

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Phạm Quang Nhật Minh**

**ÁP DỤNG PHƯƠNG PHÁP SVD TÍNH LỰC XẤP  
XỈ TRONG BÀI TOÁN MÔ PHỎNG ĐỘNG LỰC  
PHÂN TỬ**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**HÀ NỘI – 2006**

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Phạm Quang Nhật Minh**

**ÁP DỤNG PHƯƠNG PHÁP SVD TÍNH LỰC XẤP  
XỈ TRONG BÀI TOÁN MÔ PHỎNG ĐỘNG LỰC  
PHÂN TỬ**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: TS Nguyễn Hải Châu**

**HÀ NỘI – 2006**

## TÓM TẮT KHÓA LUẬN

Mô phỏng động lực phân tử là một trong những phương pháp phổ biến để nghiên cứu các hệ vật lý và hóa học. Trong mô phỏng động lực phân tử, thời gian tính toán lực tương tác giữa các hạt trong hệ chiếm phần lớn tổng thời gian mô phỏng. Thuật toán khai triển đa cực nhanh Fast Multipole Method [5, 7, 8] và các cải tiến của nó là những phương pháp được sử dụng phổ biến trong mô phỏng động lực phân tử nhằm tăng tốc độ tính toán lực. Trong cài đặt thuật toán khai triển đa cực nhanh, phương pháp phân tích ma trận SVD (Singular Value Decomposition [17, 18]) được sử dụng để nhằm tăng độ chính xác của tính lực xấp xỉ. Một trong những vấn đề chưa được giải quyết trong cài đặt thuật toán khai triển đa cực nhanh là nghiên cứu ảnh hưởng của phương pháp SVD đến độ chính xác của tính lực xấp xỉ. Khóa luận sẽ nghiên cứu vấn đề nêu trên bằng thực nghiệm, nhằm tìm ra cách ứng dụng phương pháp SVD hợp lý để làm tăng độ chính xác và hiệu năng của thuật toán khai triển đa cực nhanh trên các máy tính chuyên dụng hoặc các máy tính thông thường. Các kết quả thu được trong khóa luận là khả quan và sẽ được ứng dụng trong các nghiên cứu về cài đặt thuật toán khai triển đa cực nhanh tiếp theo.

**Từ khóa:** Molecular Dynamics Simulation, Fast multipole method, Singular Value Decomposition, Pseudo-Particle Mutilpole Method.

## **LỜI CẢM ƠN**

Đầu tiên, em muốn gửi lời cảm ơn sâu sắc đến TS. Nguyễn Hải Châu, người đã hướng dẫn và chỉ bảo em tận tình trong suốt quá trình làm khóa luận. Cảm ơn thầy vì những định hướng, những tài liệu quý báu và những động viên, khích lệ, giúp em hoàn thành tốt khóa luận.

Em xin gửi lời cảm ơn tới TS Nguyễn Năng Tâm, giảng viên trường Đại học Sư phạm Hà Nội II vì những hỗ trợ về mặt toán học được sử dụng trong khóa luận.

Em xin bày tỏ lời cảm ơn sâu sắc đến các thầy cô giáo đã giảng dạy em trong bốn năm qua, những kiến thức mà em nhận được trên giảng đường Đại học sẽ giúp em vững bước trong tương lai.

Cuối cùng, tôi xin gửi lời cảm ơn sâu sắc tới những người thân trong gia đình, những người luôn quan tâm, động viên khích lệ tôi trong học tập và trong cuộc sống.

Sinh viên thực hiện khóa luận  
Phạm Quang Nhật Minh

## MỤC LỤC

TÓM TẮT KHÓA LUẬN.....	i
LỜI CẢM ƠN.....	ii
MỤC LỤC .....	iii
DANH MỤC HÌNH VẼ .....	v
DANH MỤC BẢNG BIỂU.....	vi
BẢNG THUẬT NGỮ .....	vii
MỞ ĐẦU .....	1
Chương 1. TỔNG QUAN VỀ BÀI TOÁN MÔ PHỎNG ĐỘNG LỰC PHÂN TỬ .....	3
1.1 Bài toán mô phỏng động lực phân tử .....	3
1.1.1 Giới thiệu chung .....	3
a. Các bước trong mô phỏng động lực phân tử .....	3
b. Ứng dụng của phương pháp mô phỏng động lực phân tử .....	4
1.1.2 Bài toán mô phỏng động lực phân tử dưới góc độ tính toán .....	4
1.2 Các phương pháp trong mô phỏng động lực phân tử .....	5
1.2.1 Phương pháp tính trực tiếp tương tác hạt-hạt .....	5
1.2.2 Thuật toán cây .....	6
1.2.3 Phương pháp khai triển đa cực nhanh .....	7
1.2.4 Một số phương pháp khác .....	7
1.3 Mục tiêu của khóa luận.....	8
1.4 Tổng kết chương.....	8
Chương 2. THUẬT TOÁN KHAI TRIỂN ĐA CỰC NHANH.....	9
2.1 Thuật toán khai triển đa cực nhanh FMM.....	9
2.1.1 Phương pháp khai triển đa cực .....	9
2.1.2 Thuật toán FMM.....	15
a. Các pha chính trong thuật toán FMM.....	16
b. Cài đặt thuật toán FMM .....	19
c. Độ phức tạp của thuật toán FMM.....	22
2.2 Các biến thể của thuật toán FMM .....	23
2.2.1 Phương pháp của Anderson.....	23
2.2.2 Phương pháp giả hạt của Makino .....	26
a. Trong hệ tọa độ 2 chiều .....	27
b. Trong hệ tọa độ 3 chiều .....	28
2.3 Tổng kết chương.....	30
Chương 3. ÁP DỤNG PHƯƠNG PHÁP SVD TRONG MÔ PHỎNG ĐỘNG LỰC PHÂN TỬ.....	31
3.1 Phương pháp SVD .....	31
3.1.1 SVD của ma trận vuông .....	32
3.1.2 Giải hệ phương trình tuyến tính .....	33
a. Cách giải hệ phương trình tuyến tính bằng SVD .....	33
b. Vấn đề chọn tham số “gần 0” trong phương pháp SVD .....	35
3.1.3 Cài đặt phương pháp SVD trên máy tính .....	35

3.2 Ứng dụng của phương pháp SVD trong inner $P^2M^2$ .....	36
3.2.1 Cài đặt thuật toán FMM trên máy GRAPE .....	36
a. Chức năng của máy GRAPE .....	36
b. Cài đặt thuật toán FMM trên máy GRAPE .....	37
3.2.2 Ứng dụng của SVD trong cài đặt inner $P^2M^2$ .....	38
3.3 Tổng kết chương .....	40
Chương 4. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ .....	41
4.1 Môi trường thực nghiệm .....	41
4.1.1 Phần cứng .....	41
4.1.2 Phần mềm .....	41
4.2 Thử nghiệm phương pháp khai triển đa cực nhanh FMM .....	41
4.2.1 Thời gian tính toán của phương pháp FMM .....	41
4.2.2 Đánh giá kết quả .....	43
4.3 Thử nghiệm phương pháp SVD trong biến đổi A2P .....	44
4.3.1 Độ chính xác của khai triển inner $P^2M^2$ và biến đổi A2P .....	44
a. Phương pháp thực nghiệm .....	44
b. Kết quả thực nghiệm .....	45
4.3.2 Ảnh hưởng của tham số gần không trong phương pháp SVD đến độ chính xác của thuật toán FMM .....	46
a. Phương pháp thực nghiệm .....	46
b. Kết quả thực nghiệm .....	47
4.4 Tổng kết chương .....	50
KẾT LUẬN .....	51
Kết quả đạt được .....	51
Hướng phát triển .....	51
Phụ lục A: Cài đặt SVD bằng ngôn ngữ C .....	53
A1. Thủ tục svdcmp() .....	53
A2. Thủ tục svbksb() .....	57
A3. Thủ tục zero_small_values() .....	57
TÀI LIỆU THAM KHẢO .....	58

## DANH MỤC HÌNH VẼ

Hình 1: Xấp xỉ trong cây (trên) và FMM (dưới) .....	6
Hình 2: Hai tập hợp hạt đủ xa trên mặt phẳng .....	12
Hình 3: Dịch chuyển tâm của khai triển đa cực. ....	14
Hình 4: Ý tưởng tính lực xấp xỉ trong FMM.....	16
Hình 5: Một vài mức phân chia trong FMM .....	17
Hình 6: Pha M2M trong thuật toán FMM .....	17
Hình 7: Danh sách hàng xóm và danh sách tương tác .....	18
Hình 8: Pha M2L trong thuật toán FMM .....	18
Hình 9: Pha L2L trong thuật toán FMM .....	19
Hình 10: Phương pháp của Anderson.....	25
Hình 11: Phương pháp giả hạt của Makino .....	26
Hình 12: Tính thế năng và lực từ phân phối khối lượng của các giả hạt .....	39
Hình 13: Thời gian tính lực của thuật toán trực tiếp (trên) và FMM (dưới) .....	43
Hình 14: Sai số trung bình bình phương của thế năng được tính bằng khai triển inner $P^2M^2$ và biến đổi A2P. Từ trên xuống, 8 đường cong tương ứng với các bậc khai triển $p = 1, 2, 3, 4, 5, 6, 7, 8$ .....	46
Hình 15: Sai số trung bình bình phương của lực được tính bằng khai triển inner $P^2M^2$ và biến đổi A2P. Từ trên xuống, 8 đường cong tương ứng với các bậc khai triển $p = 1, 2, 3, 4, 5, 6, 7, 8$ .....	46
Hình 16: Sai số trung bình bình phương của thế năng ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 1 đến 5 .....	48
Hình 17 : Sai số trung bình bình phương của thế năng ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 6 đến 10 .....	48
Hình 18: Sai số trung bình bình phương của lực ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 1 đến 5 .....	49
Hình 19 : Sai số trung bình bình phương của lực ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 6 đến 10 .....	49

