$c = [c_1, \dots, c_n]^T, \quad y = [y_1, \dots, y_n]^T.$$

Phương trình (1.15) có nghiệm duy nhất khi  $\det(A) \neq 0$ , câu hỏi đặt ra là chọn cơ sở  $\{u_1, u_2, \dots, u_n\}$  như thế nào để điều kiện trên được thỏa mãn? Trong trường hợp  $d = 1$  thì ta có thể chọn cơ sở là

$$\{u_1, u_2, \dots, u_n\} = \{1, x, x^2, \dots, x^{n-1}\}.$$

**Định nghĩa 1.2.1.** (Không gian Haar) Giả sử  $\Omega \subset \mathbb{R}^d$  chứa ít nhất  $n$  điểm và  $F \subset C(\Omega)$  là không gian tuyến tính  $n$  chiều. Ta nói  $F$  là không gian Haar  $n$  chiều trên  $\Omega$  nếu với bất kỳ các điểm phân biệt  $x_1, x_2, \dots, x_n \in \Omega$  và  $f_1, f_2, \dots, f_n \in \mathbb{R}$ , thì tồn tại hàm  $s \in V$  sao cho  $s(x_i) = f_i, i = 1, 2, \dots, n$  [22].

**Định lý 1.2.1.**  $V$  là không gian Haar  $n$  chiều khi và chỉ khi với bất kỳ các điểm phân biệt  $x_1, x_2, \dots, x_n \in \Omega$  và cơ sở  $u_1, u_2, \dots, u_n$  của  $V$ , ta luôn có  $\det(A) \neq 0$ , trong đó ma trận  $A$  được xác định bởi (1.16) [22].

Sự tồn tại của không gian Haar đảm bảo tính khả nghịch của ma trận nội suy, nghĩa là tồn tại duy nhất nghiệm của Bài toán nội suy (1.13). Không gian các đa thức một biến bậc  $n - 1$  chính là không gian Haar  $n$  chiều với tập dữ liệu  $(x_j; y_j), x_j, y_j \in \mathbb{R}, j = 1, \dots, n$ .

**Định lý 1.2.2.** (Mairhuber Curtis) Giả sử rằng  $\Omega \subset \mathbb{R}^d, d \geq 2$ , chứa một điểm trong. Khi đó không tồn tại không gian Haar của các hàm liên tục trên  $\Omega$  [22].

Định lý Mairhuber Curtis cho thấy rằng nếu muốn giải được bài toán nội suy dữ liệu phân tán nhiều biến thì cơ sở cần phụ thuộc vào các vị trí dữ liệu. Để thu được các không gian xấp xỉ phụ thuộc dữ liệu, chúng ta cần xét các hàm xác định dương và ma trận xác định dương.

### 1.3. Hàm cơ sở bán kính

**Định nghĩa 1.3.1.** Một hàm  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$  được gọi là hàm cơ sở bán kính (RBF) nếu ở đó tồn tại một hàm  $\varphi : [0, +\infty) \rightarrow \mathbb{R}$  sao cho

$$\phi(x) = \varphi(\|x\|_2),$$

trong đó  $\|x\|_2$  là chuẩn Euclid [22].

Khi nhân  $r$  với một số lớn hơn không thì hàm  $\phi(x)$  vẫn là xác định dương, nên khi thêm tham số hình dạng  $\delta > 0$  vào hàm  $\phi$  ta có Bảng 1.2 sau:

Bảng 1.1: Một số hàm cơ sở bán kính, trong đó  $r = \|x - x_k\|$ .

Tên hàm	Viết tắt	Định nghĩa
Multiquadric	MQ	$\phi_{mq}(r) = \sqrt{1 + r^2}$
Inverse multiquadric	IMQ	$\phi_{imq}(r) = 1/\sqrt{1 + r^2}$
Gaussian	Gauss	$\phi_g(r) = e^{-r^2}$

Bảng 1.2: Một số hàm cơ sở bán kính với tham số hình dạng  $\delta > 0$ .

Tên hàm	Viết tắt	Định nghĩa
Multiquadric	MQ	$\phi_{mq}(r) = \sqrt{\delta^2 + r^2}$
Inverse multiquadric	IMQ	$\phi_{imq}(r) = 1/\sqrt{\delta^2 + r^2}$
Gaussian	Gauss	$\phi_g(r) = e^{-(r/\delta)^2}$

## 1.4. Ma trận xác định dương, hàm xác định dương

### Ma trận xác định dương:

**Định nghĩa 1.4.1.** Ma trận thực, đối xứng  $A = [A_{ij}]_{n \times n}$  được gọi là xác định dương nếu dạng toàn phương tương ứng không âm, tức là

$$\sum_{i=1}^n \sum_{j=1}^n c_j c_k A_{ij} \geq 0, \quad c = (c_1, c_2, \dots, c_n)^T \in \mathbb{R}^n$$

hay

$$c^T A c \geq 0 \quad c = (c_1, c_2, \dots, c_n)^T \in \mathbb{R}^n.$$

Dấu "=" xảy ra khi và chỉ khi  $c = (0, 0, \dots, 0)^T$  [22].

Tính chất quan trọng của ma trận xác định dương là các giá trị riêng của nó dương, do đó ma trận xác định dương là không suy biến.

Với cơ sở  $u_k$ , nếu Bài toán nội suy (1.13) tạo ra ma trận nội suy A xác định dương thì hệ (1.15) có nghiệm duy nhất.

### Hàm xác định dương:

**Định nghĩa 1.4.2.** Hàm  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$  liên tục, được gọi là xác định dương trên  $\mathbb{R}^d$  nếu và chỉ nếu nó là hàm chẵn và với mọi bộ tâm phân biệt từng đôi một  $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$ , với

mọi véc tơ  $c = (c_1, c_2, \dots, c_n) \in \mathbb{R}^n$  thì dạng toàn phương

$$\sum_{j=1}^n \sum_{k=1}^n c_j c_k \Phi(x_j - x_k) \geq 0. \quad (1.17)$$

Dấu "=" xảy ra khi và chỉ khi  $c$  là véc tơ 0 [22, 13].

**Định nghĩa 1.4.3.** Hàm một biến  $\varphi : [0, +\infty) \rightarrow \mathbb{R}$  được gọi là xác định dương trên  $\mathbb{R}^d$  nếu hàm nhiều biến tương ứng  $\Phi(x) = \varphi(\|x\|)$ ,  $x \in \mathbb{R}^d$ , là xác định dương [22, 13].

Từ định nghĩa trên và tính chất của ma trận xác định dương ta thấy có thể sử dụng các hàm xác định dương  $u_n = \Phi(x - x_k)$  là hàm cơ sở bán kính RBF và khi đó ta có:

$$P_f(x) = \sum_{k=1}^n c_k \Phi(x - x_k).$$

Ma trận nội suy  $A = [A_{ij}]_{n \times n}$ , với  $A_{ij} = u_k(x_i) = \Phi(x_i - x_j)$ ;  $i, j = 1, \dots, n$ .

**Định nghĩa 1.4.4.** Một hàm được gọi là hàm bán kính xác định dương nếu nó vừa là hàm bán kính vừa đồng thời xác định dương [22, 13].

Giả sử  $\Phi(x)$  là hàm xác định dương và được xác định

$$\Phi_k(x) = \Phi(x - x_k) = \varphi(\|x - x_k\|), \quad k = 1, 2, \dots, n, \quad x \in \mathbb{R}^d.$$

Khi đó ma trận của bài toán nội suy theo hàm  $\Phi(x)$  có dạng

$$A = \begin{pmatrix} \Phi(0) & \cdots & \Phi(x_1 - x_n) \\ \cdots & \cdots & \cdots \\ \Phi(x_n - x_1) & \cdots & \Phi(0) \end{pmatrix}.$$

Theo định nghĩa hàm xác định dương thì  $\det(A) \neq 0$ .

## 1.5. Nội suy hàm cơ sở bán kính

Giả sử có bộ tâm  $\{x_k, k = 1, 2, \dots, n\}$  và hàm  $\Phi(x)$  sao cho

$$\Phi_k(x) = \Phi(x - x_k) = \varphi(\|x - x_k\|), \quad k = 1, 2, \dots, n, \quad x \in \mathbb{R}^d.$$

Khi đó, nội suy hàm số dựa trên các hàm cơ sở bán kính là tìm hàm

$$P_f(x) = \sum_{k=1}^n c_k \Phi_k(x) = \sum_{k=1}^n c_k \varphi(\|x - x_k\|)$$

thỏa mãn điều kiện nội suy (1.13).

Nếu  $\Phi_k(x)$  là hàm xác định dương thì theo điều kiện nội suy ta có

$$P_f(x_i) = y_i, \quad i = 1, 2, \dots, n$$

hay

$$\sum_{k=1}^n c_k \varphi(\|x_i - x_k\|) = y_i, \quad i = 1, 2, \dots, n$$

hay

$$Ac = y,$$

trong đó

$$A = \begin{pmatrix} \Phi(0) & \Phi(x_1 - x_2) & \cdots & \Phi(x_1 - x_n) \\ \Phi(x_2 - x_1) & \Phi(0) & \cdots & \Phi(x_2 - x_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Phi(x_n - x_1) & \Phi(x_n - x_2) & \cdots & \Phi(0) \end{pmatrix},$$

$$c = [c_1, \dots, c_n]^T, \quad y = [y_1, \dots, y_n]^T.$$

Theo định nghĩa hàm xác định dương ta có  $\det(A) \neq 0$ .

## 1.6. Một số kiến thức cơ bản về lý thuyết xử lý song song

### Định nghĩa:

**Định nghĩa 1.6.1.** *Xử lý song song là quá trình xử lý gồm nhiều tiến trình được kích hoạt đồng thời và cùng tham gia giải quyết một vấn đề, nói chung là thực hiện trên những đa bộ xử lý [1].*

Một trong các mục đích chính của xử lý song song là nghiên cứu và xây dựng những thuật toán thích hợp để cài đặt trên các máy tính song song, nghĩa là phát triển các thuật toán song song. Đối với thuật toán tuần tự thì chúng ta có thể thống nhất cách đánh giá dựa vào thời gian thực hiện thuật toán, không gian bộ nhớ và khả năng lập trình. Đánh giá thuật toán song song thì phức tạp hơn nhiều, ngoài những tiêu chuẩn trên còn phải bổ sung thêm những tham số về số bộ xử lý, khả năng của các bộ nhớ cục bộ, sơ đồ truyền thông và các giao thức đồng bộ hoá, v.v.

Để cài đặt các thuật toán song song trên các máy tính song song chúng ta phải sử dụng những ngôn ngữ lập trình song song. Nhiều ngôn ngữ lập trình song song đang được sử dụng như: Fortran 90, nCUBE C, Occam, C-Linda, PVM với C/C++, CDC 6600, v.v.

## Thuật toán song song:

Thuật toán song song là một tập các tiến trình hoặc các tác vụ có thể thực hiện đồng thời và có thể trao đổi dữ liệu với nhau để kết hợp cùng giải một bài toán đặt ra. Thuật toán song song có thể xem như là một tập hợp các đơn thể độc lập, một số trong số chúng có thể thực hiện tương tranh trên máy tính song song [1]. Để thiết kế được các thuật toán song song cần phải thực hiện.

- Phân chia dữ liệu cho các tác vụ.
- Chỉ ra cách truy cập và chia sẻ dữ liệu.
- Phân các tác vụ cho các tiến trình (bộ xử lý).
- Các tiến trình được đồng bộ ra sao.

Khi thiết kế một thuật toán song song có thể sử dụng một số nguyên lí chính trong thiết kế thuật toán song song

- *Nguyên lí lập lịch*: Sử dụng các thuật toán lập lịch cho các bộ xử lý để giảm tối thiểu thời gian tính toán.
- *Nguyên lí hình ống*: Nguyên lí này được áp dụng khi bài toán xuất hiện một dãy các thao tác  $\{T_1, T_2, \dots, T_n\}$ , trong đó  $T_{i+1}$  thực hiện sau khi  $T_i$  kết thúc.
- *Nguyên lí chia để trị*: Chia bài toán thành những phần nhỏ hơn tương đối độc lập với nhau và giải quyết chúng một cách song song.

Có ba cách tiếp cận để thiết kế thuật toán song song:

1. Song song hoá những thuật toán tuần tự, biến đổi những cấu trúc tuần tự để tận dụng khả năng song song tự nhiên của tất cả các thành phần trong hệ thống xử lý.
2. Thiết kế thuật toán song song mới trên cơ sở thuật toán song song đã có.
3. Thiết kế thuật toán song song hoàn toàn mới thích ứng với những cấu trúc song song.

Như vậy, cách làm thông dụng là biến đổi các thuật toán tuần tự về song song, hay chuyển từ một dạng song song về dạng song song phù hợp hơn sao vẫn bảo toàn được tính tương đương trong tính toán. Do đó, khi biến đổi chúng ta cần trả lời hai câu hỏi

- Kiến trúc nào phù hợp cho bài toán?
- Những bài toán loại nào sẽ xử lý hiệu quả trong kiến trúc song song cho trước?

### Phân tích và đánh giá thuật toán song song:

Để đánh giá được độ phức tạp tính toán của các thuật toán song song, ngoài số bước tính toán chúng ta còn cần đánh giá thời gian truyền thông của các tiến trình [19]. Trong một hệ thống truyền thông, thời gian truyền thông cũng phải được xem trong thời gian thực hiện của thuật toán. Thời gian thực hiện song song, ký hiệu là  $t_p$  gồm hai phần  $t_{comp}$  và  $t_{comm}$

$$t_p := t_{comp} + t_{comm},$$

trong đó,  $t_{comp}$  là thời gian tính toán và  $t_{comm}$  - thời gian truyền thông dữ liệu.

Thời gian tính toán  $t_{comp}$  được xác định giống như thuật toán tuần tự. Khi có nhiều tiến trình tiến trình thực hiện đồng thời thì tính thời gian thực hiện của tiến trình phức tạp nhất (thực hiện lâu nhất). Trong phân tích độ phức tạp tính toán, chúng ta luôn giả thiết rằng, tất cả các bộ xử lý là giống nhau và thao tác cùng một tốc độ như nhau. Đối với những cụm máy tính không thuần nhất thì điều này không đảm bảo do vậy, việc đánh giá thời gian tính toán của những hệ như thế là rất phức tạp.

Thời gian truyền thông  $t_{comm}$  lại phụ thuộc vào kích cỡ của các thông điệp, vào cấu hình kết nối mạng đường truyền và cả cách thức truyền tải thông điệp, v.v. Công thức ước lượng thời gian truyền thông được xác định như sau

$$t_{comm} := t_{startup} + n \times t_{data},$$

trong đó  $t_{startup}$  là thời gian cần thiết để gửi những thông điệp không phải là dữ liệu. Nó bao gồm cả thời gian để đóng gói thông điệp ở nơi gửi và thời gian mở gói ở nơi nhận. Để đơn giản chúng ta giả thiết thời gian này là hằng số.  $t_{data}$  là thời gian cần thiết để chuyển một từ dữ liệu (một mục dữ liệu) từ nơi gửi tới nơi nhận, được giả thiết là hằng số và  $n$  là số từ dữ liệu được trao đổi trong hệ thống.

Đánh giá thuật toán tuần tự có thể căn cứ chủ yếu vào thời gian thực hiện được tính theo hàm của kích cỡ dữ liệu vào (input). Hàm này được gọi là độ phức tạp tính toán thời gian  $f(n)$  của thuật toán và được ký hiệu là  $\mathcal{O}(f(n))$  và  $\mathcal{O}(1)$  ký hiệu cho một hằng số bất kỳ. Một cách hình thức  $\mathcal{O}(g(x))$  được định nghĩa như sau:

**Định nghĩa 1.6.2.** Cho  $g(x), f(x)$  là hai hàm có giá trị dương. Ta nói một thuật toán có độ phức tạp tính toán  $f(x) = \mathcal{O}(g(x))$  khi và chỉ khi tồn tại số  $C$  dương và số nguyên  $x_0$  sao cho  $0 \leq f(x) \leq C \times g(x)$ , với mọi số lượng dữ liệu vào  $x \geq x_0$ .

Độ phức tạp tính toán của thuật toán song song không chỉ phụ thuộc vào kích cỡ của dữ liệu đầu vào mà còn phụ thuộc vào kiến trúc máy tính song song và số lượng các bộ xử lý được phép sử dụng trong hệ thống.

Độ phức tạp thời gian là thước đo quan trọng nhất đánh giá mức độ hiệu quả của thuật toán song song. Chúng ta giả thiết rằng mô hình tính toán có  $p$  bộ xử lý. Nghĩa là mức độ song song là có giới hạn. Ngược lại, mức độ song song không bị giới hạn khi số các bộ xử lý là không bị chặn.

Độ phức tạp thời gian của thuật toán song song sử dụng  $p$  bộ xử lý để giải một bài toán có kích cỡ  $n$  là hàm  $f(n, p)$  xác định thời gian cực đại trôi qua giữa thời điểm bắt đầu thực hiện thuật toán bởi một bộ xử lý và thời điểm kết thúc của các bộ xử lý đối với bộ dữ liệu vào bất kỳ. Có hai loại thao tác khác nhau trong các thuật toán song song

- Các phép toán cơ sở như  $+$ ,  $-$ ,  $\times$ ,  $/$ , AND, OR, v.v.
- Các phép toán truyền dữ liệu trên các kênh truyền.

Độ phức tạp thời gian của thuật toán song song được xác định bởi số các phép toán cơ sở và số các bước truyền tải dữ liệu giữa các bộ xử lý với nhau. Từ đó suy ra, độ phức tạp thời gian của thuật toán song song không chỉ phụ thuộc vào mô hình tính toán mà còn phụ thuộc vào số bộ xử lý được sử dụng.

Nói chung, chương trình tính toán song song thường bắt đầu bằng việc nhập dữ liệu vào bộ nhớ và kích hoạt một phần tử xử lý. Mỗi bước tính toán, phần tử xử lý này có thể đọc một số dữ liệu từ bộ nhớ, thực hiện một số phép toán cơ sở và ghi kết quả vào bộ nhớ riêng hoặc bộ nhớ chung. Đồng thời mỗi bước tính toán, một phần tử xử lý có thể kích hoạt một hay một số phần tử xử lý khác. Thực tế thì các máy tính đều có số bộ xử lý là hữu hạn, nên những thuật toán song song không bị giới hạn chỉ có nghĩa sử dụng khi chúng có thể chuyển đổi về thuật toán song song bị giới hạn.

Có ba cách định nghĩa khái niệm liên quan đến độ phức tạp của thuật toán song song



- Một thuật toán song song có độ phức tạp tính toán  $\mathcal{O}(T)$  với  $P$  bộ xử lý khi nó thực hiện nhiều nhất là  $\mathcal{O}(T \times P)$  phép toán cơ sở (định lý Brent).
- Một thuật toán song song có độ phức tạp tính toán  $\mathcal{O}(T)$  sử dụng rất nhiều bộ xử lý để thực hiện  $\mathcal{O}(e)$  phép toán cơ sở khi cài đặt với  $P$  bộ xử lý thì sẽ có độ phức tạp thời gian là  $\mathcal{O}\left(\left\lceil \frac{e}{P} \right\rceil + T\right)$ .
- Một thuật toán song song có độ phức tạp tính toán  $\mathcal{O}(T)$  với  $P$  bộ xử lý có thể cài đặt với  $\left\lceil \frac{P}{p} \right\rceil$ ,  $1 \leq p \leq P$  bộ xử lý thì sẽ có độ phức tạp thời gian là  $\mathcal{O}(p \times T)$ .

Kết quả thứ 2 chỉ ra rằng khi số bộ xử lý được sử dụng giảm xuống trong một phạm vi nhất định thì thuật toán tiếp tục làm việc nhưng thời gian thực hiện sẽ tăng lên. Kết quả thứ 3 khẳng định rằng có cách để cài đặt thuật toán song song khi số các bộ xử lý được sử dụng bị giảm xuống.

Mức độ hiệu quả của thuật toán được thể hiện ở mức độ song song của thuật toán. Mức độ song song của thuật toán là số lượng cực đại các phép toán độc lập có thể thực hiện đồng thời ở mỗi thời điểm thực hiện của thuật toán. Ký hiệu  $P(W)$  là độ song song của thuật toán, thì thuật toán hiệu quả giải để giải bài toán có cỡ  $W$  là những thuật toán chỉ cần sử dụng nhiều nhất  $P(W)$  bộ xử lý. Ngoài ra, để đánh giá được thuật toán song song chúng ta còn phải xét tới hệ số gia tốc của nó. Hệ số gia tốc của thuật toán song song sử dụng  $p$  bộ xử lý được xác định bởi công thức

$$S_p = \frac{t_S}{t_p}, \quad 1 \leq S_p \leq p,$$

trong đó  $t_S$  là thời gian thực hiện tính toán trên một bộ xử lý và  $t_p$  là thời gian thực hiện tính toán trên  $p$  bộ xử lý, với giả thiết là bộ xử lý tuần tự và bộ xử lý song song là như nhau.

## Chương 2

# Tính toán song song cho phương pháp RBF-FD trong không gian hai chiều

Trong chương này chúng tôi sẽ trình bày các phần sau: 2.1 Giới thiệu về véc tơ trọng số trong công thức vi phân số; 2.2 Cách rời rạc bài toán truyền nhiệt dừng; 2.3 Thuật toán tuần tự chọn bộ tâm hỗ trợ; 2.4 Song song hóa việc chọn bộ tâm hỗ trợ và tính véc tơ trọng số; 2.5 Thử nghiệm số và cuối cùng là phần Kết luận.

Các kết quả trong chương này được công bố trong [3, 4, 5].

### 2.1. Véc tơ trọng số trong công thức vi phân số

Giả sử  $D$  là toán tử vi phân tuyến tính và  $X = \{x_i\}_{i=0}^n$  là bộ tâm phân bố không đều trong  $\mathbb{R}^d$ . Ta sẽ tìm công thức vi phân số của toán tử  $D$

$$Du(x) \approx \sum_{i=0}^n w_i(x)u(x_i), \quad (2.1)$$

được xác định bởi các trọng số  $w_i = w_i(x)$ . Véc tơ  $w = [w_0, \dots, w_n]^T$  được gọi là *stencil* hay còn gọi là *véc tơ trọng số*.

Thật vậy, véc tơ trọng số trong công thức (2.1) có thể được tìm bằng nội suy dữ liệu phân tán như sau:

Cho  $s$  là một xấp xỉ của  $u$  trên bộ tâm phân tán  $X = \{x_0, \dots, x_n\} \subset \mathbb{R}^d$  với các giá trị tương ứng  $(x_i, u(x_i))$  có dạng

$$s = \sum_{i=0}^m a_i s_i,$$

trong đó  $s_i, i = 0, \dots, m$  là các hàm cơ sở và véc tơ hệ số  $a = [a_0, \dots, a_m]^T$  phụ thuộc tuyến tính vào  $u|_X$ ,

$$a_i = \sum_{j=0}^n b_{ij}u(x_j), \quad i = 0, \dots, m,$$

hay

$$a = B \cdot u|_X, \quad (2.2)$$

trong đó  $B = [b_{ij}]_{i=0, j=0}^{m, n}$ .

Để tìm công thức vi phân của toán tử  $D$ , ta xấp xỉ

$$Du(x) \approx Ds(x) = \sum_{i=0}^m a_i Ds_i(x) = \sum_{j=0}^n \left( \sum_{i=0}^m b_{ij} Ds_i(x) \right) u(x_j) = \sum_{j=0}^n w_j u(x_j),$$

và ta nhận được công thức (2.1) với stencil  $w = [w_0, \dots, w_n]^T$  được cho bởi

$$w_j = \sum_{i=0}^m b_{ij} Ds_i(x), \quad j = 0, \dots, n,$$

hay

$$w = B^T \cdot [Ds_i(x)]_{i=0}^m. \quad (2.3)$$

Xét trường hợp  $m = n$  và các hệ số  $a_j$  thu được bằng cách giải bài toán nội suy không suy biến

$$s(x_i) = \sum_{j=0}^n a_j s_j(x_i) = u(x_i), \quad i = 0, \dots, n,$$

hay

$$S_X a = u|_X,$$

hay

$$a = S_X^{-1} u|_X,$$

trong đó  $S_X := [s_j(x_i)]_{i, j=0}^n$ . Từ (2.2), (2.3) suy ra  $B = S_X^{-1}$  và véc tơ trọng số  $w$  được tính bởi công thức

$$w = S_X^{-T} \cdot [Ds_i(x)]_{i=0}^n,$$

hay giải hệ phương trình tuyến tính

$$S_X^T w = [Ds_i(x)]_{i=0}^n, \quad (2.4)$$

hay

$$\sum_{i=0}^n w_i s_j(x_i) = Ds_j(x), \quad j = 0, \dots, n.$$

Độ chính xác của véc tơ trọng số liên quan trực tiếp đến độ chính xác của phương pháp xấp xỉ.

Trong phương pháp không lưới RBF-FD, véc tơ trọng số  $w$  có thể được tìm bằng nội suy Hermite (xem chi tiết trong [5, 10]) với cách tiếp cận toàn cục, tuy nhiên khi giải bài toán bằng các tiếp cận này dẫn đến số điều kiện của ma trận tăng lên khi số tâm nội suy tăng hoặc

miền có hình học phức tạp. Vì vậy, véc tơ  $w$  chủ yếu được tìm bằng sử dụng nội suy hàm cơ sở bán kính RBF [3, 4, 10, 18], với hàm nội suy cơ sở bán kính  $s$  của hàm  $u$  có dạng

$$s(x) = \sum_{j=0}^n a_j \Phi(x - x_j), \quad \Phi(x) := \varphi(\|x\|), x \in \mathbb{R}^d,$$

trong đó  $\varphi : \mathbb{R}_+ \rightarrow \mathbb{R}$  là hàm xác định dương và  $\|x\|$  là chuẩn Euclid. Khi đó véc tơ trọng số  $w$  trong công thức (2.4) được tìm bởi công thức

$$\Phi_X w = D\Phi(x_0 - \cdot)|_X, \quad (2.5)$$

với  $\Phi_X := [\Phi(x_i - x_j)]_{i,j=0}^n, D\Phi(x_0 - \cdot)|_X := [Ds(x_0 - x_i)]_{i=0}^n$ .

Khi tìm véc tơ trọng số trong các thử nghiệm số, các tác giả đã dùng hàm RBF Gauss  $\varphi(r) = e^{-(cr)^2}$ , đây là hàm xác định dương đối với mọi giá trị *tham số hình dạng*  $c > 0$  (xem chi tiết [11]). Với hàm này, ma trận  $\Phi_X$  có dạng

$$\Phi_X = [e^{-c^2\|x_i - x_j\|^2}]_{i,j=1}^n.$$

Khi đó Laplacian của hàm Gauss  $\Phi(x) = e^{-c^2\|x\|^2}$  là

$$\Delta\Phi(x) = 2c^2 e^{-c^2\|x\|^2} (2c^2\|x\|^2 - d).$$

Trong phần tiếp theo, ta sử dụng công thức vi phân số (2.1) để rời rạc bài toán truyền nhiệt dừng (1.6)-(1.7), trong đó toán tử vi phân  $D$  là toán tử Laplace  $\Delta$ , xem thêm trong [4, 10, 18].

## 2.2. Rời rạc bài toán truyền nhiệt dừng

Bài toán truyền nhiệt dừng (1.6)-(1.7) được viết dưới dạng bài toán Dirichlet đối với phương trình Poisson trong không gian hai chiều là:

Cho miền mở  $\Omega \subset \mathbb{R}^2$  và các hàm số  $f$  xác định trên  $\Omega$ ,  $g$  được xác định trên  $\partial\Omega$ . Tìm  $u : \bar{\Omega} \rightarrow \mathbb{R}$  sao cho

$$\Delta u = f \text{ trên } \Omega, \quad (2.6)$$

$$u|_{\partial\Omega} = g. \quad (2.7)$$

Bài toán được rời rạc hóa như sau:

Gọi  $\Xi \subset \bar{\Omega}$  là bộ hữu hạn các tâm rời rạc,  $\partial\Xi := \Xi \cap \partial\Omega$  và  $\Xi_{\text{int}} := \Xi \setminus \partial\Xi$ . Giả sử với mỗi  $\zeta \in \Xi_{\text{int}}$ , bộ  $\Xi_\zeta \subset \Xi$  được chọn sao cho  $\zeta \in \Xi_\zeta$  và

$$\Xi = \bigcup_{\zeta \in \Xi_{\text{int}}} \Xi_\zeta.$$

Đối với mỗi  $\zeta \in \Xi_{\text{int}}$ , sử dụng công thức (2.1) đối với toán tử Laplace  $\Delta$  ta được

$$\Delta u(\zeta) \approx \sum_{\xi \in \Xi_\zeta} w_{\zeta, \xi} u(\xi), \quad (2.8)$$

trong đó  $[w_{\zeta, \xi}]_{\xi \in \Xi_\zeta}$  là các véc tơ trọng số. Khi đó bài toán (2.6)–(2.7) được rời rạc thành hệ phương trình tuyến tính

$$\sum_{\xi \in \Xi_\zeta} w_{\zeta, \xi} \hat{u}(\xi) = f(\zeta), \quad \zeta \in \Xi_{\text{int}}, \quad (2.9)$$

$$\hat{u}(\xi) = g(\xi), \quad \xi \in \partial\Xi. \quad (2.10)$$

Nếu hệ phương trình (2.9)–(2.10) không suy biến thì giải hệ ta sẽ tìm được nghiệm xấp xỉ của bài toán là  $\hat{u} : \Xi \rightarrow \mathbb{R}$ .