## **DANH MỤC BẢNG BIỂU**

Bảng 1: Các phần mềm mô phỏng động lực phân tử tiêu biểu .....	4
Bảng 2: Phân tích độ phức tạp của thuật toán FMM.....	23
Bảng 3: Công cụ sử dụng trong thử nghiệm.....	41
Bảng 4: Thời gian tính toán của FMM với số hạt thay đổi .....	42
Bảng 5: Thời gian tính toán trực tiếp với số hạt thay đổi.....	42
Bảng 6: Tham số gần 0 ứng với các mức khai triển khác nhau .....	50



## **BẢNG THUẬT NGỮ**

<b>Từ hoặc cụm từ</b>	<b>Từ viết tắt</b>	<b>Tên tiếng Anh</b>
Bài toán giá trị biên		boundary value problem
Bước thời gian		Time step
Coulomb		Lực Coulomb
Danh sách tương tác		Interaction list
Danh sách hàng xóm		Neighbor list
Động lực phân tử	MD	Molecular Dynamics
Giả hạt		Pseudoparticle
Hạng	Rank	Rank
Khai triển đa cực		Multipole expansion
Phương pháp khai triển đa cực nhanh	FMM	Fast multipole method
Khai triển địa phương		Local expansion
Mô phỏng động lực phân tử	MD Simulation	Molecular Dynamics Simulation
Nghịch đảo ma trận		Matrix inversion
Nullspace		Nullspace
Range		Range
Số điều kiện		Condition Number
SVD	SVD	Singular value decomposition
Phương pháp giả hạt	$P^2M^2$	Pseudo-particle multipole method
Tương tác hạt-hạt	PP	Particle-Particle
Vật lý thiên văn		astrophysics

## MỞ ĐẦU

Sự phát triển nhanh chóng của công nghệ thông tin, đặc biệt là sự xuất hiện của các hệ thống siêu máy tính có tốc độ tính toán nhanh đã mở ra một phương pháp mới trong nghiên cứu khoa học, đó là phương pháp mô phỏng bằng máy tính. Mô phỏng bằng máy tính đóng vai trò như cầu nối giữa lý thuyết với thực hành, giữa các thí nghiệm thực tế với các thí nghiệm được thực hiện trên máy tính. Các lý thuyết có thể được kiểm định bằng các hệ mô phỏng, mặt khác tính chính xác của một hệ mô phỏng cũng có thể được kiểm định bằng các kết quả thí nghiệm thực tế. Hơn thế nữa, các thí nghiệm mà hiện nay con người chưa thể tiến hành được trong phòng thí nghiệm (ví dụ các thí nghiệm yêu cầu phải làm việc trong một môi trường nhiệt độ, hay áp suất rất cao) có thể được mô phỏng bằng máy tính. Như vậy có thể nói, mô phỏng bằng máy tính là một phương pháp có vai trò quan trọng, và ngày càng được sử dụng nhiều trong nghiên cứu khoa học.

Mô phỏng động lực phân tử là một phương pháp phổ biến để nghiên cứu các hệ vật lý và hóa học. Bài toán mô phỏng động lực phân tử xét dưới trên khía cạnh tính toán thực chất là bài toán tính toán tương tác giữa các hạt trong một hệ phân tử. Dễ thấy nếu sử dụng phương pháp tính toán trực tiếp tương tác của từng cặp hạt, độ phức tạp tính toán sẽ là  $O(N^2)$  với  $N$  là số hạt trong hệ. Như vậy đối các hệ có số hạt lớn (ví dụ vài triệu hạt) thì thời gian tính toán là lớn đến mức không thể chấp nhận được trong thực tế.

Đối với hầu hết các bài toán mô phỏng động lực phân tử, thời gian tính toán lực thường chiếm tới 95% tổng thời gian mô phỏng. Do đó đã có nhiều nghiên cứu nhằm làm giảm thời gian tính toán lực của bài toán mô phỏng. Các hướng nghiên cứu chính gồm có: Phát triển các thuật toán tính toán nhanh có độ phức tạp tính toán  $O(N \log N)$  hoặc  $O(N)$ , phát triển các phần cứng đặc biệt để tăng tốc độ tính lực và kết hợp hai hướng nghiên cứu trên.

Thuật toán khai triển đa cực nhanh [5, 7, 8] là thuật toán tính toán nhanh do Greengard và Rokhlin phát triển có độ phức tạp  $O(N)$ . Thuật toán khai triển đa cực nhanh (viết tắt FMM) được phát triển trên các máy tính thông thường nên không áp dụng được trên các máy tính đặc biệt. Do đó đã có nhiều cải tiến của thuật toán FMM như các cải tiến của Anderson [2], phương pháp “giả hạt” của Makino [16], L. Ying

[21, 22]. Các thuật toán này đã đơn giản hóa cài đặt của thuật toán FMM gốc và có thể áp dụng các cài đặt này trên các máy tính đặc biệt.

Dựa trên các nghiên cứu của Anderson và Makino, các tác giả Chau, Kawai, Ebisuzaki ([13]) đã cài đặt thuật toán FMM trên máy tính chuyên dụng GRAPE ([15, 20]) trong đó có sử dụng phương pháp SVD (Singular Value Decomposition [17, 18]) để tăng độ chính xác trong tính lực xấp xỉ. Một vấn đề chưa được giải quyết trong cài đặt thuật toán khai triển đa cực nhanh là nghiên cứu ảnh hưởng của phương pháp SVD đến độ chính xác của tính lực xấp xỉ. Vì vậy khóa luận sẽ nghiên cứu vấn đề chưa được giải quyết nêu trên nhằm làm tăng độ chính xác và hiệu năng của thuật toán khai triển đa cực nhanh trên máy tính chuyên dụng cũng như các máy tính thông dụng khác. Phương pháp nghiên cứu trong khóa luận là dựa trên thực nghiệm.

Ngoài phần mở đầu và kết luận, kết cấu của khóa luận bao gồm bốn chương:

- Chương 1 “**Tổng quan về bài toán mô phỏng động lực phân tử**” trình bày cơ bản về bài toán mô phỏng động lực phân tử và các phương pháp được sử dụng trong mô phỏng động lực phân tử.
- Chương 2 “**Thuật toán khai triển đa cực nhanh**” sẽ trình bày các vấn đề cơ bản về thuật toán khai triển đa cực nhanh và các biến thể của thuật toán.
- Chương 3 “**Áp dụng phương pháp SVD trong mô phỏng động lực phân tử**” trình bày về bài toán mà khóa luận đưa ra và đề xuất cách giải quyết dựa trên thực nghiệm.
- Chương 4 “**Kết quả thực nghiệm và đánh giá**” mô tả quá trình thực nghiệm, các bảng số liệu, đồ thị, và đưa ra đánh giá về kết quả thu được.

## Chương 1. TỔNG QUAN VỀ BÀI TOÁN MÔ PHỎNG ĐỘNG LỰC PHÂN TỬ

### 1.1 Bài toán mô phỏng động lực phân tử

#### 1.1.1 Giới thiệu chung

Động lực phân tử là sự mô phỏng hoạt động theo thời gian của một hệ phân tử. Phương pháp mô phỏng động lực phân tử dựa trên định luật 2 Newton về chuyển động,  $F = ma$ , trong đó  $F$  là lực tác dụng trên hạt,  $m$ ,  $a$  tương ứng là khối lượng và gia tốc của hạt. Từ các thông tin về lực tác dụng trên mỗi hạt, xác định gia tốc của mỗi hạt trong hệ. Lấy tích phân của phương trình chuyển động để sinh ra một đường cong mô tả vị trí, vận tốc, gia tốc của các hạt tại các mốc thời gian khác nhau. Từ đường cong này, trạng thái trong tiếp theo hay trạng thái trước của một hệ phân tử sẽ được dự báo.

#### a. Các bước trong mô phỏng động lực phân tử

Các bước chính trong mô phỏng động lực phân tử như sau:

1. Chọn vị trí ban đầu của các hạt trong hệ.
2. Chọn một tập hợp ban đầu vận tốc của các hạt. Các vận tốc này thường được chọn theo phân phối Boltzmann đối với một vài nhiệt độ, sau đó được chuẩn hóa sao cho tổng động lượng cuối cùng của toàn hệ bằng 0.
3. Tính động lượng của mỗi hạt từ vận tốc và khối lượng của chúng
4. Tính các lực tương tác trên mỗi hạt từ biểu thức năng lượng.
5. Tính vị trí mới cho các hạt sau một khoảng thời gian ngắn sau đó. Khoảng thời gian này được gọi là bước thời gian (time step). Việc tính toán này được thực hiện bằng tính tích phân số của các phương trình về chuyển động sử dụng các thông tin thu được trong bước trước.
6. Tính toán vận tốc và gia tốc mới cho các hạt trong hệ.
7. Lặp lại các bước từ bước 3 đến bước 6.
8. Lặp lại quá trình này đủ lâu để cho hệ đạt tới trạng thái cân bằng. Trong trường hợp này, trạng thái cân bằng không phải là cấu hình có năng lượng thấp nhất, nó là một cấu hình chấp nhận được cho hệ với năng lượng có độ lớn cho trước.