Khi thực hiện trên miền hình vuông  $\Omega \subset \mathbb{R}^2$ , phương pháp được trình bày này là phương pháp sai phân hữu hạn,  $\Xi$  là lưới đều và (2.8) là công thức sai phân 5 - điểm đối với toán tử Laplace.

Độ chính xác của phương pháp khi giải bài toán (2.9)–(2.10) phụ thuộc vào cách chọn bộ tâm hỗ trợ tính toán véc tơ trọng số  $\Xi_\zeta$ . Hiện nay có nhiều thuật toán chọn  $\Xi_\zeta$ , xem [10, 18, 20, 23]. Trong phần tiếp theo, chúng tôi xin giới thiệu thuật toán chọn tâm [18, Thuật toán 1] được đề xuất gần đây nhất và đề xuất song song hóa thuật toán này cùng với tính véc tơ trọng số trong [3].

### 2.3. Thuật toán chọn tâm

#### Ý tưởng của thuật toán:

Với mỗi  $\zeta \in \Xi_{\text{int}}$ , ta chọn được tập  $\Xi_\zeta = \{\zeta, \zeta_1, \dots, \zeta_n\} \subset \Xi$ , trong đó các điểm  $\zeta_1, \dots, \zeta_n$  được xếp theo thứ tự ngược chiều kim đồng hồ đối với  $\zeta$ . Đặt

$$\mu(\zeta_1, \dots, \zeta_n) := \sum_{i=1}^n \alpha_i^2, \quad \underline{\alpha}(\zeta_1, \dots, \zeta_n) := \min_{1 \leq i \leq n} \alpha_i, \quad \bar{\alpha}(\zeta_1, \dots, \zeta_n) := \max_{1 \leq i \leq n} \alpha_i,$$

trong đó  $\alpha_i$  là góc giữa các tia  $\zeta_{\zeta_i}, \zeta_{\zeta_{i+1}}$  theo hướng ngược chiều kim đồng hồ, với chu trình  $\zeta_{n+i} := \zeta_i$ . Vì  $\sum_{i=1}^n \alpha_i = 2\pi$ , nên biểu thức  $\sum_{i=1}^n \alpha_i^2$  đạt giá trị nhỏ nhất khi các tia  $\zeta_{\zeta_i}$  phân bố đều hay các góc  $\alpha_i$  bằng nhau  $\alpha_1 = \dots = \alpha_n = 2\pi/n$ . Trong thực nghiệm rất khó để chọn được các điểm  $\zeta_1, \dots, \zeta_n$  sao cho biểu thức  $\sum_{i=1}^n \alpha_i^2$  đạt giá trị nhỏ nhất đối với tập con  $\{\zeta_1, \dots, \zeta_n\}$  của tập hữu hạn  $\Xi$ , nên trong [10, 18] các tác giả sử dụng  $\mu$  làm độ đo các góc đều khi so sánh 2 tập con cùng dạng và các thuật toán sử dụng điều kiện dừng

$$\bar{\alpha}(\zeta_1, \dots, \zeta_n) \leq v \underline{\alpha}(\zeta_1, \dots, \zeta_n),$$

với  $v > 1.0$ .

Các thuật toán đều bắt đầu với  $k$  điểm gần nhất, sau đó thay thế điểm  $\zeta_j$  bởi các điểm có khoảng cách lớn hơn  $\xi \in \Xi \setminus \Xi_{\zeta}$  sao cho  $\mu(\zeta_1, \dots, \zeta_k)$  nhỏ hơn. Mặt khác, vì các điểm ở gần cũng phù hợp, nên thuật toán chọn tâm trong [18] sử dụng thêm điều kiện dừng thứ hai

$$\|\zeta - \xi\| \geq \frac{c}{2k} \sum_{j=1}^k (\|\zeta_j - \zeta\| + \|\zeta_j - \zeta_{j+1}\|),$$

với hệ số  $c > 1.0$  và  $\xi \in \Xi \setminus \Xi_{\zeta}$ .

Thử nghiệm số cho thấy, các giá trị  $v = 2.5$ ,  $c = 3.0$  được xác định là cho kết quả tốt và luôn chọn  $k = 6$  để đảm bảo mật độ của ma trận hệ số (2.10) có thể so sánh với mật độ của ma trận hệ số của phương pháp phần tử hữu hạn với hàm tuyến tính, xem chi tiết trong [10].

Để tính toán nhanh, ban đầu các thuật toán chọn  $m = 50$  điểm gần nhất và nhân đôi kích thước tập được chọn khi quá trình thay thế đã hết.

## Nội dung của thuật toán:

**Algorithm 1.** [Thuật toán 1, [18]] Thuật toán ODP chọn tập các tâm hỗ trợ

*Input:*  $\Xi, \zeta$ .

*Output:*  $\Xi_{\zeta}$ .

*Các tham số:*  $k$  (số điểm trong  $\Xi_{\zeta} \setminus \{\zeta\}$ ),  $v > 1.0$  (tham số góc đều),  $c > 1.0$  (khoảng cách), và  $m > k$ . Các giá trị tham số được sử dụng trong thử nghiệm số  $k = 6$ ,  $v = 2.5$ ,  $c = 3.0$  và  $m = 50$ .

- I. Tìm  $m$  điểm gần nhất  $\xi_1, \dots, \xi_m$  trong  $\Xi \setminus \{\zeta\}$  đến  $\zeta$ , sắp xếp theo chiều tăng dần về mặt khoảng cách đến  $\zeta$ , tập ban đầu  $\Xi_{\zeta} := \{\zeta, \zeta_1, \dots, \zeta_k\} = \{\zeta, \xi_1, \dots, \xi_k\}$  và  $i := k + 1$ .

II. *While*  $i \leq m$ :

1. Nếu  $\|\zeta - \xi_i\| \geq \frac{c}{2k} \sum_{j=1}^k (\|\zeta_j - \zeta\| + \|\zeta_j - \zeta_{j+1}\|)$ , thì STOP và trả về  $\Xi_\zeta$ .
2. Tính các góc  $\alpha'_1, \dots, \alpha'_{k+1}$  được tạo bởi tập mở rộng  $\{\zeta'_1, \dots, \zeta'_{k+1}\} = \{\zeta_1, \dots, \zeta_k, \xi_i\}$ .  
Nếu các góc giữa tia  $\zeta \xi_i$  và hai tia lân cận của nó lớn hơn góc nhỏ nhất  $\underline{\alpha}' := \underline{\alpha}(\zeta'_1, \dots, \zeta'_{k+1})$ :
  - a. Tìm  $j$  sao cho  $\alpha'_j = \underline{\alpha}'$ . Chọn  $p = j$  hoặc  $p = j + 1$  phụ thuộc vào  $\alpha'_{j-1} < \alpha'_{j+1}$  hoặc  $\alpha'_{j-1} \geq \alpha'_{j+1}$ .
  - b. Nếu  $\mu(\{\zeta'_1, \dots, \zeta'_{k+1}\} \setminus \{\zeta'_p\}) < \mu(\zeta_1, \dots, \zeta_k)$ :
    - i) Cập nhật  $\Xi_\zeta := \{\zeta, \zeta_1, \dots, \zeta_k\} = \{\zeta, \zeta'_1, \dots, \zeta'_{k+1}\} \setminus \{\zeta'_p\}$ .
    - ii) Nếu  $\bar{\alpha}(\zeta_1, \dots, \zeta_k) \leq v \underline{\alpha}(\zeta_1, \dots, \zeta_k)$ , thì STOP và trả về  $\Xi_\zeta$ .
3. Nếu  $i = m$ :

Tìm  $m$  điểm gần nhất tiếp theo  $\xi_{m+1}, \dots, \xi_{2m}$  trong  $\Xi \setminus \{\zeta\}$  đến  $\zeta$ , xếp theo chiều tăng dần về mặt khoảng cách đến  $\zeta$  và đặt  $m := 2m$ .

4. Đặt  $i := i + 1$ .

Khi giải bài toán (2.9)–(2.10), với mỗi  $\zeta \in \Xi_{\text{int}}$  sử dụng thuật toán ODP chọn được bộ tâm hỗ trợ tính toán véc tơ trọng số  $\Xi_\zeta$  và tính toán véc tơ trọng số  $w_\zeta$  tương ứng với bộ tâm này (xem chi tiết trong [18]). Công đoạn này hoàn toàn độc lập trên mỗi  $\zeta \in \Xi_{\text{int}}$  và rất tốn thời gian, đặc biệt với các bài toán có dữ liệu lớn, nên cần được song song hóa. Trong phần tiếp theo chúng tôi sẽ trình bày quá trình song song hóa công đoạn nói trên và các thử nghiệm số để thấy được sự hiệu quả của quá trình này (tham khảo [3]).

*Độ phức tạp của thuật toán 1:* Giả sử tập các tâm rời rạc  $\Xi$  có  $q$  điểm và tập  $\Xi_{\text{int}}$  có  $n$  điểm. Khi đó độ phức tạp tính toán của thuật toán 1 là  $\mathcal{O}(n \cdot m \cdot \log(q))$ .

Thật vậy, với mỗi tâm  $\zeta \in \Xi_{\text{int}}$ , chi phí tính toán đối với các bước trong thuật toán là:

I. Chi phí tính toán với bước I:

1. Thời gian tìm  $m$  điểm gần  $\zeta$  nhất là  $m \cdot \mathcal{O}(\log(q))$ .
2. Thời gian sắp xếp  $m$  điểm theo chiều tăng dần về mặt khoảng cách đến  $\zeta$  là  $\mathcal{O}(m^2)$ .
3. Thời gian xác định  $k$  điểm đầu tiên là  $\mathcal{O}(k)$ .

Áp dụng theo quy tắc cộng suy ra chi phí thời gian để xác định tập  $\Xi_\zeta$  ban đầu là  $\mathcal{O}(m \cdot \log(q))$ .

II. Chi phí để loại bỏ điểm "tồi" lấy điểm "tốt hơn" với  $m - k$  tâm còn lại như sau

1. Chi phí tính toán bước II.1 là  $(m - k) \cdot \mathcal{O}(k^2)$ .
2. Chi phí tính toán các góc được tạo bởi các tập mở rộng ở bước II.2 là  $\mathcal{O}(k + 1)$ .
  - a. Chi phí tính toán bước II.2.a là  $\mathcal{O}((k + 1)^2)$ .
  - b. Chi phí tính toán bước II.2.b là  $\mathcal{O}(1)$ .

Sử dụng quy tắc cộng ta có chi phí tính toán của bước II.2 là  $\mathcal{O}((m - k) \cdot \log(k + 1)^2)$ .

3. Chi phí tính toán bước II.3 là  $m \cdot \mathcal{O}(\log(k + 1)^2)$ .
4. Chi phí tính toán bước II.4 là  $\mathcal{O}(1)$ .

Vậy chi phí tính toán của thuật toán theo quy tắc cộng là  $\mathcal{O}(n \cdot m \cdot \log(q))$ .

## 2.4. Song song hóa việc chọn tâm và tính véc tơ trọng số

### Ý tưởng thuật toán

Giả sử có  $N$  bộ xử lý và  $\Xi \subset \bar{\Omega}$  là tập các tâm rời rạc hữu hạn,  $\Xi_{\text{int}}$  là tập các tâm nằm trong miền.

- Trước tiên, ta cần phân hoạch  $\Xi_{\text{int}}$  thành  $N$  phần xấp xỉ bằng nhau, tương ứng với  $N$  bộ xử lý;
- Tiếp theo, thực hiện thuật toán ODP trên mỗi bộ xử lý để tìm tập  $\Xi_\zeta$  và tính véc tơ trọng số  $w_\zeta$  tương ứng với  $\Xi_\zeta$ , đồng thời lưu trữ các véc tơ trọng số vừa tìm được vào tập  $w_\zeta$ .

Để thực hiện được tính toán song song khi sử dụng thuật toán ODP và tính véc tơ trọng số, ta cần phân luồng dữ liệu đầu vào phù hợp. Nghĩa là, tách và phân phối  $n$  tâm trong tập  $\Xi_{\text{int}}$  đều khắp trên  $N$  bộ xử lý, sao cho mỗi bộ xử lý có số tâm gần bằng nhau, tương ứng là

$$\Xi_{\text{int}}^{(i)} := \left\{ \zeta_j^{(i)} : j = 1, 2, \dots, n^{(i)} \right\}, \quad i = 1, 2, \dots, N$$



với

$$n^{(1)} \approx n^{(2)} \approx \dots \approx n^{(N)} \approx \left\lfloor \frac{n}{N} \right\rfloor.$$

Từ đó mỗi bộ xử lý sẽ tính số tập các tâm hỗ trợ  $\Xi_{\zeta_j}^{(i)}$ ,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, n^{(i)}$  và véc tơ trọng số  $w_{\zeta_j}^{(i)}$ ,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, n^{(i)}$  tương ứng. Quá trình xử lý song song như trong sau:

**Algorithm 2. [Thuật toán 2, [3]] Thuật toán tính toán song song**

*Input:* Bộ tâm rời rạc  $\Xi, \Xi_{\text{int}}, N$ .

*Output:* Tập các véc tơ trọng số  $w_{\zeta}, \zeta \in \Xi_{\text{int}}$ .

*Các tham số:* Các tham số của thuật toán ODP:  $k, v, c, m$ .

Khởi tạo  $w_{\zeta} := \emptyset$ .

I. Phân hoạch dữ liệu cho  $N$  bộ xử lý

1.  $n_1 := \left\lfloor \frac{n}{N} \right\rfloor; i := 1; j := 0$ .
2. While  $i \leq N$ 
  - a. If  $i = N$  then  $n^{(i)} := n_1 (1 - N) + n$   
Else  $n^{(i)} := n_1$ ;
  - b.  $\Xi_{\text{int}}^{(i)} := \left\{ \zeta_{j+1}, \zeta_{j+2}, \dots, \zeta_{j+n^{(i)}} \right\}$ ;
  - c.  $j := j + n^{(i)}$ ;
  - d.  $i := i + 1$ .

II. Đối với mỗi bộ xử lý thứ  $i \in N$

1. Với mỗi  $\zeta_j \in \Xi_{\text{int}}^{(i)}$ 
  - a. Sử dụng thuật toán chọn tâm ODP, tìm các tập  $\Xi_{\zeta_j}^{(i)}$ ,  $j = 1, 2, \dots, n^{(i)}$ ;
  - b. Tính các véc tơ trọng số  $w_{\zeta_j}^{(i)}$ ,  $j = 1, 2, \dots, n^{(i)}$  bởi công thức (2.5) tương ứng với các tập  $\Xi_{\zeta_j}^{(i)}$ ,  $j = 1, 2, \dots, n^{(i)}$ .
2. Lưu trữ các véc tơ trọng số vừa tính  $w_{\zeta} := w_{\zeta} \cup \left\{ w_{\zeta_j}^{(i)} : j = 1, 2, \dots, n^{(i)} \right\}$ .

Lưu đồ tính toán song song sử dụng thuật toán hỗ trợ chọn tâm ODP được biểu diễn trong hình (3.1).

Để tìm độ phức tạp tính toán của thuật toán 2 ta tính chi phí tính toán đối với các bước trong thuật toán như sau:



Hình 2.1: Lưu đồ song song hóa thuật toán hỗ trợ chọn tâm ODP và tính véc tơ trọng số.

I. Chi phí thời gian phân hoạch dữ liệu cho N bộ xử lý

1. Chi phí tính toán bước I.1 là  $\mathcal{O}(1)$ .
2. Chi phí thời gian tính toán bước I.2 là  $\frac{n}{N} \cdot \mathcal{O}(\log(n))$ 
  - a. Chi phí tính toán bước I.2.a là  $\mathcal{O}(1)$ .
  - b. Chi phí tính toán bước I.2.b là  $\frac{n}{N} \cdot \mathcal{O}(\log(n))$ .

c. Chi phí tính toán bước I.2.c là  $\mathcal{O}(1)$ .

d. Chi phí tính toán bước I.2.d là  $\mathcal{O}(1)$ .

Do đó chi phí tính toán phân hoạch  $n$  cho  $N$  bộ xử lý ở bước I.2 là  $\mathcal{O}(\frac{n}{N} \cdot \log(n))$ .

Áp dụng quy tắc cộng ta có chi phí thời gian phân hoạch dữ liệu ở bước I là  $\mathcal{O}(\frac{n}{N} \cdot \log(n))$ .

II. Chi phí thời gian tính toán trên mỗi bộ xử lý như sau

1. Chi phí tính toán bước II.1:

a. Chi phí tính toán cho thuật toán ODP trên mỗi bộ xử lý là  $\mathcal{O}(\frac{n}{N} \cdot m \cdot \log(q))$ .

b. Chi phí tính toán bước II.1.b là  $\mathcal{O}(\frac{n}{N} \cdot \log(k+1))$ .

Do đó chi phí tính toán bước II.1 là  $\mathcal{O}(\frac{n}{N} \cdot m \cdot \log(q))$ .

2. Chi phí tính toán bước II.2 là  $\mathcal{O}(1)$ .

Áp dụng quy tắc cộng ta có chi phí tính toán bước II là  $\mathcal{O}(\frac{n}{N} \cdot m \cdot \log(q))$ .

Vì thời gian truyền dữ liệu cho các bộ xử lý là không đáng kể và giữa các bộ xử lý không truyền dữ liệu cho nhau nên thời gian truyền thông của các bộ xử lý bằng 0. Vậy thuật toán 2 có chi phí tính toán là  $\mathcal{O}(\frac{n}{N} \cdot \log(n)) + \mathcal{O}(\frac{n}{N} \cdot m \cdot \log(q))$ .

## 2.5. Thử nghiệm số

Mục tiêu của thử nghiệm là so sánh hiệu quả về mặt thời gian giữa việc sử dụng thuật toán tuần tự và sử dụng thuật toán song song hóa quá trình chọn bộ tâm hỗ trợ và tính véc tơ trọng số cho phương pháp RBF-FD giải phương trình Poisson, điều này được thể hiện thông qua thời gian chạy của số bộ xử lý trong các thử nghiệm số.

Chúng tôi thử nghiệm trên bộ tâm  $\Xi$  được tạo bởi PDE Toolbox của MATLAB cho FEM. Với phương pháp RBF-FD, chỉ sử dụng lưới thô nhất của FEM, sau đó sử dụng Thuật toán 2 trong [18] để thêm các điểm vào bộ tâm  $\Xi$  qua các lần làm mịn, xem chi tiết trong [10, 18]. Véc tơ trọng số được tính bằng cách sử dụng hàm nội suy RBF Gauss-QR  $\varphi(r) = e^{-(\varepsilon r)^2}$ , với tham số hình dạng  $\varepsilon = 10^{-5}$ .

Để so sánh sự hiệu quả của phương pháp RBF-FD chúng tôi dùng công thức sai số trung bình bình phương  $rms$  (root mean square)

$$rms := \left( \frac{1}{\#\Xi_{\text{int}}} \sum_{\zeta \in \Xi_{\text{int}}} (\tilde{u}(\zeta) - u(\zeta))^2 \right)^{1/2},$$

trong đó  $\#\Xi_{\text{int}}$  là số tâm trong miền. Kết quả sai số  $rms$  của phương pháp RBF-FD được đối sánh với FEM trong các thử nghiệm số. Các tham số được sử dụng trong thuật toán ODP là  $v = 2.5, c = 3.0, k = 6, m = 50$ .

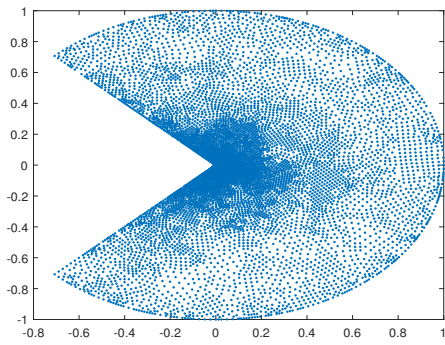
**Bài toán 1.** Xét bài toán Dirichlet với phương trình Laplace  $\Delta u = 0$  trong miền hình quạt  $\Omega$  được xác định bởi bất phương trình  $r < 1, -3\pi/4 < \varphi < 3\pi/4$  trong tọa độ cực, điều kiện biên Dirichlet được cho bởi  $u(r, \varphi) = \cos(2\varphi/3)$  dọc theo cung cong và  $u(r, \varphi) = 0$  dọc theo hai đoạn thẳng. Nghiệm giải tích là  $u(r, \varphi) = r^{2/3} \cos(2\varphi/3)$ .

Kết quả thử nghiệm số của Bài toán (1) được minh họa trong Bảng (2.1) và Hình (2.2).

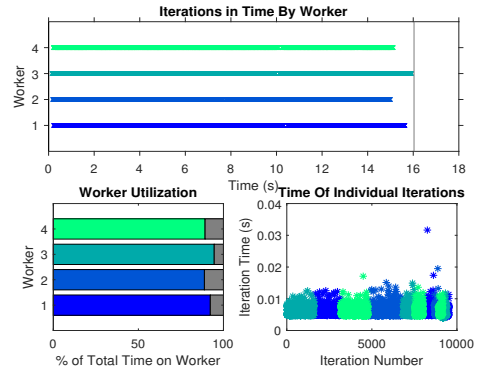
$\#\Xi_{\text{int}}$	Sai số $rms$		Thời gian chạy (giây)		
	FEM	RBF-FD	1 bộ xử lý	2 bộ xử lý	4 bộ xử lý
91	4.01e-03	2.82e-03	0,48	0,62	0,21
140	1.79e-03	7.32e-04	0,32	0,30	0,12
229	9.31e-04	4.57e-04	1,65	1,63	0,59
416	3.90e-04	1.32e-04	6,79	7,33	2,88
828	3.23e-04	1.43e-04	37,29	21,21	11,92
1566	2.57e-04	4.66e-05	101,51	75,02	25,01
2875	2.11e-04	3.07e-05	259,12	208,20	117,61
5210	1.53e-04	1.81e-05	883,43	695,67	409,84
9528	1.31e-04	8.54e-06	3328,28	2435,99	1203,44
17676	9.45e-05	4.26e-06	12001,34	9042,83	4706,90
32841	5.47e-06	2.85e-06	23404,76	14392,11	7255,46

Bảng 2.1: Các sai số  $rms$  so với nghiệm chính xác của FEM và phương pháp RBF-FD, thời gian chạy của phương pháp RBF-FD trên các bộ xử lý của Bài toán 1.

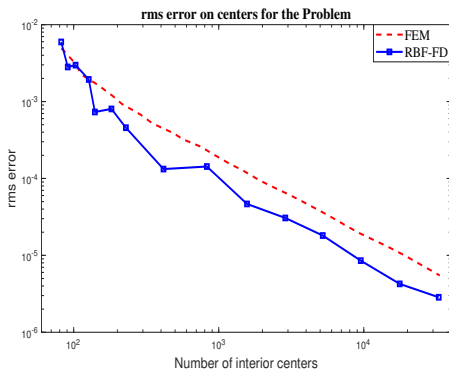
Hình (2.2)(b) minh họa việc phân luồng dữ liệu trong tính toán song song khi sử dụng thuật toán ODP cho phương pháp RBF-FD, thử nghiệm với 9528 tâm và 4 bộ xử lý (4 worker), tương ứng với 4 màu khác nhau. Hình (2.2)(d) biểu diễn thời gian tính toán trên các bộ xử lý của phương pháp RBF-FD. Ta thấy mỗi bộ xử lý có số tâm và thời gian tính toán xấp xỉ nhau.



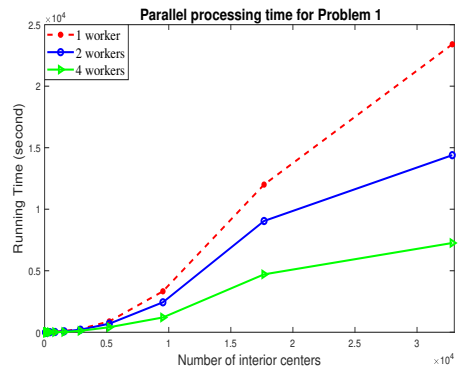
(a) Miền ( $\Omega$ ) với (10158) tâm



(b) Phân luồng dữ liệu trên 4 bộ xử lý với (9528) tâm



(c) Các sai số *rms*



(d) Thời gian chạy trên các bộ xử lý

Hình 2.2: Kết quả thử nghiệm số của Bài toán 1: (a) Miền  $\Omega$  của phương pháp RBF-FD. (b) Phân luồng dữ liệu chạy song song trên 4 bộ xử lý của phương pháp RBF-FD. (c) Các sai số *rms* của FEM và phương pháp RBF-FD. (d) Thời gian chạy của phương pháp RBF-FD trên các bộ xử lý.

Sai số rms của phương pháp FEM được thể hiện trong cột thứ 2 của Bảng (2.1) và đường màu đỏ, nét rời có nhãn “FEM” trong Hình (2.2)(c). Sai số rms của phương pháp RBF-FD là cột thứ 3 trong Bảng (2.1) và đường màu xanh, nét liền có nhãn “RBF-FD” trong Hình (2.2)(c). Các kết quả thử nghiệm này cho thấy độ chính xác của phương pháp không lưới RBF-FD tốt hơn FEM.

Thời gian tính toán của phương pháp RBF-FD trên các bộ xử lý được thể hiện trong các Cột 4, 5, 6 của Bảng (2.1) và Hình (2.2)(d), cụ thể: Cột 4 và đường gắn nhãn ‘1 worker’ biểu diễn thời gian tính toán tuần tự, Cột 5, 6 tương ứng với đường có nhãn ‘2 workers’ và đường ‘4 workers’ biểu diễn thời gian của quá trình song song với 2 bộ xử lý và 4 bộ xử lý. Các kết quả này cho thấy khi miền có mật độ tâm phân bố càng nhiều thì hiệu quả của việc áp dụng quá trình song song vào việc chọn tâm và tính véc tơ trọng số càng cao, thời gian chạy của

phương pháp RBF-FD giảm mạnh khi số bộ xử lý tăng lên.

**Bài toán 2.** Xét bài toán Dirichlet với phương trình Laplace  $\Delta u = 0$  trong miền  $\Omega = (0.01, 1.01)^2$  với điều kiện biên Dirichlet được chọn thỏa mãn nghiệm giải tích  $u(x, y) = \log(x^2 + y^2)$ .

Tương tự như Bài toán 1, nghiệm chính xác của bài Bài toán 2 cũng có kỳ dị tại các điểm quanh gốc tọa độ, nên số tâm quanh gốc tọa độ được làm mịn dày lên rất nhiều [18]. Các kết quả số của bài toán này được trình bày trong các Bảng 2.2 và Hình 2.3.

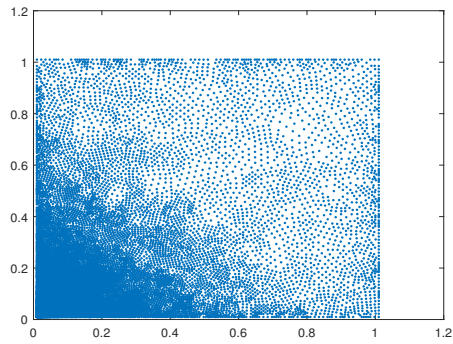
# $\mathcal{E}_{\text{int}}$	Sai số <i>rms</i>		Thời gian chạy (giây)		
	FEM	RBF-FD	1 bộ xử lý	2 bộ xử lý	4 bộ xử lý
144	8.90e-03	1.85e-02	0,30	0,34	0,3702
207	5.26e-03	2.42e-03	0,98	0,89	0,5382
302	4.19e-03	1.14e-03	2,86	2,90	1,466
489	2.37e-03	1.25e-03	9,42	7,36	4,52
883	1.65e-03	3.83e-04	24,47	22,09	13,38
1776	6.45e-04	2.45e-04	73,05	80,27	43,29
3287	4.10e-04	1.14e-04	323,12	322,98	145,5
5822	1.57e-04	6.53e-05	1398,08	947,92	460,3
11146	1.00e-04	4.98e-05	4914,02	3490,87	1794
20604	4.64e-05	2.96e-05	18109,33	12445,85	7467,5

Bảng 2.2: Các sai số *rms* so với nghiệm chính xác của FEM và phương pháp RBF-FD, thời gian chạy của phương pháp RBF-FD trên các bộ xử lý của Bài toán 2.