9. Mỗi khi hệ đạt tới trạng thái cân bằng, bắt đầu ghi lại tọa độ của các hạt sau một vài vòng lặp. Các thông tin này thường được ghi lại sau từ 5 đến 25 vòng lặp. Danh sách các tọa độ này tạo thành một quỹ đạo chuyển động của hệ hạt.
10. Tiếp tục quá trình lặp đi lặp lại và ghi lại dữ liệu cho đến khi có đủ dữ liệu được tập hợp để đưa ra các kết quả với độ chính xác mong muốn.
11. Phân tích các quỹ đạo chuyển động để thu được thông tin về hệ.

## **b. Ứng dụng của phương pháp mô phỏng động lực phân tử**

Mô phỏng động lực phân tử là một trong những phương pháp phổ biến để nghiên cứu các hệ vật lý và hóa học. Hiện nay phương pháp động lực phân tử được sử dụng nhiều trong việc nghiên cứu cấu trúc, động lực, nhiệt động lực của các phân tử sinh học. Trong vật lý, mô phỏng động lực phân tử được sử dụng để kiểm tra động lực học của các hiện tượng ở mức phân tử mà không thể quan sát được một cách trực tiếp. Nó cũng được dùng để kiểm tra các đặc tính vật lý của các thiết bị dùng công nghệ nano mà chưa hoặc không thể quan sát bằng các thiết bị hiện tại.

Hiện nay trên thế giới, đã có nhiều hệ phần mềm mô phỏng động lực phân tử. Bảng 1 liệt kê các hệ phần mềm mô phỏng động lực phân tử tiêu biểu hiện nay:

**Bảng 1: Các phần mềm mô phỏng động lực phân tử tiêu biểu**

STT	Tên hệ mô phỏng	Nguồn
1	AMBER	<a href="http://amber.scripps.edu/">http://amber.scripps.edu/</a>
2	CHARMM	<a href="http://www.charmm.org/">http://www.charmm.org/</a>
3	DL_POLY	<a href="http://www.cse.clrc.ac.uk/msi/software/DL_POLY/">http://www.cse.clrc.ac.uk/msi/software/DL_POLY/</a>
4	GROMOS	<a href="http://www.igc.ethz.ch/gromos-docs/index.html">http://www.igc.ethz.ch/gromos-docs/index.html</a>
5	GROMACS	<a href="http://www.gromacs.org/">http://www.gromacs.org/</a>
6	NAMD	<a href="http://www.ks.uiuc.edu/Research/namd/">http://www.ks.uiuc.edu/Research/namd/</a>
7	LAMMPS	<a href="http://www.cs.sandia.gov/%7Esjplimp/lammps.html">http://www.cs.sandia.gov/%7Esjplimp/lammps.html</a>
8	QUANTUM 3.1	<a href="http://www.q-pharm.com/home">http://www.q-pharm.com/home</a>

### **1.1.2 Bài toán mô phỏng động lực phân tử dưới góc độ tính toán**

Xét trên khía cạnh tính toán, bài toán mô phỏng động lực phân tử được mô tả như sau: Cho trước một hệ cô lập gồm  $N$  hạt phân bố trong một miền nào đó (Thông dụng nhất là miền hình lập phương trong không gian 3 chiều). Các hạt này tương tác với nhau theo lực Coulomb hoặc lực hấp dẫn. Bài toán mô phỏng động lực phân tử

ngiên cứu các hệ cô lập bằng cách cập nhật vị trí của các hạt sau một khoảng thời gian nhất định thể hiện qua  $m$  bước lặp. Tại mỗi bước lặp ta phải tính lực tác dụng tổng hợp lên mỗi hạt theo công thức tính lực Coulomb hoặc hấp dẫn. Áp dụng định luật 2 của Newton để tính được gia tốc rồi suy ra vận tốc của mỗi hạt. Sau đó tính toán lại vị trí mới của hạt dựa trên vận tốc của các hạt đó. Dễ dàng thấy rằng thuật toán đơn giản nhất để giải bài toán này là tính lực tương tác cho tất cả các cặp hạt trong hệ. Thuật toán này có độ phức tạp  $O(N^2)$ . Như vậy đối với các hệ có số hạt lớn (ví dụ vài triệu hạt) thì thời gian tính toán là lớn đến mức không chấp nhận được trong thực tế.

Đối với hầu hết các bài toán mô phỏng động lực phân tử, thời gian tính lực thường chiếm tới 95% thời tổng thời gian mô phỏng. Do đó đã có nhiều nghiên cứu nhằm làm giảm thời gian tính lực của bài toán mô phỏng. Các hướng nghiên cứu gồm có: Phát triển các thuật toán có độ phức tạp tính toán  $O(N \log N)$  hoặc  $O(N)$ , và phát triển các phần cứng đặc biệt để tăng tốc độ tính lực và kết hợp cả hai hướng nghiên cứu trên.

## **1.2 Các phương pháp trong mô phỏng động lực phân tử**

### **1.2.1 Phương pháp tính trực tiếp tương tác hạt-hạt**

Phương pháp tính toán trực tiếp tương tác hạt-hạt là phương pháp đơn giản nhất. Trong phương pháp này, đầu tiên ta tính các lực  $F(i, j)$  của hạt  $j$  tác động lên hạt  $i$ . Sau đó lấy tích phân phân của các phương trình chuyển động và cập nhật bộ đếm thời gian. Lặp lại quá trình này cho bước thời gian tiếp theo.

Ví dụ, trong mô phỏng lực hấp dẫn của một hệ gồm  $N$  hạt, một hạt có khối lượng  $M$  sẽ hút một hạt khác có khối lượng  $m$  với một lực :  $-(GMm/r^3).r$ , trong đó  $G$  là hằng số hấp dẫn,  $r$  là khoảng cách giữa hai hạt. Trong hệ có  $N$  hạt, do đó chúng ta phải tính lực  $(N-1)$  lần. Sau đó tách công thức tính lực thành hai phương trình vi phân bậc nhất của gia tốc và vận tốc. Cuối cùng, sử dụng một phương pháp giải phương trình vi phân số như Euler hay Runge-Kutta để thu được vị trí và vận tốc của các hạt.

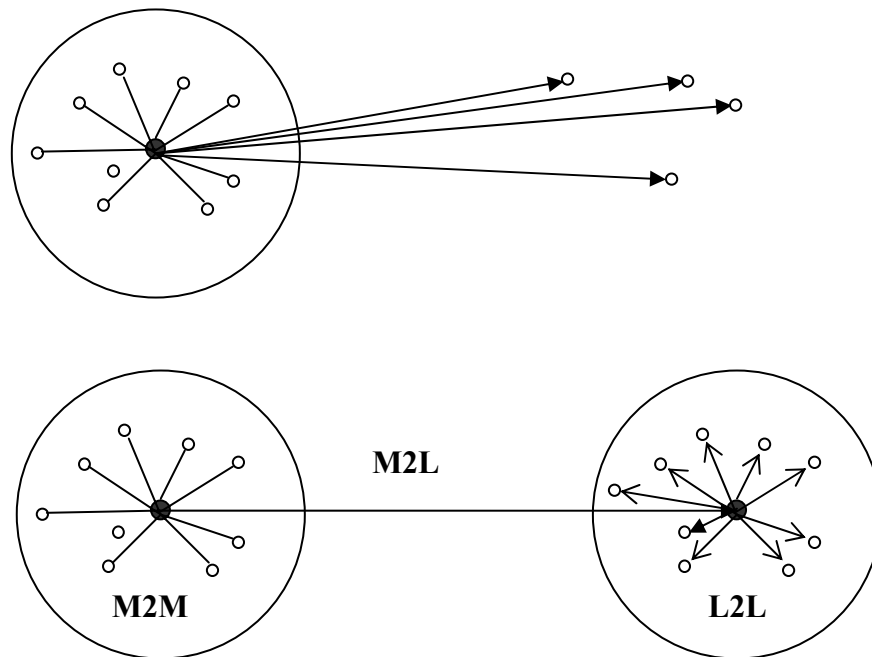
Phương pháp tính trực tiếp tương tác hạt-hạt là đơn giản và khá mềm dẻo nhưng có thể thấy rằng phương pháp này yêu cầu độ phức tạp tính toán cao:  $O(N^2)$  để tính toán lực cho tất cả  $N$  hạt. Với hệ có số lượng hạt  $N$  rất lớn, độ phức tạp này là không thể chấp nhận được. Vì vậy phương pháp này được áp dụng trong các hệ có số

lượng hạt nhỏ (Ít hơn  $N = 1000$  hạt) hoặc khi ta chỉ quan tâm đến các tương tác giữa các hạt trong một vùng nhất định.

### 1.2.2 Thuật toán cây

Ý tưởng cơ bản của thuật toán cây là thay thế các lực hấp dẫn (nếu trong trường hợp ta tính các tương tác hấp dẫn giữa các hạt trong hệ) từ một cụm các hạt ở xa bằng lực tác dụng gây ra từ trọng tâm của chúng, hoặc bằng khai triển đa cực (chi tiết về khai triển đa cực sẽ được nói rõ hơn trong phần sau của khóa luận) nếu với độ chính xác cao được yêu cầu. Các hạt được tổ chức thành một cấu trúc cây bát phân (octree structure) với gốc là toàn bộ hệ và các nút lá tương ứng với các hạt.

Lực tác dụng trên một hạt gây ra bởi một nút được định nghĩa (và được tính) một cách đệ quy. Nếu nút và hạt là *đủ xa nhau*, lực tác dụng từ nút đó tới hạt sẽ được tính bằng khai triển đa cực của nút đó tại vị trí của hạt. Nếu chúng không đủ xa nhau, lực tương tác sẽ được tính bằng cách lấy tổng các lực tác dụng từ các nút con của nút đó. Để tính lực tác dụng lên một hạt, độ phức tạp sẽ là  $O(\log N)$ , vì độ phức tạp là tỉ lệ thuận với chiều cao của cây.



**Hình 1: Xấp xỉ trong cây (trên) và FMM (dưới)**

Các khai triển đa cực của các nút trong cây phải được tính trước. Hệ số khai triển cho một nút có thể được tính đệ quy từ các nút con của nó. Độ phức tạp tính toán trong

phần này là  $O(N)$ . Như vậy độ phức tạp để tính lực tác dụng lên các hạt trong hệ sẽ là  $O(N \log N)$

Chi tiết hơn về cài đặt thuật toán có trong tài liệu tham khảo [4].

### **1.2.3 Phương pháp khai triển đa cực nhanh**

Thuật toán khai triển đa cực nhanh FMM được đưa ra lần đầu tiên bởi Greengard và Rokhlin [7, 8] để tăng tốc độ tính lực trong bài toán mô phỏng động lực phân tử. Trong thuật toán FMM, lực được tính xấp xỉ bằng khai triển đa cực. Sau đó khai triển đa cực này sẽ được chuyển thành khai triển cực bộ tại điểm quan sát. Lực tại mỗi điểm sẽ được tính nhờ khai triển cực bộ. Trên hình 1 chúng ta có thể thấy được sự khác nhau giữa hai thuật toán cây và thuật toán FMM.

Độ phức tạp tính toán trong thuật toán này là  $O(N)$ . Tuy nhiên với cùng một độ chính xác, thời gian tính toán thực tế của thuật toán FMM cổ điển là cao hơn so với thuật toán cây khi số lượng hạt chưa đủ lớn. Như vậy FMM cổ điển chỉ thực sự hiệu quả hơn khi số lượng hạt  $N$  là rất lớn.

Có nhiều cải tiến của thuật toán FMM nhằm làm giảm thời gian tính toán cũng như cách cài đặt của nó, điển hình là các cải tiến của Anderson [2] và của Makino [16]. Ý tưởng chính của các cải tiến này là giữ nguyên mô hình, các pha của thuật toán FMM nhưng thay thế việc sử dụng các khai triển đa cực phức tạp bằng các công thức đơn giản hơn (Anderson) hay sử dụng phân phối các giả hạt trên mặt cầu để thể hiện các hệ số của khai triển đa cực (Makino). Các cải tiến này đã giúp cho việc cài đặt của thuật toán FMM trên máy tính là đơn giản hơn so với thuật toán FMM cổ điển.

### **1.2.4 Một số phương pháp khác**

Các phương pháp tính toán trong mô phỏng động lực phân tử là rất nhiều nên việc trình bày chi tiết hết các phương pháp này vượt khỏi khuôn khổ của một luận văn tốt nghiệp đại học. Ở đây chúng tôi liệt kê một số phương pháp khác ngoài các phương pháp đã trình bày ở trên: Phương pháp Particle-Mesh (PM), phương pháp Particle-Particle/Particle-Mesh (P3M), Particle Multiple-Mesh (PM2), Nested Grid Particle-Mesh (NGPM), Tree-code (Top down) và Tree-code (Bottom up), Tree-code Particle Mesh (TPM), Self-Consistent Field (SCF), phương pháp Symplectic.



### **1.3 Mục tiêu của khóa luận**

Trong biến thể của phương pháp FMM, Makino dùng các giả hạt trên một đường tròn (mặt cầu trong không gian ba chiều) thay cho các hệ số trong khai triển đa cực, để tính được phân phối khối lượng (điện tích) của các giả hạt này, cần phải dùng đến các phương pháp giải hệ phương trình tuyến tính và các thao tác tính nghịch đảo ma trận. Một giải pháp cho vấn đề này là ứng dụng phương pháp SVD (Singular Value Decomposition [17, 18]) trong tính lực để làm tăng độ chính xác của thuật toán. Nhưng vẫn còn một vấn đề chưa được nghiên cứu trong cài đặt thuật toán khai triển đa cực nhanh đó là nghiên cứu ảnh hưởng phương pháp SVD đến độ chính xác của tính lực xấp xỉ. Khóa luận sẽ nghiên cứu vấn đề chưa được giải quyết nói trên, nhằm làm tăng độ chính xác và hiệu năng của thuật toán khai triển đa cực nhanh. Mặt khác khóa luận cũng trình bày các kết quả thử nghiệm hiệu năng của thuật toán FMM và phương pháp tính lực trực tiếp.

### **1.4 Tổng kết chương**

Trong chương đầu tiên của khóa luận, chúng ta đã có một cái nhìn tổng quan về bài toán mô phỏng động lực phân tử, các bước trong mô phỏng động lực phân tử, các cách tiếp cận để tăng tốc độ tính lực trong bài toán cũng như các vấn đề chưa được giải quyết trong bài toán này. Chương 2 của khóa luận sẽ trình bày kỹ hơn về phương pháp khai triển đa cực nhanh và các cải tiến của nó mà đang được dùng rộng rãi trong mô phỏng động lực phân tử.

## Chương 2. THUẬT TOÁN KHAI TRIỂN ĐA CỰC NHANH

### 2.1 Thuật toán khai triển đa cực nhanh FMM

#### 2.1.1 Phương pháp khai triển đa cực

Trong bài toán mô phỏng động lực phân tử, sau mỗi bước thời gian, chúng ta cần phải cập nhật lại vị trí, vận tốc, và gia tốc của các hạt. Để tính được các giá trị này, chúng ta cần tính lực tương tác trên mỗi hạt. Như chúng ta đã biết trong vật lý, lực tương tác trên mỗi hạt trong một hệ bao gồm  $N$  hạt được tính theo công thức:

$$m_i \frac{d^2 x_i}{dt^2} = -\nabla_i \Phi \text{ với } i = 1, \dots, N$$

ở đây  $m_i$  là khối lượng của hạt thứ  $i$ . Lực thu được từ gradient của hàm thế năng  $\Phi$ .

Bài toán mô phỏng động lực phân tử được giới thiệu trong khóa luận này giới hạn trong trường hợp thế năng (hoặc lực) tại một điểm là tổng của các tương tác từng đôi một. Với cách tiếp cận tính toán thông thường, thuật toán tính tất cả các tương tác giữa các cặp hạt có độ phức tạp tính toán là  $O(N^2)$ . Với số lượng hạt  $N$  trong mô phỏng là rất lớn, độ phức tạp này là khó có thể chấp nhận.

Để rõ ràng hơn, chúng ta biểu diễn thế năng tại một điểm dưới dạng:

$$\Phi = \Phi_{far} + \Phi_{near}$$

với  $\Phi_{near}$  là thế năng gây ra bởi các hạt “gần” và  $\Phi_{far}$  là thế năng gây ra bởi các hạt ở xa. Thế năng gây ra bởi các hạt ở xa có ảnh hưởng nhỏ hơn rất nhiều so với thế năng gây ra bởi các hạt ở gần. Do đó trong mô phỏng động lực phân tử, để giảm khối lượng tính toán trên máy tính, chúng ta có thể xấp xỉ thế năng gây ra bởi các hạt ở xa này với một sai số có thể chấp nhận được. Phương pháp thường sử dụng trong trường hợp này là phương pháp khai triển đa cực.

Ý tưởng cơ bản của phương pháp khai triển đa cực là các hạt ở “xa” được phân cụm lại. Việc tính toán thế năng gây ra bởi các hạt ở xa này được xem như là tính tương tác với một hạt nhằm tăng tốc độ tính toán.

Để minh họa phương pháp khai triển đa cực, chúng ta xem xét một mô hình vật lý hai chiều bao gồm một tập hợp gồm  $N$  hạt tích điện (thuật toán trong không gian 3 chiều là tương tự về mặt ý tưởng nhưng công thức phức tạp hơn nên ở đây chúng tôi trình bày trong trường hợp 2 chiều). Thế năng và lực tại các hạt được tính từ định luật Coulomb. Giả sử rằng một điện tích điểm với cường độ đơn vị được đặt tại điểm  $(x_0, y_0) = \mathbf{x}_0 \in \mathbb{R}^2$ . Thế thì với mọi  $\mathbf{x} = (x, y) \in \mathbb{R}^2$  mà  $\mathbf{x} \neq \mathbf{x}_0$ , trường thế năng và điện trường gây ra do điện tích này được tính bằng công thức:

$$\Phi_{x_0}(x, y) = -\log(\|\mathbf{x} - \mathbf{x}_0\|)$$

và

$$E_{x_0}(x, y) = \frac{(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{x} - \mathbf{x}_0\|}$$

$\Phi_{x_0}$  là hàm điều hòa trong mọi khoảng mà không chứa điểm  $x_0$ . Hơn nữa, với mọi hàm điều hòa  $u$ , tồn tại một hàm khả vi  $w: C \rightarrow C$  sao cho  $u(x, y) = \operatorname{Re}(w(x, y))$  và  $w$  là duy nhất. Đặt  $x + iy = z \in C$ , công thức sẽ trở thành:

$$\Phi_{x_0}(x) = \operatorname{Re}(-\log(z - z_0))$$

Trong các phần tiếp theo chúng ta sẽ tiếp tục sử dụng kí hiệu hàm phức để biểu diễn thế năng gây ra bởi các hạt trong hệ.