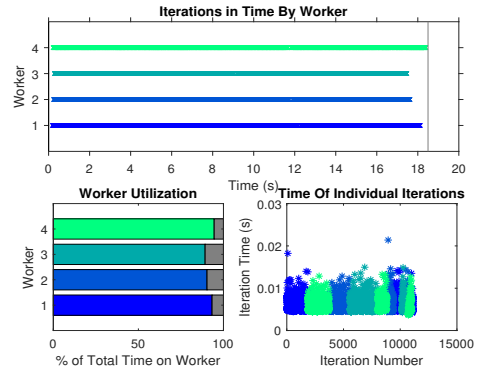
Sử hiệu quả của phương pháp không lưới RBF-FD được thể hiện trong tất cả các thử nghiệm số. Sai số *rms* của phương pháp RBF-FD luôn tốt hơn FEM và thời gian chạy của phương pháp RBF-FD được cải thiện rất nhiều khi sử dụng thuật toán song song.

## 2.6. Kết luận

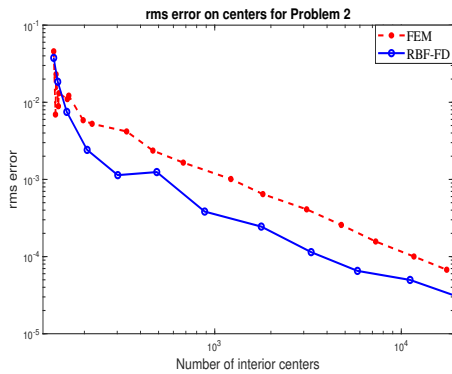
Trong chương này chúng tôi đã trình bày thuật toán chọn tập các tâm hỗ trợ tuần tự và thuật toán song song hóa việc chọn bộ tâm hỗ trợ và tính véc tơ trọng số cho phương pháp RBF-FD cùng với các kết quả thử nghiệm trên các thuật toán. Các thử nghiệm số cho thấy thuật toán song song đã cải thiện được thời gian tính toán, điều đó chứng tỏ hiệu quả của thuật toán này.



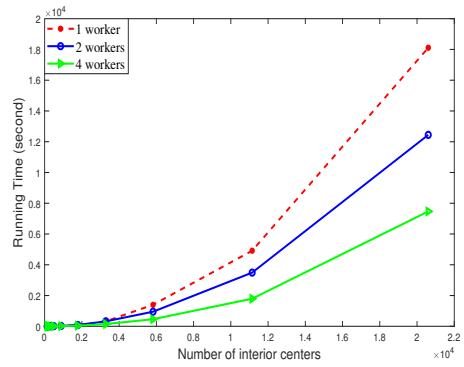
(a) Miền ( $\Omega$ ) với (21680) tâm



(b) Phân luồng dữ liệu trên 4 bộ xử lý với (11146) tâm



(c) Các sai số *rms*



(d) Thời gian chạy trên các bộ xử lý

Hình 2.3: Kết quả thử nghiệm số của Bài toán 2: (a) Miền  $\Omega$  của phương pháp RBF-FD. (b) Phân luồng dữ liệu chạy song song trên 4 bộ xử lý của phương pháp RBF-FD. (c) Các sai số *rms* của FEM và phương pháp RBF-FD. (d) Thời gian chạy của phương pháp RBF-FD trên các bộ xử lý.

## Chương 3

# Phương pháp không lưới RBF-FD trong không gian ba chiều

Trong chương này chúng tôi trình bày phương pháp không lưới RBF-FD giải phương trình truyền nhiệt dừng trong không gian 3 chiều trong Mục 3.1; Các thuật toán chọn bộ tâm hỗ trợ dựa trên các góc khối cho phương pháp RBF-FD Mục 3.2; Các kết quả thử nghiệm số Mục 3.3 và cuối cùng là phần Kết luận.

Các kết quả trong chương này được công bố trong [12].

### 3.1. Phương pháp không lưới RBF-FD

Xét bài toán truyền nhiệt dừng trong không gian ba chiều (1.11)- (1.12) như sau: Cho miền mở  $\Omega \subset \mathbb{R}^3$  và các hàm số  $f$  xác định trên  $\Omega$ ,  $g$  xác định trên  $\partial\Omega$ . Tìm hàm  $u : \bar{\Omega} \rightarrow \mathbb{R}$  thỏa mãn

$$\begin{aligned} \Delta u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega. \end{aligned} \quad (3.1)$$

trong đó  $\Delta$  là toán tử Laplace.

Phương pháp không lưới tính toán nghiệm xấp xỉ rời rạc  $\hat{u}$  của (3.1) trên tập các nút rời rạc hữu hạn  $\Xi \subset \bar{\Omega}$  như sau. Gọi  $\Xi_{\text{int}} := \Xi \cap \Omega$  là tập các tâm trong miền và  $\partial\Xi := \Xi \cap \partial\Omega$  là tập các tâm trên biên. Với mỗi  $\zeta \in \Xi_{\text{int}}$ , ta chọn được tập  $\Xi_{\zeta} := \{\xi_0, \xi_1, \xi_2, \dots, \xi_k\} \subset \Xi$  với  $\xi_0 = \zeta$ , và xấp xỉ  $\Delta u(\zeta)$  bởi công thức vi phân

$$\Delta u(\zeta) \approx \sum_{\xi \in \Xi_{\zeta}} w_{\zeta, \xi} u(\xi), \quad \zeta \in \Xi_{\text{int}}. \quad (3.2)$$

Khi đó bài toán (3.1) được rời rạc hóa thành hệ phương trình

$$\begin{aligned} \sum_{\xi \in \Xi_{\zeta}} w_{\zeta, \xi} \hat{u}(\xi) &= f(\zeta), \quad \zeta \in \Xi_{\text{int}}, \\ \hat{u}(\xi) &= g(\xi), \quad \xi \in \partial\Xi, \end{aligned} \quad (3.3)$$

trong đó  $\hat{u}(\xi)$  là nghiệm xấp xỉ của nghiệm  $u$  của bài toán (3.1) tại các điểm  $\xi \in \Xi$ .

Để giải được hệ phương trình (3.3) ta cần giải quyết 3 vấn đề:



- Làm thế nào để tạo ra được tập các tâm rời rạc  $\Xi$ ?
- Chọn tập các tâm hỗ trợ  $\Xi_\zeta$  như thế nào?
- Cách tính véc tơ trọng số  $w_{\zeta,\xi}$  phù hợp?

Tập các tâm rời rạc  $\Xi$  có thể được tạo ra bằng cách giống nhau là sử dụng phương pháp lưới (phương pháp sai phân hữu hạn, phương pháp phần tử hữu hạn hoặc phương pháp giá trị hữu hạn). Tuy nhiên trong các ứng dụng việc tạo lưới thường rất khó khăn, đặc biệt là cho các mô hình 3D phức tạp. Động lực chính cho các phương pháp không có lưới là không cần đường liên kết giữa các điểm trong  $\Xi$ , điều này giúp đơn giản hóa hơn trong việc tạo nút.

Mỗi phương pháp không lưới đều đưa ra thuật toán chọn tập các tâm rời rạc  $\Xi$  và véc tơ trọng số, đó là việc chọn các tập  $\Xi_\zeta$ ,  $\zeta \in \Xi_{\text{int}}$  và tính các trọng số  $w_{\zeta,\xi}$ ,  $\zeta \in \Xi_{\text{int}}$ ,  $\xi \in \Xi_\zeta$  tương ứng. Cách tính véc tơ trọng số ảnh hưởng đến độ chính xác của công thức (3.2), nó có thể được tính bằng cách sử dụng các đa thức có bậc nhất định hoặc hàm cơ sở bán kính (RBF) (xem [15]). Độ chính xác của công thức vi phân số trên các tâm phân bố không đều của một số phương pháp được giới thiệu trong [7, 8, 9].

Các tập tâm hỗ trợ  $\Xi_\zeta$  có thể được chọn với cấu trúc đơn giản là  $k$  điểm gần  $\zeta$  nhất, để đảm bảo độ chính xác của nghiệm  $\hat{u}$  thì số  $k$  điểm lân cận được chọn phải lớn, trừ khi  $\Xi$  là tập đều địa phương, đây là vấn đề lớn khi tạo ra các nút điểm. Số  $k$  điểm lân cận có ảnh hưởng trực tiếp đến mật độ ma trận của hệ (3.3), thường là  $k + 1$  điểm khác không trên mỗi dòng, do đó một trong những mục tiêu quan trọng của các phương pháp không lưới là giữ  $k$  nhỏ nhất có thể. Ngay cả khi sử dụng thuật toán chọn  $\Xi_\zeta$  tốn kém hơn, thì tính toán song song để giải hệ phương trình tuyến tính thưa (3.3) sẽ giải quyết vấn đề này đối với các bài toán có quy mô lớn. Mật độ của ma trận  $A \in \mathbb{R}^{n \times n}$  là trung bình cộng của các đầu vào khác không trên mỗi dòng, được tính bởi công thức  $\text{nnz}(A)/n$ , với  $\text{nnz}(A)$  là tổng số đầu vào khác không trên mỗi hàng của  $A$ .

Trong các nghiên cứu [10, 18], các tác giả đã phát triển các thuật toán chọn tâm và sinh tâm thích nghi cho phương pháp không lưới RBF-FD giải phương trình elliptic trong không gian 2 chiều, với các tập tâm hỗ trợ được lựa chọn gồm 7 điểm (tức là  $k = 6$ ), nên mật độ ma trận của hệ phương trình (3.3) tương tự như của phương pháp phần tử hữu hạn. Trong phần tiếp theo, chúng tôi sẽ giới thiệu các thuật toán chọn tập các tâm hỗ trợ tính toán véc tơ trọng số trong không gian 3 chiều, là tổng quát của thuật toán chia 4 góc trong không gian 2 chiều

được công bố trong [16]. Với thuật toán chia 8 góc khối, 2 điểm gần nhất trong mỗi góc khối tâm  $\zeta$  sẽ được lựa chọn hoặc với thuật toán chia 16 góc khối, sẽ lựa chọn điểm gần nhất trong mỗi góc khối. Trong cả hai trường hợp đều chọn  $k \leq 16$ , nên trong các thử nghiệm số mật độ ma trận của hệ cao hơn một chút so với mật độ của ma trận cứng ứng với FEM trong không gian 3 chiều với các hàm tuyến tính. Phương pháp FD trong không gian 3 chiều được các tác giả giới thiệu trong [21] với đề xuất lựa chọn 3 điểm trên mỗi góc khối, tuy nhiên nó dẫn đến ma trận hệ số có 25 đầu vào khác không trên mỗi hàng, do đó mật độ của ma trận hệ số xấp xỉ mật độ ma trận hệ số của FEM bậc 2 (khoảng 27), gần với các phương pháp bậc cao.

### 3.2. Thuật toán chọn tâm dựa trên các góc khối

Với mỗi  $\zeta \in \Xi_{\text{int}}$ , mục tiêu của thuật toán là chọn các điểm  $\xi_1, \xi_2, \dots, \xi_k \in \Xi$  xung quanh góc  $\zeta$  sao cho khoảng cách  $\|\zeta - \xi_i\|$ ,  $i = 1, 2, \dots, k$  là nhỏ, trong khi vẫn giữ được các điểm  $\xi_1, \xi_2, \dots, \xi_k$  đều nhất có thể. Để đạt được mục tiêu đó, chúng tôi tính toán và phân hoạch tập  $m$  điểm gần nhất  $\{\xi_1, \dots, \xi_m\} \subset \Xi \setminus \{\zeta\}$  vào 8 góc khối (octants) có gốc tại  $\zeta$ , sau đó lựa chọn 2 điểm gần nhất trên mỗi góc khối. Khi đó  $\Xi_\zeta$  gồm các điểm này và  $\zeta$ .

Trong hệ trục tọa độ Euclide 3 chiều, mỗi góc khối trong 8 góc khối có gốc  $\zeta$  được xác định bởi dấu của tọa độ véc tơ  $\overrightarrow{\zeta\xi_i} = \xi_i - \zeta := (x_i, y_i, z_i)$ ,  $i = 1, 2, \dots, m$  ( $m > k$ ), trong đó  $\xi_1, \xi_2, \dots, \xi_m$  là  $m$  điểm gần  $\zeta$  nhất trong  $\Xi \setminus \{\zeta\}$ . Ký hiệu góc khối thứ  $j$  là  $O_j$ ,  $j = 1, 2, \dots, 8$ , mỗi góc khối được xác định bằng dấu của véc tơ  $\xi_i - \zeta$  tương ứng trong Bảng 3.1, trong đó dấu bằng '+' chứa cả giá trị bằng 0.

Góc khối	$x$	$y$	$z$
$O_1$	+	+	+
$O_2$	+	+	-
$O_3$	+	-	+
$O_4$	+	-	-
$O_5$	-	+	+
$O_6$	-	+	-
$O_7$	-	-	+
$O_8$	-	-	-

Bảng 3.1: 8 - góc khối

Ý tưởng của thuật toán 8 - góc khối (octants) là chọn 2 điểm gần  $\zeta$  trong mỗi góc khối

cho tập  $\Xi_\zeta$ . Ta bắt đầu với  $m$  điểm  $\{\xi_1, \xi_2, \dots, \xi_m\} \subset \Xi \setminus \{\zeta\}$  gần  $\zeta$  nhất (trong thử nghiệm  $m = 99$  để tập này có 100 điểm gồm cả  $\zeta$ ). Các điểm này được phân hoạch trên các góc khối và 2 điểm gần nhất sẽ được chọn vào tập  $\Xi_\zeta$  (nếu có). Nếu tất cả các góc khối có đủ 2 điểm thì tập  $\Xi_\zeta$  có 17 điểm kể cả  $\zeta$ . Nếu các góc khối không chứa điểm nào hoặc có 1 điểm thì tập  $\Xi_\zeta$  có ít hơn 17. Nếu đoạn thẳng nối  $\zeta$  với điểm  $\xi_i$  chứa điểm không thuộc  $\Omega$  thì  $\xi_i$  được thay thế bằng điểm gần nhất trên biên (điều này chỉ xảy ra khi  $\Omega$  là miền không lồi, xem ví dụ Bài 5 phía sau). Các điểm biên được lựa chọn nằm trong tập  $\Xi'_\zeta$  và được thêm vào miền rời rạc  $\Xi$  sau khi kết thúc quá trình tìm tập  $\Xi_\zeta$  với tất cả các điểm  $\zeta$  nằm trong miền.