Về cơ sở vật lý và toán học của phương pháp khai triển đa cực, chúng ta xem xét các bổ đề sau đây.

**Bổ đề 2.1:** Nếu  $u(x, y) = \operatorname{Re}(w(x, y))$  mô tả trường thế năng tại  $(x, y)$  thì lực tương ứng được cho bởi:

$$\nabla u = (u_x, u_y) = (\operatorname{Re}(w'), -\operatorname{Im}(w')),$$

ở đây  $w'$  là đạo hàm của  $w$

Bổ đề (2.1) ở trên là hệ quả trực tiếp của các phương trình Cauchy -Riemann.

Bổ đề sau đây được sử dụng để đạt được khai triển đa cực đối với trường thế năng gây ra bởi  $m$  điện tích

**Bổ đề 2.2.** Cho một điện tích điểm với cường độ  $q$  đặt tại  $z_0$ . Với mọi  $z$  sao cho  $|z| > |z_0|$ ,

$$\Phi_{z_0}(z) = q \log(z - z_0) = q \left( \log(z) - \sum_{k=1}^{\infty} \frac{1}{k} \left( \frac{z_0}{z} \right)^k \right) \quad (2.1)$$

**Chứng minh:** Chú ý rằng  $\log(z - z_0) - \log(z) = \log(1 - z_0/z)$  và  $|z_0/z| < 1$ . Bỏ đề đạt được từ khai triển:

$$\log(1 - w) = (-1) \sum_{k=1}^{\infty} \frac{w^k}{k},$$

Khai triển này đúng với mọi  $w$  thỏa mãn  $|w| < 1$  (Điều phải chứng minh)

**Định lý 2.1** (Khai triển đa cực). Giả sử rằng  $m$  điện tích với các cường độ  $\{q_i, i=1, \dots, m\}$  được đặt tại các điểm  $\{z_i, i=1, \dots, m\}$ , với  $|z_i| < r$ . Với mọi  $z \in C$  với  $|z| > r$ , thế năng  $\Phi(z)$  được cho bởi

$$\Phi(z) = Q \log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k} \quad (2.2)$$

ở đây:

$$Q = \sum_{i=1}^m q_i, \quad a_k = \sum_{i=1}^m \frac{-q_i z_i^k}{k} \quad (2.3)$$

Hơn nữa, với mọi  $p \geq 1$

$$\left| \Phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| \leq \alpha \left| \frac{z}{r} \right|^{p+1} \leq \left( \frac{A}{c-1} \right) \left( \frac{1}{c} \right)^p \quad (2.4)$$

trong đó:

$$c = \left| \frac{z}{r} \right|, \quad A = \sum_{i=1}^m |q_i|, \quad \text{và} \quad \alpha = \frac{A}{1 - |r/z|} \quad (2.5)$$

### Chứng minh

Dạng của khai triển đa cực (2.2) là một hệ quả trực tiếp của các bổ đề 2.1 với chú ý là  $\Phi(z) = \sum_{i=1}^m \Phi_{z_i}(z)$ . Để đạt được (2.4), chú ý rằng

$$\left| \Phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| = \left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right|$$

Thay thế  $a_k$  ở công thức (2.3) chúng ta có

$$\left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right| \leq A \sum_{k=p+1}^{\infty} \frac{r^k}{k |z|^k} \leq A \sum_{k=p+1}^{\infty} \left| \frac{r}{z} \right|^{p+1} = \left( \frac{A}{c-1} \right) \left( \frac{1}{c} \right)^p$$

Đặc biệt với  $c \geq 2$  thì

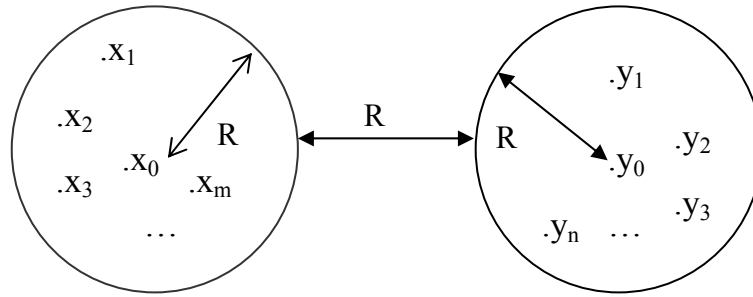
$$\left| \Phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| \leq A \left( \frac{1}{2} \right)^k \quad (2.6)$$

Chúng tôi sẽ minh họa, với một ví dụ đơn giản, cách một khai triển đa cực được dùng để tăng tốc độ tính toán thế năng. Giả sử rằng các điện tích với cường độ  $q_1, q_2, \dots, q_m$  được đặt tại các điểm  $x_1, x_2, \dots, x_m \in C$  và  $\{y_1, y_2, \dots, y_n\}$  là một tập các điểm khác trong mặt phẳng phức  $C$  (hình 2). Chúng ta nói rằng tập hợp  $\{x_i\}$  và  $\{y_i\}$  là “đủ xa” nhau nếu tồn tại các điểm  $x_0, y_0 \in C$  và một số thực  $r > 0$  sao cho

$$|x_i - x_0| < r \text{ với } i = 1, \dots, m,$$

$$|y_i - y_0| < r \text{ với } i = 1, \dots, n,$$

$$|x_0 - y_0| > 3r$$



**Hình 2: Hai tập hợp hạt đủ xa trên mặt phẳng**

Để đạt được thế năng (hoặc lực) tại các điểm  $\{y_j\}$  gây ra bởi các điểm  $\{x_i\}$  một cách trực tiếp, chúng ta có thể tính

$$\sum_{i=1}^m \Phi_{x_i}(y_j) \text{ với } j=1, \dots, n \quad (2.7)$$

Rõ ràng việc tính toán trên yêu cầu độ phức tạp bậc  $nm$  (Tính  $m$  thế năng tại  $n$  điểm). Bây giờ giả sử rằng đầu tiên chúng ta tính các hệ số của khai triển đa cực đến cấp  $p$  ( $p$ -term) của thế năng gây ra bởi các điện tích  $q_1, q_2, \dots, q_m$  xung quanh điểm  $x_0$  (sử dụng định lý 2.1). Nó yêu cầu một số lượng các phép toán tỷ lệ với  $mp$ . Việc tính khai triển đa cực tại tất cả các điểm  $y_j$  từ khai triển trên có độ phức tạp tính toán là  $np$ , như vậy độ phức tạp tính toán tổng cộng là  $O(mp + np)$ . Hơn nữa

$$\left| \sum_{i=1}^m \Phi_{x_i}(y_j) - Q \log(y_j - x_0) - \sum_{k=1}^p \frac{a_k}{|y_j - x_0|^k} \right| \leq A \left( \frac{1}{2} \right)^p$$

Để đạt được một độ chính xác tương đối  $\varepsilon$ , bậc  $p$  phải bằng  $-\log_2(\varepsilon)$ . Như vậy khi độ chính xác là xác định, do  $p$  là hằng số nên độ phức tạp tính toán giảm xuống thành

$$O(m) + O(n)$$

Độ phức tạp trên là nhỏ hơn rất nhiều so với  $nm$  khi  $n, m$  lớn.

### Phép dịch chuyển và cận trên sai số

Trước khi tìm hiểu thuật toán khai triển đa cực nhanh FMM, chúng ta công nhận ba bổ đề dưới đây. Ba bổ đề này là những công cụ giải tích để biến đổi khai triển đa cực theo cách thức phù hợp với thuật toán FMM. Vì lý do dễ hiểu, các bổ đề này được trình bày trong hệ tọa độ 2 chiều. Trong trường hợp 3 chiều, ý tưởng của các bổ đề này là tương tự, chỉ khác nhau về công thức biểu diễn. Bổ đề (2.3) đưa ra một công thức dịch chuyển tâm của khai triển đa cực. Bổ đề (2.4) mô tả cách chuyển đổi một khai triển như thế thành khai triển cục bộ (khai triển Taylor) trong một miền có dạng hình tròn, và bổ đề (2.5) cung cấp một kỹ thuật cho việc dịch chuyển tâm của khai triển Taylor. Các bổ đề này cũng đưa ra các giới hạn về sai số giúp chúng ta có thể tính xấp xỉ với bất kỳ độ chính xác cho trước.

#### **Bổ đề 2.3** Giả sử rằng

$$\Phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \quad (2.8)$$

Là khai triển đa cực của thế năng gây ra bởi một tập  $m$  điện tích với cường độ là  $q_1, q_2, \dots, q_m$ , tất cả các điện tích này nằm trong một đường tròn  $D$  với bán kính  $R$  và có tâm tại  $z_0$ . Thế thì đối với  $z$  nằm ngoài đường tròn  $D_1$  bán kính  $(R + |z_0|)$  và có tâm tại gốc tọa độ, ta có

$$\Phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l} \quad (2.9)$$

ở đây

$$b_l = \left( \sum_{k=1}^l a_k z_0^{l-k} \binom{l-1}{k-1} \right) - \frac{a_0 z_0^l}{l} \quad (2.10)$$

Với  $\binom{l}{k}$  là hệ số khai triển nhị thức. Hơn nữa với mọi  $p \geq 1$

$$\left| \Phi(z) - a_0 \log(z) - \sum_{l=1}^p \frac{b_l}{z^l} \right| \leq \left( \frac{A}{1 - \frac{|z_0| + R}{|z|}} \right) \left| \frac{|z_0| + R}{z} \right|^{p+1} \quad (2.11)$$

$A$  được định nghĩa như trong (2.5)

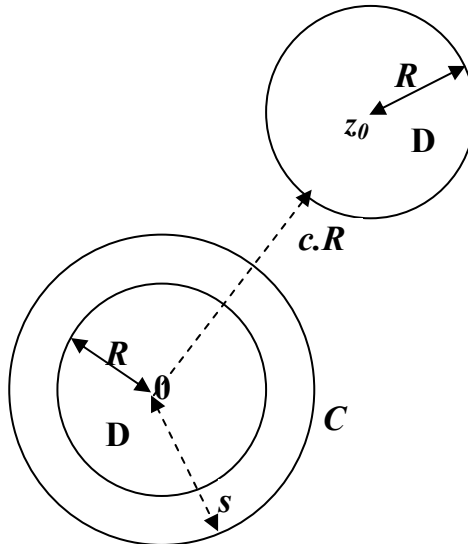
**Chú ý:** Khi các giá trị  $\{a_0, a_1, \dots, a_p\}$  trong khai triển (2.8) xung quanh  $z_0$  được tính, chúng ta có thể thu được  $\{b_1, \dots, b_p\}$  một cách chính xác bằng (2.10). Mặt khác chúng ta có thể dịch chuyển tâm của khai triển đa cực đã được cắt cụt mà không làm giảm độ chính xác.

#### **Bổ đề 2.4**

Giả sử rằng  $m$  điện tích với độ lớn  $q_1, q_2, \dots, q_m$  nằm bên trong đường tròn  $D_1$  với bán kính  $R$  và tâm  $z_0$ , và rằng  $|z_0| > (c+1)R$  với  $c > 1$  (hình 3). Thế thì khai triển đa cực tương ứng (2.8) hội tụ bên trong đường tròn  $D_2$  bán kính  $R$  với tâm ở gốc tọa độ. Bên trong  $D_2$ , thế năng gây ra bởi các điện tích được biểu diễn bằng chuỗi lũy thừa

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \quad (2.12)$$

ở đây



**Hình 3: Dịch chuyển tâm của khai triển đa cực.**

$$b_0 = \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k + a_0 \log(-z_0) \quad (2.13)$$

và

$$b_l = \left( \frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k \right) - \frac{a_0}{l \cdot z_0^l} \text{ với } l \geq 1. \quad (2.14)$$

Hơn nữa, với mọi  $p \geq \max(2, 2c/(c-1))$ , một cận trên của sai số cho chuỗi rút gọn được cho bởi

$$\left| \phi(z) - \sum_{l=0}^p b_l \cdot z^l \right| < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} \left( \frac{1}{c} \right)^{p+1}, \quad (2.15)$$

ở đây  $A$  đã được định nghĩa như trong (2.5) và  $e$  là cơ sở của logarit cơ số tự nhiên.

Bổ đề (2.5) sau đây là một hệ quả trực tiếp của định lý Macloranh. Nó biểu diễn chính xác thao tác dịch chuyển tâm với bậc khai triển hữu hạn, và không yêu cầu giới hạn nào về sai số.

**Bổ đề 2.5** Cho bất kỳ số phức  $z_0, z$  và  $\{a_k\}, k=0, 1, 2, \dots, n$ ,

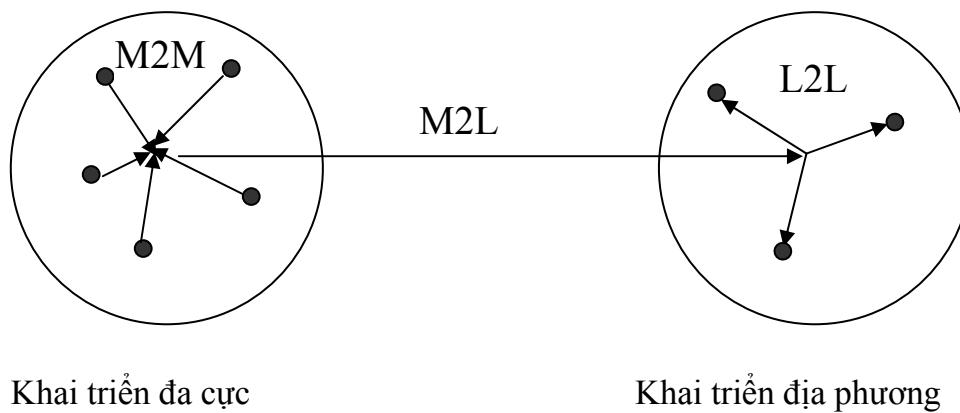
$$\sum_{k=0}^n a_k (z - z_0)^k = \sum_{l=0}^n \left( \sum_{k=l}^n a_k \binom{k}{l} (-z_0)^{k-l} \right) z^l \quad (2.16)$$

### 2.1.2 Thuật toán FMM

Trong phần 2.1.1, chúng ta đã xem xét phương pháp khai triển đa cực để tính xấp xỉ thế năng. Có nhiều thuật toán sử dụng phương pháp khai triển đa cực như phương pháp cây do Appel [3], Barnes và Hut [4] phát triển với độ phức tạp  $O(N \log N)$ , phương pháp “Pannel clustering” của Hackbush và Nowak ([9]) với độ phức tạp  $O(N(\log N)^{d+2})$ , phương pháp của Beylkin, Coifman, Rokhlin [6] với độ phức tạp  $O(N \log N)$ , khai triển đa cực nhanh do Greengard và Rokhlin ([7, 8]) phát triển có độ phức tạp  $O(N)$ . Trong các phương pháp sử dụng khai triển đa cực, phương pháp khai triển đa cực nhanh là một phương pháp được ứng dụng rộng rãi trong mô phỏng động lực phân tử.



FMM là một thuật toán xấp xỉ để tính thế năng (lực) giữa các hạt. Trong trường hợp các hạt trong hệ có phân phối chuẩn, thuật toán FMM sẽ giảm độ phức tạp tính toán từ  $O(N^2)$  xuống  $O(N)$ . Chiến lược trung tâm được sử dụng là chiến lược phân cụm các hạt tại các độ dài không gian khác nhau và tính tương tác với các cụm khác mà đủ xa theo nghĩa khai triển đa cực. Các tương tác giữa các hạt gần nhau được tính trực tiếp. Tương tác với các hạt ở xa nhau sẽ được tính thông qua tương tác cụm-cụm. Hình vẽ sau mô tả ý tưởng cơ bản của thuật toán FMM và các pha của trong thuật toán.



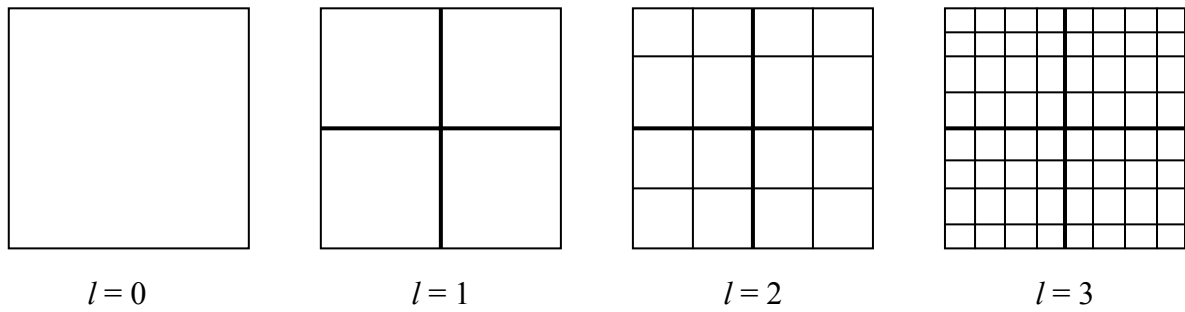
Hình 4: Ý tưởng tính lực xấp xỉ trong FMM

#### a. Các pha chính trong thuật toán FMM

Thuật toán đầu tiên được trình bày trong trường hợp 2 chiều [7] sau đó được mở rộng đối với trường hợp 3 chiều [8]. Trong khóa luận này, chúng ta sẽ tìm hiểu cách cài đặt thuật toán đối với trường hợp 2 chiều. Thuật toán được cài đặt theo các “pha” chính sau đây:

##### i. Tạo cây

Ban đầu chúng ta định nghĩa một hình vuông đủ lớn (Nút gốc) chứa tất cả các hạt trong hệ. Chúng ta tạo một cây tứ phân (quadtree) bằng cách chia nhỏ dần hình vuông theo các cấp. Thủ tục phân chia bắt đầu từ nút gốc tại mức  $l = 0$  bao gồm toàn bộ hệ. Mức ở đây chính là độ sâu của cây mà chúng ta tạo ra. Việc phân chia được thực hiện đệ quy cho tất cả các nút con, và sẽ dừng lại cho tới mức  $l_{\max}$ . Mức  $l_{\max}$  được chọn sao cho số lượng hạt trung bình trong các nút là bằng với một số lượng được xác định trước nào đó (có thể có sai số cho phép) để tối ưu tốc độ tính toán.