Thuật toán 8 góc khối như sau:

**Thuật toán 3.** 8- góc khối.

*Input:*  $\Xi, \zeta \in \Xi_{\text{int}}$ .

*Output:*  $\Xi_\zeta, \Xi'_\zeta$ .

*Tham số:*  $m \geq 16$  (số điểm được lựa chọn ban đầu gồm cả  $\zeta$ );  $m = 99$  trong các thử nghiệm số.

*Khởi tạo:*  $\Xi_\zeta := \{\zeta\}, \Xi'_\zeta := \emptyset$ .

- I. Tìm  $m$  điểm  $\xi_1, \xi_2, \dots, \xi_m$  thuộc  $\Xi \setminus \{\zeta\}$  và gần  $\zeta$  nhất.
- II. Phân hoạch các điểm  $\xi_1, \xi_2, \dots, \xi_m$  vào 8 góc khối  $O_j = \{\xi_{j1}, \xi_{j2}, \dots\}$ ,  $j = 1, 2, \dots, 8$ , tương ứng với 8 góc khối trong Bảng 3.1, thỏa mãn  $\|\xi_{j1} - \zeta\| \leq \|\xi_{j2} - \zeta\| \leq \dots$ .
- III. For  $j = 1$  to 8
  - a. If  $\#O_j = 1$  then  $\Xi_\zeta := \Xi_\zeta \cup \{\xi_{j1}\}$ .
  - b. ElseIf  $\#O_j > 1$  then  $\Xi_\zeta := \Xi_\zeta \cup \{\xi_{j1}, \xi_{j2}\}$ .
- IV. Với mỗi  $\xi \in \Xi_\zeta \setminus \{\zeta\}$ , xét đoạn thẳng  $(\zeta, \xi) = \{\zeta + \alpha(\xi - \zeta) : 0 < \alpha < 1\}$ .  
If  $(\zeta, \xi) \cap \partial\Omega \neq \emptyset$ , then  $\Xi_\zeta := \Xi_\zeta \setminus \{\xi\} \cup \{\xi'\}$  và  $\Xi'_\zeta := \Xi'_\zeta \cup \{\xi'\}$ , trong đó  $\xi'$  là điểm thuộc  $(\zeta, \xi) \cap \partial\Omega$  gần  $\zeta$  nhất.

Sau khi áp dụng Thuật toán 8 góc khối cho tất cả các điểm  $\zeta \in \Xi_{\text{int}}$ , ta cập nhật tập  $\Xi$  bởi công thức

$$\Xi := \Xi \cup \bigcup_{\zeta \in \Xi_{\text{int}}} \Xi'_\zeta. \quad (3.4)$$

Bằng cách tương tự, chúng tôi chia đôi mỗi góc khối trong 8 góc khối bởi một mặt phẳng, khi đó ta có thuật toán 16 - góc khối với một điểm gần nhất trong mỗi góc khối sẽ được lựa chọn. Cụ thể, phân hoạch các điểm  $\{\xi_1, \xi_2, \dots, \xi_m\}$  vào 16 góc khối có góc  $\zeta$  được xác định bởi dấu của tọa độ véc tơ  $\overrightarrow{\xi_i \zeta} = \zeta - \xi_i := (x_i, y_i, z_i)$ ,  $i = 1, 2, \dots, m$  ( $m > k$ ) và các giá trị  $x_i - y_i, x_i - z_i, y_i - z_i$  tương ứng trong Bảng 3.2. Khi đó điểm gần nhất trên mỗi góc khối không rỗng sẽ được chọn vào tập  $\Xi_\zeta$  và  $\Xi'_\zeta$  có nhiều nhất 17 điểm.

Góc khối	$x$	$y$	$z$	điều kiện
$O_1$	+	+	+	$x \geq y$
$O_2$	+	+	+	$x < y$
$O_3$	+	+	-	$x \geq y$
$O_4$	+	+	-	$x < y$
$O_5$	+	-	+	$x \geq z$
$O_6$	+	-	+	$x < z$
$O_7$	+	-	-	$-y \geq -z$
$O_8$	+	-	-	$-y < -z$
$O_9$	-	+	+	$y \geq z$
$O_{10}$	-	+	+	$y < z$
$O_{11}$	-	+	-	$-x \geq -z$
$O_{12}$	-	+	-	$-x < -z$
$O_{13}$	-	-	+	$-x \geq -y$
$O_{14}$	-	-	+	$-x < -y$
$O_{15}$	-	-	-	$-x \geq -z$
$O_{16}$	-	-	-	$-x < -z$

Bảng 3.2: 16 - góc khối.

Nội dung của thuật toán 16 góc khối:

**Thuật toán 4.** 16-góc khối.

*Input:*  $\Xi, \zeta \in \Xi_{\text{int}}$ .

*Output:*  $\Xi_\zeta, \Xi'_\zeta$ .

*Tham số:*  $m \geq 16$  (số điểm được lựa chọn ban đầu gồm cả  $\zeta$ );  $m = 99$  trong các thử nghiệm số.

*Khởi tạo:*  $\Xi_\zeta := \{\zeta\}, \Xi'_\zeta := \emptyset$ .

I. Tìm  $m$  điểm  $\xi_1, \xi_2, \dots, \xi_m$  thuộc  $\Xi \setminus \{\zeta\}$  và gần  $\zeta$  nhất.

II. Phân hoạch các điểm  $\xi_1, \xi_2, \dots, \xi_m$  vào 16 tập  $O_j = \{\xi_{j1}, \xi_{j2}, \dots\}$ ,  $j = 1, 2, \dots, 16$ , tương ứng với 16 góc khối như trong Bảng 3.2, thỏa mãn  $\|\xi_{j1} - \zeta\| \leq \|\xi_{j2} - \zeta\| \leq \dots$ .

III. For  $j = 1$  to 16.

If  $\#O_j \geq 1$  then  $\Xi_\zeta := \Xi_\zeta \cup \{\xi_{j1}\}$ .

IV. Với mỗi  $\xi \in \Xi_\zeta \setminus \{\zeta\}$ , xét đoạn thẳng  $(\zeta, \xi)$ .

If  $(\zeta, \xi) \cap \partial\Omega \neq \emptyset$ , then  $\Xi_\zeta := \Xi_\zeta \setminus \{\xi\} \cup \{\xi'\}$  và  $\Xi'_\zeta := \Xi'_\zeta \cup \{\xi'\}$ , trong đó  $\xi'$  thuộc  $(\zeta, \xi) \cap \partial\Omega$  và gần  $\zeta$  nhất.

Sau khi áp dụng Thuật toán 16 góc khối cho tất cả các điểm  $\zeta \in \Xi_{\text{int}}$ , ta cập nhật  $\Xi$  bởi công thức (3.4).

### 3.3. Thử nghiệm số

Mục tiêu của các thử nghiệm số là đánh giá hiệu quả của thuật toán 8 góc khối và 16 góc khối trong việc so sánh với FEM dựa trên các hàm tuyến tính, do đó chúng tôi rời rạc các miền là các tập  $\Xi$  gồm các đỉnh của các tứ diện được tạo bởi hàm generateMesh trong PDE Toolbox. Với mỗi bài toán thử nghiệm, chúng tôi chọn độ dài cạnh lưới tối đa Hmax cho phép tạo ra các tam giác lưới thô nhất, sau đó tạo các tam giác lưới mịn hơn bằng cách giảm Hmax nhiều lần với hệ số  $2^{-1/3}$  và được gần gấp đôi số đỉnh. Số lượng đỉnh trong miền ký hiệu là  $\#\Xi_{\text{int}}$  tương ứng trong các bảng ghi kết quả thử nghiệm số của mỗi bài toán.

Với mỗi  $\zeta \in \Xi_{\text{int}}$ , ta chọn được tập tâm hỗ trợ  $\Xi_\zeta = \{\zeta \equiv \xi_0, \xi_1, \dots, \xi_k\}$  và véc tơ trọng số  $w$  tương ứng được tính bằng cách sử dụng hàm cơ sở bán kính RBF [15] như sau

$$w := \left( \Phi_{\Xi_\zeta} \right)^{-1} D\Phi(\zeta - \cdot)|_{\Xi_\zeta}, \quad (3.5)$$

trong đó

$$\Phi_{\Xi_\zeta} := \begin{pmatrix} \Phi(\xi_0 - \xi_0) & \Phi(\xi_0 - \xi_1) & \cdots & \Phi(\xi_0 - \xi_k) \\ \Phi(\xi_1 - \xi_0) & \Phi(\xi_1 - \xi_1) & \cdots & \Phi(\xi_1 - \xi_k) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\xi_k - \xi_0) & \Phi(\xi_k - \xi_1) & \cdots & \Phi(\xi_k - \xi_k) \end{pmatrix}$$

và

$$D\Phi(\zeta - \cdot)|_{\Xi_\zeta} := [D\Phi(\zeta - \xi_0), D\Phi(\zeta - \xi_1), \dots, D\Phi(\zeta - \xi_k)]^T,$$

với hàm RBF  $\varphi(r) = r^5$ . Do  $\Phi(x) := \varphi(\|x\|)$ ,  $x \in \mathbb{R}^d$  là hàm cơ sở bán kính xác định dương nên ma trận  $\Phi_{\Xi_\zeta}$  luôn khả nghịch và ta luôn tìm được duy nhất véc tơ trọng số trong công thức (3.5).

Các tập tâm hỗ trợ  $\Xi_\zeta$ ,  $\zeta \in \Xi_{\text{int}}$  được chọn bởi thuật toán 8 góc khối (RBF-FD 1) và 16 góc khối (RBF-FD 2) hoặc chọn  $\Xi_\zeta$  gồm  $\zeta$  và tất cả các nút  $\xi \in \Xi$  có liên kết với  $\zeta$  bởi một cạnh của tam giác FEM (RBF-FD 3). Phương pháp RBF-FD 3 ít được quan tâm vì nó dựa vào lưới của phần tử hữu hạn, nhưng chúng tôi giới thiệu để so sánh với các phương pháp khác. Hơn nữa, để so sánh sai số của phương pháp RBF-FD, đối với Bài toán 5 chúng tôi còn thêm trường hợp chọn tập các tâm hỗ trợ tính toán véc tơ trọng số cho phương pháp RBF-FD bằng các điểm gần  $\zeta$  nhất.

Để đánh giá sự hiệu quả của các thuật toán, chúng tôi so sánh giá trị sai số tương đối trung bình bình phương RRMS (relative root mean square) và coi nó như độ đo độ chính xác của nghiệm xấp xỉ  $\hat{u}$  của hệ (3.3) với nghiệm chính xác  $u$  của hệ (3.1) tại các điểm trong miền, công thức tính sai số RRMS như sau

$$E_c = \text{RRMS}(u, \hat{u}, \Xi_{\text{int}}) := \sqrt{\frac{\sum_{\zeta \in \Xi_{\text{int}}} (u(\zeta) - \hat{u}(\zeta))^2}{\sum_{\zeta \in \Xi_{\text{int}}} (u(\zeta))^2}}. \quad (3.6)$$

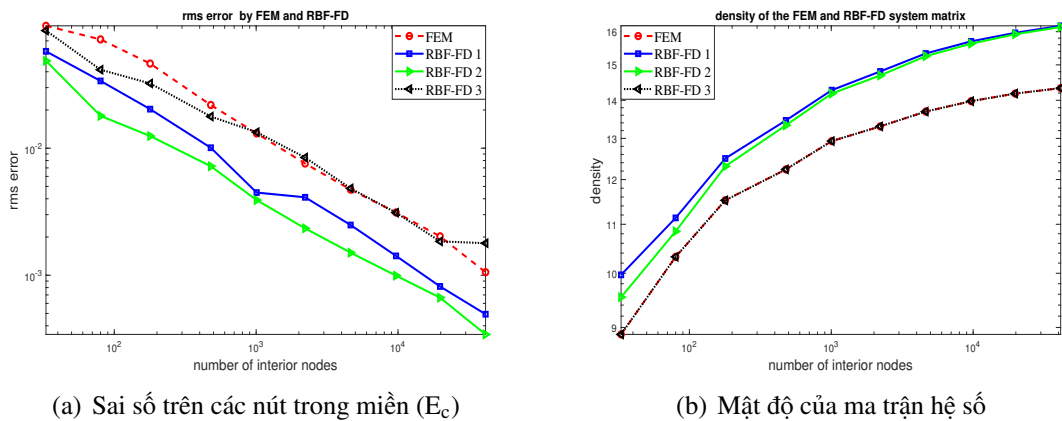
**Bài toán 3.** Xét phương trình Poisson  $\Delta u = -3\pi^2 \sin \pi x \sin \pi y \sin \pi z$  trên miền  $\Omega = [0, 1]^3$  với điều kiện biên Dirichlet đồng nhất  $u|_{\partial\Omega} = 0$ . Nghiệm chính xác của bài toán là  $u(x, y, z) = \sin \pi x \sin \pi y \sin \pi z$ .

Kết quả thử nghiệm số của bài toán được biểu diễn trong Bảng 3.3 và Hình 3.1. Lưới tam giác thô nhất có 33 điểm trong miền ứng với giá trị  $H_{\text{max}} = 0.25$ . Tiếp theo là lưới tam giác có 80 điểm trong miền được tạo bởi hàm `generateMesh` với  $H_{\text{max}} = 0.25/2^{1/3}$ , ... Ngoài các sai số được giới thiệu trong Bảng 3.3 và Hình 3.1(a), Hình 3.1(b) so sánh mật độ của các ma trận hệ số. Khi tính mật độ ma trận của hệ (3.3) và của ma trận cứng ứng với FEM đã loại đi các điểm trên biên, do các điểm nằm trên biên chiếm một tỷ lệ đáng kể nên mật độ của ma trận hệ số thấp hơn so với miền rời rạc thô. Mật độ của ma trận hệ số gần bằng 16 khi sử dụng thuật toán RBF-RD 1, RBF-FD 2 với tập điểm lớn nhất và xấp xỉ 14 cho cả FEM và phương pháp RBF-FD 3.