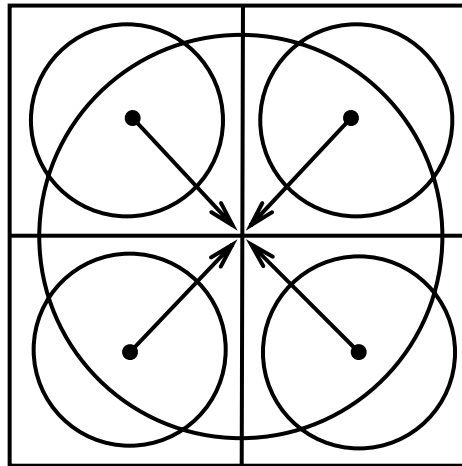


**Hình 5: Một vài mức phân chia trong FMM**

## ii. Biến đổi M2M

Tiếp theo chúng ta tính toán khai triển đa cực cho mỗi nút lá với các hạt nằm trong nút đó theo định lý 2.1.

Sau đó, chúng ta tính toán khai triển đa cực cho tất cả các nút không phải là nút lá ở tất cả các mức. Việc tính toán bắt đầu từ nút cha của các nút. Đối với mỗi nút, các khai triển đa cực của các nút con của nó sẽ được dịch chuyển tới tâm hình học của nút đó (Biến đổi M2M dùng bổ đề 2.3) và sau đó lấy tổng của các khai triển này. Thủ tục này tiếp tục cho đến khi lên tới nút gốc.



**Hình 6: Pha M2M trong thuật toán FMM**

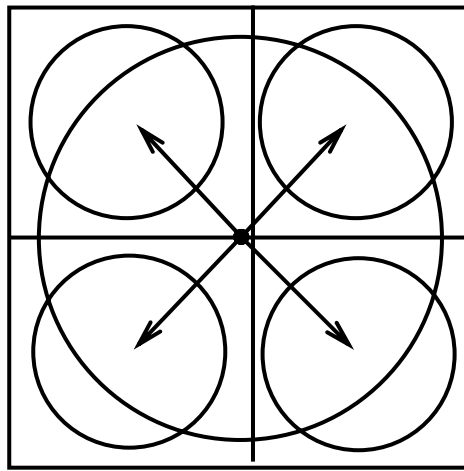
## iii. Biến đổi M2L

Trước hết chúng ta định nghĩa hai thuật ngữ “danh sách hàng xóm” (neighbor list) và “danh sách tương tác” (interaction list). Danh sách hàng xóm của một nút là tập hợp của các nút trong cùng một mức mà kề với nút đang xét. Với một nút, ta xét các hàng xóm của nút cha. Tập hợp các nút con của các hàng xóm này mà không kề với



#### iv. Biến đổi L2L

Trong bước tiếp theo, chúng ta sẽ duyệt cây theo thứ tự trên xuống. Lấy tổng của các khai triển địa phương (Khai triển Taylor) tại các mức khác nhau để đạt được tổng thể năng tại các nút lá. Đối với mỗi nút ở mức  $l$ , chúng ta dịch chuyển tâm của khai triển đa cực của nút cha của nút đó ở mức  $l-1$  (Biến đổi L2L), sau đó cộng khai triển đó vào khai triển địa phương của nút đang xét. Bằng thủ tục này, tất cả các nút ở mức  $l$  sẽ có khai triển địa phương của thể năng gây ra bởi các hạt ở “xa” (các hạt ở xa là các hạt không nằm trong danh sách hàng xóm của hạt đang xét). Lặp lại thủ tục này cho tất cả các mức, chúng ta thu được trường thể năng của tất cả các nút lá.



Hình 9: Pha L2L trong thuật toán FMM

#### v. Tính lực

Bước cuối cùng trong thuật toán này là tính lực tác dụng lên mỗi hạt trong tất cả các nút lá. Như ta đã biết, lực tương tác sẽ thu được bằng gradient của thể năng. Thể năng tổng cộng tại vị trí các hạt sẽ được tính bằng tổng của thể năng gây ra do các hạt đủ xa đã được tính trong các bước trên và thể năng gây ra bởi các hạt nằm trong khoảng cách gần với vị trí đang xét. Thể năng “gần” sẽ được tính trực tiếp qua tương tác từng đôi giữa các hạt.

#### b. Cài đặt thuật toán FMM

Các kí hiệu sau đây được dùng trong mô tả của cài đặt thuật toán FMM:

$\Phi_{li}$ : Khai triển đa cực cấp  $p$  (Xung quanh tâm của nút  $i$ ) của trường thể năng tạo ra bởi các hạt nằm bên trong nút  $i$  tại mức  $l$

$\Psi_{l,i}$ : Khai triển cấp  $p$  xung quanh tâm của nút  $i$  tại mức  $l$ , mô tả trường thế năng gây ra bởi tất cả các hạt bên ngoài nút và bên ngoài các hàng xóm của nó.

$\tilde{\Psi}_{l,i}$ : khai triển địa phương (khai triển Taylor) xung quanh tâm của nút  $i$  tại mức  $l$ , mô tả trường thế năng gây ra bởi tất cả các hạt bên ngoài nút cha nút  $i$  và bên ngoài các hàng xóm gần nhất của nút cha của  $i$ .

Giả sử rằng tại mức  $l-1$ , khai triển địa phương  $\Psi_{l-1,i}$  đã được tính cho tất cả các nút. Sử dụng bổ đề (2.5) để dịch chuyển (đối với mọi  $i$ ) khai triển  $\Psi_{l-1,i}$  tới khai triển của các nút con của  $i$ , chúng ta sẽ thu được: với mỗi nút  $j$  tại mức  $l$ , một khai triển địa phương của trường thế năng gây ra bởi các hạt bên ngoài các hàng xóm của cha mẹ  $j$  gọi là  $\tilde{\Psi}_{l,j}$ . Sau đó cộng thế năng gây ra bởi các nút trong danh sách tương tác của  $j$  với  $\tilde{\Psi}_{l,j}$  để sinh ra  $\Psi_{l,j}$ . Thế năng gây ra bởi các nút trong danh sách tương tác của  $j$  nhờ bổ đề (2.4) chuyển khai triển đa cực của các nút trong danh sách tương tác này thành khai triển địa phương tại tâm của nút hiện tại. Cũng chú ý rằng,  $\Psi_{0,i}$  và  $\Psi_{1,i}$  là bằng 0 bởi vì không có các nút nào đủ xa nhau được xem xét và chúng ta có thể bắt đầu hình thành khai triển địa phương ở mức 2.

Sau đây chúng ta sẽ mô tả thuật toán dưới dạng giả mã cùng với một số giải thích.

### **Khởi tạo**

Chọn một mức phân chia  $n \approx \log_4 N$ , một độ chính xác  $\varepsilon$  và chọn cấp khai triển  $p \approx -\log_2 \varepsilon$ .

### **Pha đi lên**

#### **Bước 1**

Giải thích [*Tính khai triển đa cực của trường thế năng gây ra bởi các hạt bên trong mỗi nút xung quanh tâm của nút tại các nút lá*]

do  $ibox=1, \dots, 4^n$

Tạo ra khai triển đa cực cấp  $p$  của  $\Phi_{n,ibox}$ , bằng cách sử dụng định lý 2.1

enddo

#### **Bước 2**

Giải thích [*Tính khai triển đa cực xung quanh các tâm của tất cả các nút ở các mức cao hơn, mỗi khai triển biểu diễn trường thế năng gây ra bởi tất cả các hạt chứa bên trong 1 nút*]

do  $l = n-1, \dots, 0$

do  $ibox = 1, \dots, 4^l$

Tính khai triển đa cực cấp  $p$   $\Phi_{l,ibox}$  bằng cách sử dụng định lý (2.3) để dịch chuyển tâm của mỗi khai triển của nút con tới nút hiện tại và lấy tổng của chúng.

enddo

enddo

### Pha đi xuống

#### Bước 3

Giải thích [*Tính khai triển địa phương tại tâm của mỗi nút trong lưới tại mỗi mức  $l \leq n$ . Khai triển cục bộ này mô tả thế năng gây ra tại tâm của nút đang xét bởi tất cả các hạt trong hệ mà không nằm trong nút hiện tại hoặc các hộp hàng xóm gần nhất của nó*]

Thiết lập  $\tilde{\Psi}_{1,1} = \tilde{\Psi}_{1,2} = \tilde{\Psi}_{1,3} = \tilde{\Psi}_{1,4} = (0,0,\dots,0)$

do  $l = 1, \dots, n-1$

do  $ibox = 1, \dots, 4^l$

Tính  $\Psi_{l,ibox}$  bằng cách sử dụng bổ đề 2.4 để biến đổi khai triển đa cực  $\Phi_{l,j}$  của mỗi nút  $j$  trong danh sách tương tác của nút  $ibox$  thành khai triển địa phương xung quanh tâm của nút  $ibox$ , sau đó lấy tổng của những khai triển địa phương này và cộng kết quả với  $\tilde{\Psi}_{l,ibox}$ .

enddo

do  $ibox = 1, \dots, 4^l$

Dùng bổ đề 2.5, tính khai triển  $\Psi_{l,ibox}$  tại tâm của các nút con của  $ibox$  ta thu được khai triển  $\tilde{\Psi}_{l+1,j}$  cho các nút con của  $ibox$ .