**Bài toán 4.** Xét phương trình Poisson  $\Delta u = 3e^{x+y+z}$  trong khối cầu đơn vị  $\Omega = \{(x, y, z) \in$

# $E_{\text{int}}$	Sai số RRMS trên các nút trong miền ( $E_c$ )			
	FEM	RBF-FD 1	RBF-FD 2	RBF-FD 3
33	9.19e-02	5.79e-02	4.87e-02	8.44e-02
80	7.19e-02	3.38e-02	1.80e-02	4.15e-02
179	4.64e-02	2.03e-02	1.25e-02	3.24e-02
479	2.19e-02	1.01e-02	7.21e-03	1.78e-02
1008	1.31e-02	4.47e-03	3.90e-03	1.34e-02
2213	7.58e-03	4.11e-03	2.34e-03	8.45e-03
4633	4.72e-03	2.49e-03	1.51e-03	4.83e-03
9684	3.12e-03	1.42e-03	9.93e-04	3.11e-03
19776	2.02e-03	8.16e-04	6.66e-04	1.85e-03
41409	1.06e-03	4.93e-04	3.42e-04	1.79e-03

Bảng 3.3: Các sai số RRMS so với nghiệm chính xác của Bài toán 3.



(a) Sai số trên các nút trong miền ( $E_c$ )

(b) Mật độ của ma trận hệ số

Hình 3.1: Kết quả thử nghiệm số của Bài toán 3. Hình (a) minh họa sai số RRMS trên các tâm trong miền. Hình (b) biểu diễn mật độ của ma trận thưa của hệ (3.3) ứng với phương pháp RBF-FD và của ma trận cứng ứng với FEM.

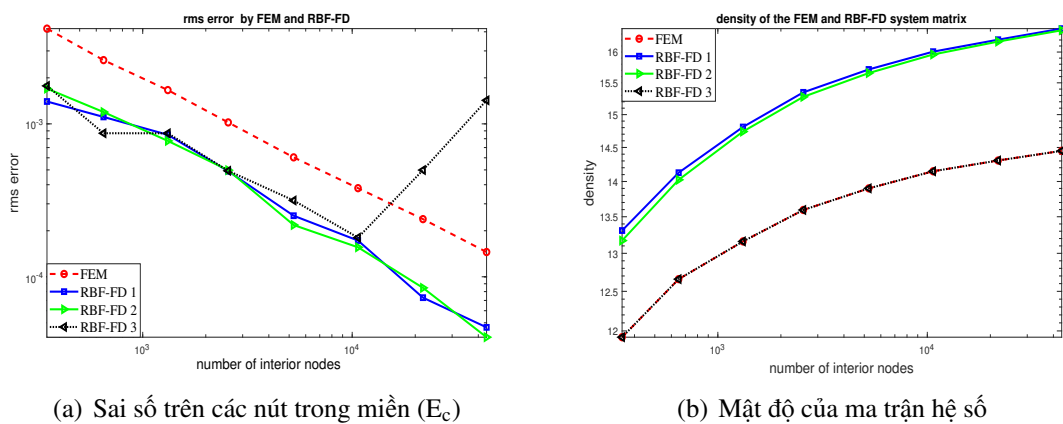
$\mathbb{R}^3 : x^2 + y^2 + z^2 \leq 1$  } với điều kiện biên Dirichlet được chọn sao cho thỏa mãn nghiệm chính xác của bài toán là  $u(x, y, z) = e^{x+y+z}$ .

Kết quả thử nghiệm số của bài toán được biểu diễn trong Bảng 3.4 và Hình 3.2. Các lưới tam giác thô nhất đạt được bởi hàm generateMesh với  $H_{\text{max}} = 0.25$  có 349 điểm trong. Tương tự như Bài toán 3, chúng tôi sử dụng  $E_c$  để đo sự hiệu quả của các thuật toán và được kết quả như sau

**Bài toán 5** ([24, Section 3], Forearm Link). Xét phương trình Poisson  $\Delta u = -10$  với điều kiện biên Dirichlet  $u|_{\partial\Omega} = 0$  trên miền  $\Omega$  trong Hình 3.3(a).

$\#\Xi_{\text{int}}$	Sai số RRMS trên các nút trong miền ( $E_c$ )			
	FEM	RBF-FD 1	RBF-FD 2	RBF-FD 3
349	4.20e-03	1.40e-03	1.70e-03	1.77e-03
650	2.61e-03	1.11e-03	1.19e-03	8.68e-04
1318	1.66e-03	8.48e-04	7.74e-04	8.68e-04
2559	1.02e-03	4.90e-04	4.98e-04	4.93e-04
5254	6.04e-04	2.51e-04	2.18e-04	3.15e-04
10662	3.80e-04	1.72e-04	1.56e-04	1.80e-04
21777	2.38e-04	7.30e-05	8.46e-05	4.99e-04
43813	1.45e-04	4.67e-05	4.03e-05	1.43e-03

Bảng 3.4: Các sai số RRMS so với nghiệm chính xác của Bài toán 4.



Hình 3.2: Kết quả thử nghiệm số của Bài toán 4: (a) Các sai số RRMS trên các nút trong miền. (b) Mật độ của ma trận hệ số.

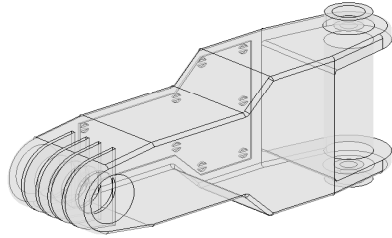


Miền của Bài toán 5 được xuất từ file STL ‘ForearmLink.stl’ của MATLAB PDE Toolbox bằng lệnh `importGeometry`, đó là một bộ phận cơ học với kích thước xấp xỉ giới hạn trong hình hộp  $135 \times 35 \times 61$ . Kết quả thử nghiệm số của bài toán được biểu diễn trong Bảng 3.5, Hình 3.3 và Hình 3.4. Các lưới tam giác thô nhất đạt được tạo ra bởi lệnh `generateMesh` với  $H_{\max} = 6.8$  với 816 điểm trong miền. Cụ thể, chúng tôi đã sử dụng đoạn code sau trong MATLAB 2017b để thiết lập lưới phần tử hữu hạn với các giá trị của  $H_{\max}$  và trích xuất các đỉnh của tam giác trong mảng `nodes`:

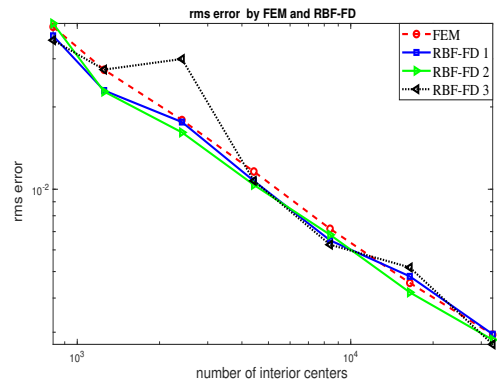
```
model = createpde(1);
importGeometry(model, 'ForearmLink.stl');
generateMesh(model, 'Hmax', Hmax, 'Hmin', Hmax/3, 'GeometricOrder', 'linear');
nodes = model.Mesh.Nodes;
```

Ngược lại với Bài toán 3 và 4, do chưa biết nghiệm chính xác  $u$  của bài toán nên khi tính sai số RRMS trong công thức (3.6) đã thay  $E_c$  bởi  $E_n = \text{RRMS}(\hat{u}_{i+1}, \hat{u}_i, \Xi_{\text{int}}^i)$  và  $E_r = \text{RRMS}(\hat{u}_{\text{ref}}, \hat{u}_i, \Xi_{\text{int}}^i)$ , trong đó  $\hat{u}_i$  và  $\hat{u}_{i+1}$  là nghiệm rời rạc của các phương pháp RBF-FD và FEM tại tập các nút  $\Xi^i$  hiện tại (thứ  $i$ -th) và tập các nút kế tiếp liền kề  $\Xi^{i+1}$  (thứ  $i+1$ -th). Nghiệm tham chiếu  $\hat{u}_{\text{ref}}$  là nghiệm của phương pháp FEM bậc 2 với 641825 nút trên khối đa giác, đạt được bởi lệnh `generateMesh` ứng với  $H_{\max} = 6.8/2^{7/3} \approx 1.35$ . Bằng việc so sánh nghiệm tham chiếu này với nghiệm đạt được bằng FEM trên khối đa giác với 604279 đỉnh, chúng tôi ước lượng sai số  $\hat{u}_{\text{ref}}$  so với nghiệm chính xác của bài toán khoảng 0.001. Sai số  $E_n, E_r$  được tính trên tập  $\Xi_{\text{int}}^i$  sau khi đã nội suy tuyến tính  $\hat{u}_{i+1}$  và nội suy bậc 2  $\hat{u}_{\text{ref}}$  tương ứng. Ngoài việc so sánh với FEM và RBF-FD 3, Hình 3.3(e)–(f) còn biểu diễn sai số và mật độ của ma trận hệ số ứng với phương pháp RBF-FD với trường hợp tập các tâm hỗ trợ tính toán véc tơ trọng số được chọn là  $k$  điểm gần  $\zeta$  nhất, với  $k$  lần lượt nhận các giá trị  $k = 16, k = 18$  và  $k = 20$ .

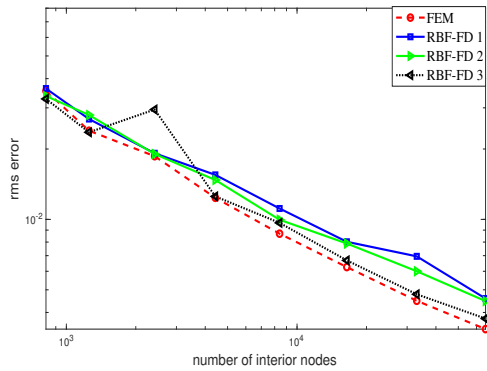
Sự hiệu quả của phương pháp RBF-FD khi sử dụng thuật toán dựa trên các góc khối được thể hiện trong các thử nghiệm số. Các Bài toán 3, 4, sai số RRMS so sánh với nghiệm chính xác thường thấp hơn 2-3 lần so với FEM. Đối với Bài toán 5 có miền phức tạp hơn trên miền không lồi, sai số RRMS được tính toán đối sánh với nghiệm của cùng một phương pháp trên tập các nút mịn hơn liền kề thấp hơn một chút so với FEM, khi sai số RRMS được tính toán với nghiệm tham chiếu là nghiệm của FEM bậc 2 thì sai số của phương pháp RBF-FD cao



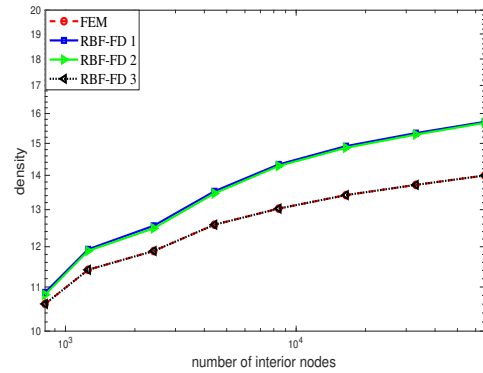
(a) Miền của bài toán



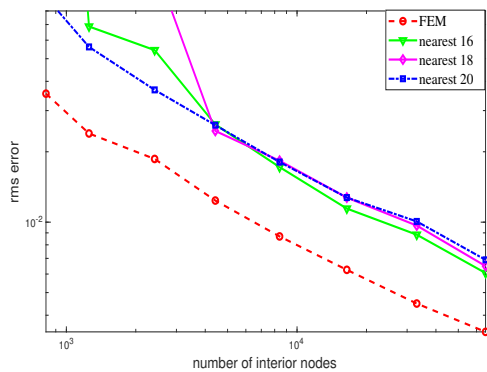
(b) Các sai số so với nghiệm của lần kế tiếp



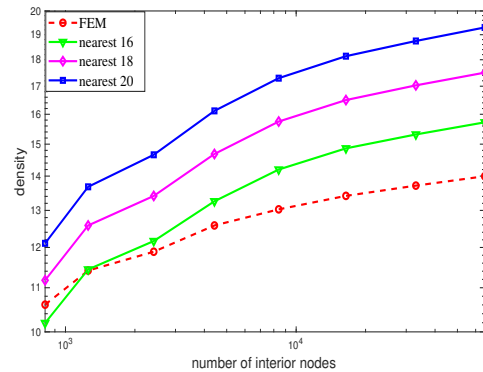
(c) Các sai số so với một nghiệm của FEM



(d) Mật độ của ma trận

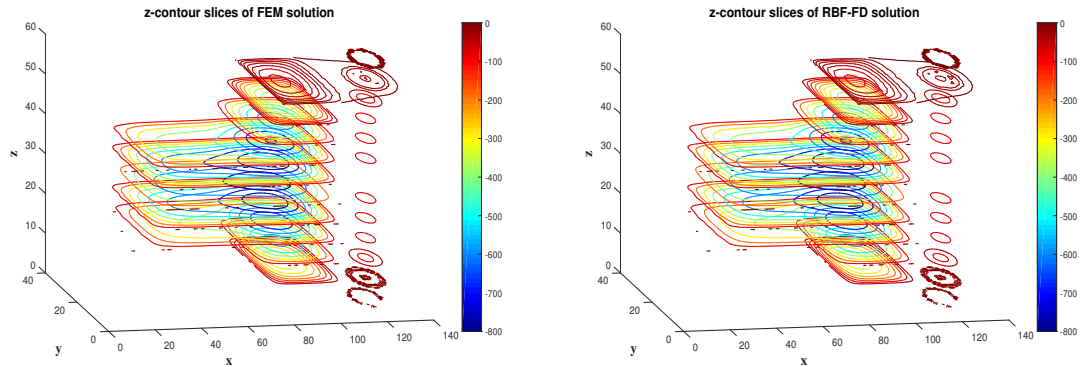


(e) Các sai số so với một nghiệm của FEM

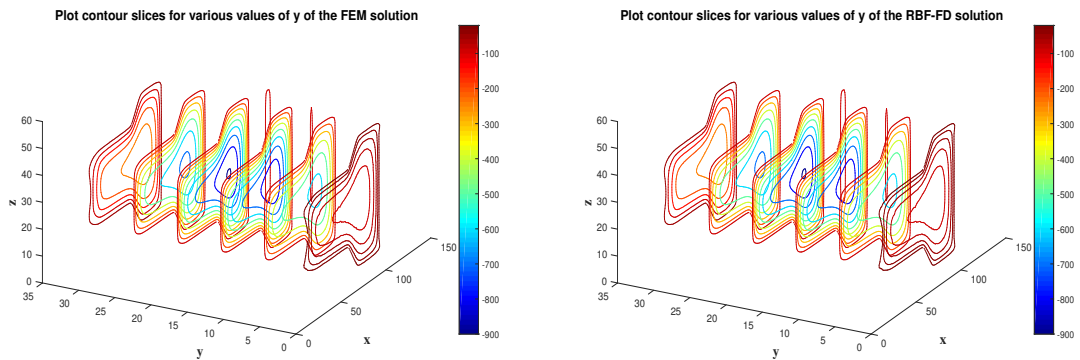


(f) Mật độ của ma trận

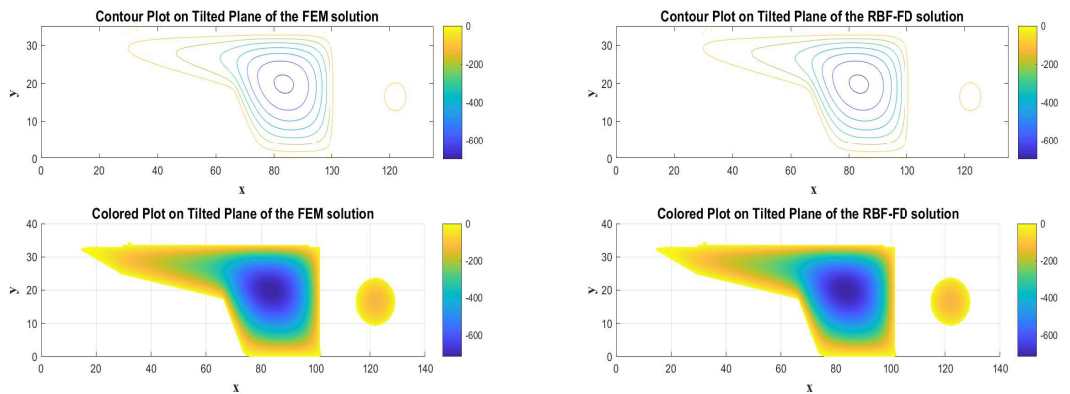
Hình 3.3: (a) Miền  $\Omega$  được tạo bởi lệnh `pdegplot` trong MATLAB PDE Toolbox. (b)–(d) Đồ thị các sai số RRMS và mật độ của ma trận hệ số sử dụng các thuật toán RBF-FD 1, 2, 3 và FEM như trong Bảng 3.5. (e)–(f) Đồ thị các sai số RRMS và mật độ của ma trận hệ số ứng với phương pháp RBF-FD chọn các tập tâm hỗ trợ tính trọng số có 16, 18 hoặc 20 điểm gần  $\zeta$  nhất và FEM.



(a) Đường đồng mức theo biến  $z$  nghiệm của FEM (b) Đường đồng mức theo biến  $z$  nghiệm của RBF-FD



(c) Đường đồng mức theo biến  $y$  nghiệm của FEM (d) Đường đồng mức theo biến  $y$  nghiệm của RBF-FD



(e) Đường đồng mức theo biến  $x$  và bản màu trên mặt phẳng nghiệm của FEM (f) Đường đồng mức theo biến  $x$  và bản màu trên mặt phẳng nghiệm của RBF-FD

Hình 3.4: Đường đồng mức các nghiệm của FEM (bên trái) và RBF-FD 1 (bên phải) với trường hợp miền rời rạc có 65652 điểm trong. (a)–(b) đường đồng mức theo biến  $z$ ; (c)–(d) đường đồng mức theo biến  $y$ ; (e)–(f) đường đồng mức theo biến  $x$  và bản màu trên mặt phẳng.

# $\Xi_{\text{int}}$	Sai số RRMS so với lần kế tiếp ( $E_n$ )				Sai số RRMS so với nghiệm tham chiếu ( $E_r$ )			
	FEM	RBF-FD 1	RBF-FD 2	RBF-FD 3	FEM	RBF-FD 1	RBF-FD 2	RBF-FD 3
816	3.93e-02	3.64e-02	4.04e-02	3.51e-02	3.54e-02	3.63e-02	3.37e-02	3.28e-02
1254	2.73e-02	2.30e-02	2.28e-02	2.74e-02	2.39e-02	2.69e-02	2.79e-02	2.36e-02
2414	1.79e-02	1.76e-02	1.61e-02	2.99e-02	1.86e-02	1.92e-02	1.90e-02	2.95e-02
4423	1.16e-02	1.07e-02	1.04e-02	1.07e-02	1.24e-02	1.55e-02	1.47e-02	1.25e-02
8401	7.17e-03	6.50e-03	6.82e-03	6.27e-03	8.69e-03	1.11e-02	9.97e-03	9.67e-03
16437	4.54e-03	4.80e-03	4.20e-03	5.18e-03	6.25e-03	8.02e-03	7.89e-03	6.68e-03
33030	2.94e-03	2.95e-03	2.81e-03	2.71e-03	4.49e-03	6.95e-03	6.00e-03	4.79e-03
65652	-	-	-	-	3.40e-03	4.59e-03	4.47e-03	3.76e-03

Bảng 3.5: Các sai số RRMS so với nghiệm của lần kế tiếp trên cùng một phương pháp ( $E_n$ ) và so với nghiệm đạt được bởi FEM ( $E_r$ ) của Bài toán 5.

hơn 1,35 lần sai số của FEM. Tuy nhiên các đường đồng mức theo các biến và bản màu trên mặt phẳng (Hình 3.4) cho thấy nghiệm của phương pháp RBF-FD và FEM trên cùng một tâm rời rạc  $\Xi$  là như nhau.

Kết quả thử nghiệm số cũng cho thấy phương pháp RBF-FD sử dụng thuật toán 16-góc khối (RBF-FD 2) hiệu quả hơn khi sử dụng thuật toán 8-góc khối (RBF-FD 1), còn nếu chỉ sử dụng cấu trúc điểm của FEM ( $\Xi_\zeta$  là tập tất cả các điểm thuộc các tứ diện có chứa  $\zeta$ ) cho phương pháp RBF-FD (RBF-FD 3) thì hiệu quả tồi hơn. Kết quả này hoàn toàn tương tự nhưng trong không gian 2 chiều đã được giới thiệu trong [10]. Mặt khác, do số điểm của  $\Xi_\zeta$  cho RBF-FD 3 và FEM nhỏ hơn  $k$  nên mật độ ma trận của hệ (3.3) và ma trận cứng bằng nhau, nhỏ hơn một chút so với RBF-FD1, RBF-FD 2. Hình 3.3 (e) cho thấy sai số của phương pháp RBF-FD khi sử dụng tập  $\Xi_\zeta$  là  $k$  điểm gần  $\zeta$  nhất là kém hơn và cao gấp đôi sai số của FEM. Nếu  $k$  nhỏ ( $k = 16$  hoặc  $k = 18$ ), thì sai số của phương pháp RBF-FD không ổn định với các tam giác thô. Để khắc phục điều này, cần tăng  $k$  lớn hơn, như  $k \geq 20$ , tuy nhiên nó sẽ làm cho mật độ ma trận của hệ phương trình cao hơn 25% so với của RBF-FD 1 và RBF-FD 2 (khoảng 16), xem Hình 3.3 (d) và (f), dẫn đến tăng chi phí khi giải hệ (3.3), điều này hoàn toàn không mong muốn đối với các bài toán có dữ liệu lớn.

### 3.4. Kết luận

Trong chương đã trình bày 2 thuật toán chọn tâm mới cho phương pháp không lưới RBF-FD giải phương trình truyền nhiệt dừng trong không gian 3 chiều. Cụ thể, so sánh sự hiệu quả của phương pháp RBF-FD sử dụng hàm cơ sở bán kính và 2 thuật toán chọn tâm với

FEM trên các nút được tạo bởi lưới tiêu chuẩn. Nghiệm xấp xỉ của các phương pháp RBF-FD chính xác hơn nghiệm của FEM với các bài toán có miền là khối lập phương hoặc khối cầu (Bài toán 3, 4) và có độ chính xác tương đương với bài toán có miền phức tạp hơn (Bài toán 5).

## KẾT LUẬN VÀ KIẾN NGHỊ

Đề tài đạt và vượt mức yêu cầu đối với sản phẩm đặt ra trong thuyết minh.

- Sản phẩm khoa học: Đề tài đề xuất 03 thuật toán mới, cụ thể là:
  - Đề xuất thuật toán song song hóa việc chọn bộ tâm hỗ trợ và tính véc tơ trọng số cho phương pháp RBF-FD trong không gian 2 chiều [3].
  - Đề xuất 2 thuật toán chọn bộ tâm hỗ trợ phương pháp không lưới RBF-FD trong không gian 3 chiều [12].
  - Thử nghiệm số đối với các thuật toán mới.
- Sản phẩm đào tạo: Hướng dẫn 02 luận văn thạc sĩ bảo vệ thành công.

Hướng phát triển nghiên cứu trong thời gian tới: Tiếp tục nghiên cứu phương pháp RBF-FD giải phương trình đạo hàm riêng trong không gian hai chiều và ba chiều.

# Tài liệu tham khảo

## I. Tiếng Việt

- [1] Đoàn Văn Ban, Nguyễn Mậu Hân (2006), "*Xử lý song song và phân tán*", Nhà xuất bản Khoa học và kỹ thuật.
- [2] Tạ Văn Đĩnh(2002), "*Phương pháp sai phân và phương pháp phần tử hữu hạn*", Nhà xuất bản Khoa học và kỹ thuật.
- [3] Đặng Thị Oanh, Ngô Mạnh Tường (2019), "Song song hóa việc chọn tâm và tính véc tơ trọng số cho phương pháp không lưới RBF-FD giải phương trình Poisson", *Tạp chí Khoa học và Công nghệ - Đại học Thái Nguyên*, tập 195, số 02, tr. 69-74.
- [4] Đặng Thị Oanh, Ngô Mạnh Tường, Trịnh Minh Đức (2017), " Nghiên cứu sự ảnh hưởng của điều kiện tách biệt trong thuật toán sinh tâm thích nghi cho phương pháp không lưới giải bài toán elliptic ", *Kỷ yếu Hội nghị Khoa học Công nghệ quốc gia lần thứ X (Fair 2017) về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin*, tr. 667-673.
- [5] Ngô Mạnh Tường (2015), "Phương pháp không lưới RBF-FD sử dụng nội suy Hermite RBF giải phương trình Poisson", *Tạp chí Khoa học và Công nghệ - Đại học Thái Nguyên*, tập 132, số 02, tr. 171-175.

## II. Tiếng Anh

- [6] Buhmann M. D. (2003), "*Radial Basis Functions*", Cambridge University Press, New York, NY, USA.
- [7] Davydov O. and Schaback R. (2016), "Error bounds for kernel-based numerical differentiation", *Numerische Mathematik*, Volume 132(2), pp.243-269.
- [8] Davydov O. and Schaback R. (2018), "Minimal numerical differentiation formulas", *Numerische Mathematik*, Volume 140(3), pp.555-592.
- [9] Davydov O. and Schaback R. (2017), "Optimal stencils in Sobolev spaces", *IMA Journal of Numerical Analysis*, published online 28 December 2017.

- [10] Davydov O. and Oanh D. T. (2011), "Adaptive meshless centres and RBF stencils for Poisson equation", *Journal of Computational Physics*, Volume 230, pp. 287-304.
- [11] Davydov O. and Oanh D. T. (2011), "On the optimal shape parameter for Gaussian radial basis function finite difference approximation of the Poisson equation", *Computers and Mathematics with Applications*, Volume 62, pp. 2143-2161.
- [12] Davydov O., Oanh D. T., Tuong N. M. (2019), " Octant-Based Stencil Selection for Meshless Finite Difference Methods in 3D", *Vietnam journal of mathematics*, accepted to publication.
- [13] Fasshauer G. F. (2007), "*Meshfree Approximation Methods with MATLAB*", World Scientific Publishing Co, Inc, River Edge, NJ, USA.
- [14] Fornberg B., Larsson E., Flyer N. (2011), "Stable computations with Gaussian radial basis functions", *SIAM J. Sci. Comput*, Volume 33(2), pp. 869-892.
- [15] Fornberg B. and Flyer N. (2015), "*A Primer on Radial Basis Functions with Applications to the Geosciences*", Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- [16] Liszka T. and Orkisz J. (1980), "The finite difference method at arbitrary irregular grids and its application in applied mechanics", *Comput. Struct.*, Volume 11, pp. 83-95.
- [17] Mitchell W. F. (2013), " A collection of 2D elliptic problems for testing adaptive grid refinement algorithms", *Applied Mathematics and Computation*, Volume 220, pp. 350 - 364.
- [18] Oanh D. T., Davydov O., Phu H. X. (2017), "Adaptive RBF-FD Method for Elliptic Problems with Point Singularities in 2D", *Applied Mathematics and Computation*, Volume 313, pp. 474-497.
- [19] Roosta H. (2000), "*Parallel Processing and Parallel Algorithms*", Springer-Verlag.
- [20] Tolstykh A. I. and Shirobokov D. A. (2003), "On using radial basis functions in a finite difference mode with applications to elasticity problems", *Computational Mechanics*, Volume 33, pp. 68-79.



- [21] Ureña M., Benito J. J., Ureña F., García Á., Gavete L., and Benito L., (2018), "Adaptive strategies to improve the application of the generalized finite differences method in 2d and 3d", *Mathematical Methods in the Applied Sciences*, Volume 41, No 17, pp.7115-7129.
- [22] Wendland H. (2005), "*Scattered Data Approximation*", Cambridge University Press.
- [23] Wright G. B, Fornberg B. (2006), "Scattered node compact finite difference-type formulas generated from radial basis functions", *Journal of Computational Physics*, Volume 212, No 1, p.99-123.
- [24] The MathWorks (2009), "*Partial Differential Equation Toolbox™ User's Guide*", Inc.