enddo

enddo

#### Bước 4

Giải thích [*Tính các tương tác tại mức độ phân chia “mịn” nhất*]

do  $ibox = 1, \dots, 4^n$

Tính  $\Psi_{l,ibox}$  bằng cách dùng bổ đề 2.4 để biến đổi khai triển đa cực  $\Phi_{l,j}$  của mỗi nút  $j$  trong danh sách tương tác của nút  $ibox$  thành khai triển địa phương xung quanh

tâm của nút ibox, cộng các khai triển địa phương này với nhau, và cộng kết quả với  $\tilde{\Psi}_{l,ibox}$

enddo

Giải thích [*Khai triển Taylor tại mức độ phân chia mịn nhất bây giờ đã được tính. Từ khai triển này chúng ta có thể tính được thế năng hoặc lực gây ra do tất cả các hạt nằm bên ngoài các nút trong danh sách hàng xóm*]

#### Bước 5

Giải thích [*Tính khai triển địa phương tại vị trí của các hạt*]

do ibox = 1, ...,  $4^n$

Với mọi hạt  $p_j$  đặt tại điểm  $z_j$  trong hộp ibox, tính  $\Phi_{n,ibox}(z_j)$

enddo

#### Bước 6

Giải thích [*Tính thế năng (hoặc lực) gây ra bởi các hạt thuộc các nút trong danh sách hàng xóm một cách trực tiếp*]

do ibox= 1,...,  $4^n$

Với mọi hạt  $p_j$  trong nút ibox, tính tương tác với tất cả các hạt khác trong nút và với các hàng xóm gần nhất của nó.

enddo

#### Bước 7

do ibox = 1, ...,  $4^n$

Với mọi hạt trong box ibox, lấy tổng các thế năng tính trực tiếp và các thế năng tính xấp xỉ.

enddo

### c. Độ phức tạp của thuật toán FMM

Chúng ta đánh giá độ phức tạp của thuật toán FMM trên hai khía cạnh là độ phức tạp về thời gian và độ phức tạp về không gian.

Bảng dưới đây phân tích ngắn gọn độ phức tạp về thời gian của thuật toán FMM

**Bảng 2: Phân tích độ phức tạp của thuật toán FMM**

Bước	Số phép tính	Giải thích
1	Bậc $Np$	Tại các nút lá, tính khai triển đa cực cấp $p$ tại $N$ hạt.
2	Bậc $Np^2$	Tại mức $l$ , ta phải thực hiện $4^l$ phép tịnh dịch chuyển với độ phức tạp là $p^2$ .
3	Bậc $\leq 28Np^2$	Có nhiều nhất là 27 nút trong danh sách tương tác của mỗi nút ở mỗi mức. Trong vòng lặp thứ hai, độ phức tạp tính toán là $Np^2$ .
4	Bậc $\leq 27Np^2$	Có nhiều nhất là 27 nút trong danh sách tương tác đối với mỗi nút và có xấp xỉ $N$ nút.
5	Bậc $\leq 27Np^2$	Ta cần tính khai triển đa cực cấp $p$ cho mỗi hạt.
6	Bậc $\frac{9}{2}Nk_n$	Cho $k_n$ là cận trên của số lượng hạt trên mỗi nút ở mức lá. Ta cần tính tương tác giữa các hạt bên trong hộp và bên trong danh sách tương tác, nhưng sử dụng định luật 3 Newton, chỉ cần tính một nửa số tương tác từng đôi một
7	Bậc $N$	Trong bước 7 chỉ cần cộng thế năng “xa” và thế năng trực tiếp. Như vậy chỉ cần yêu cầu một phép lấy tổng hai số hạng

Như vậy, ước lượng thời gian tính toán của thuật toán FMM sẽ là:

$$N(-2a \log_2(\varepsilon) + 56b(\log_2(\varepsilon))^2 + 4.5k_n + e)$$

trong đó  $\varepsilon$  là cận trên của sai số, như ta đã biết muốn có sai số nhỏ hơn  $\varepsilon$  bậc khai triển phải là:  $p = -\log_2(\varepsilon)$ ; các hằng số  $a, b, c, d$  và  $e$  được xác định bởi một hệ thống máy tính, ngôn ngữ, cài đặt,...

Về độ phức tạp không gian của thuật toán FMM, trong thuật toán các đại lượng  $\Phi_{l,j}$  và  $\Psi_{l,j}$  phải được lưu trữ cũng như là vị trí của các hạt, điện tích (hay khối lượng khi xét lực hấp dẫn), và kết quả tính toán (thế năng, hoặc điện trường). Vì mọi nút ở mọi mức có một cặp khai triển cấp  $p$ ,  $\Phi$  và  $\Psi$  và độ dài của tất cả các mảng lưu trữ là tỉ lệ với số hạt  $N$ , nên dễ thấy độ phức tạp về không gian của thuật toán sẽ là:

$$(\alpha + \beta.p).N = (\alpha - \beta \log_2(\varepsilon)).N$$

với các hệ số  $\alpha, \beta$  tùy thuộc và hệ thống máy tính, ngôn ngữ, cài đặt.

## 2.2 Các biến thể của thuật toán FMM

### 2.2.1 Phương pháp của Anderson

Phương pháp FMM gốc do Greengard và Rokhlin đề xuất mặc dù có độ phức tạp là  $O(N)$  nhưng không thực sự đưa ra những ưu điểm vượt trội hơn so với thuật



toán cây (Barnes và Hut [4]) khi cài đặt thực tế. Một trong số những lý do là cơ sở toán học được sử dụng trong FMM là phức tạp hơn rất nhiều và cài đặt thuật toán FMM cổ điển trên máy tính cũng khó khăn hơn so với cài đặt thuật toán cây. Chính vì vậy chỉ có một vài cài đặt của thuật toán FMM trên máy tính được đưa ra. Nhưng những cài đặt này cũng không được sử dụng một cách rộng rãi và không thật sự có được sự tối ưu cần thiết. Thuật toán FMM cổ điển là một thuật toán rất phức tạp, và chỉ một vài sai sót nhỏ trong cài đặt cũng có thể dẫn tới sự kém hiệu quả của thuật toán. Trong khi đó thuật toán cây là đơn giản hơn rất nhiều và vì thế rất dễ dàng đạt được hiệu quả cao.

Anderson [2] đã đề xuất một cải tiến của thuật toán FMM sử dụng một công thức mới của khai triển đa cực và khai triển địa phương. Phương pháp của Anderson dựa trên công thức Poisson. Công thức này là lời giải của “bài toán giá trị biên” (boundary value problem) của phương trình Laplace. Trong không gian ba chiều, nếu thế năng trên một mặt cầu bán kính  $a$  là được cho trước, thế năng  $\Phi$  tại điểm  $\vec{r} = (r, \phi, \theta)$  được tính bằng công thức:

$$\Phi(\vec{r}) = \frac{1}{4\pi} \int_S \sum_{n=0}^{\infty} (2n+1) \left(\frac{a}{r}\right)^{n+1} P_n\left(\frac{\vec{r} \cdot \vec{s}}{r}\right) \Phi(a\vec{s}) d\vec{s} \quad (2.17)$$

với  $r \geq a$  và:

$$\Phi(\vec{r}) = \frac{1}{4\pi} \int_S \sum_{n=0}^{\infty} (2n+1) \left(\frac{r}{a}\right)^n P_n\left(\frac{\vec{r} \cdot \vec{s}}{r}\right) \Phi(a\vec{s}) d\vec{s} \quad (2.18)$$

với  $r \leq a$ . Trong các công thức (2.17) và (2.18) chúng ta sử dụng hệ tọa độ cầu. Ở đây  $\Phi(a\vec{s})$  là thế năng cho trước trên mặt cầu. Vùng lấy tích phân  $S$  là mặt cầu đơn vị có tâm chính là gốc tọa độ. Hàm  $P_n$  là ký hiệu của đa thức Legendre bậc  $n$ :

$$P_n(x) = (2^n n!)^{-1} \frac{d^n}{dx^n} [(x^2 - 1)^n]$$

Để sử dụng các công thức (2.17) và (2.18) thay cho các khai triển đa cực và khai triển địa phương, Anderson đã đề xuất một phiên bản rời rạc của chúng, Anderson rút gọn vế phải của phương trình (2.17)-(2.18) tại giá trị  $n$  hữu hạn, và thay thế tích phân trên mặt  $S$  với các giá trị số phân bố theo  $t$ -design cầu, với  $t$ -design cầu được Hardin và Sloane [10] định nghĩa như ở dưới đây.

Một tập hợp  $K$  điểm  $\mathcal{P} = (P_1, \dots, P_K)$  trên hình cầu đơn vị  $\Omega_d = S^{d-1} = \{x = (x_1, \dots, x_d) \in R^d : x \cdot x = 1\}$  hình thành một  $t$ -design hình cầu nếu phương trình:

$$\int_{\Omega_d} f(x) d\mu(x) = \frac{1}{K} \sum_{i=1}^K f(P_i) \quad (2.19)$$

(ở đây  $\mu$  là đơn vị chuẩn trên  $\Omega_d$  được chuẩn hóa) đúng cho tất cả các đa thức  $f$  với bậc  $\leq t$ .

Chú ý rằng tập tốt nhất, ví dụ tập nhỏ nhất của  $t$ -design là chưa được tìm ra với  $t$  tổng quát. Trong thực hành chúng ta sử dụng  $t$ -design do Hardin và Sloane tìm thấy bằng thực nghiệm. Các ví dụ về các tọa độ vị trí của các tập điểm như vậy có thể được tìm thấy tại <http://www.research.att.com/~njas/sphdesigns/>.

Sử dụng  $t$ -design cầu. Anderson đã thu được công thức đã được rời rạc hóa của (2.17) và (2.18) như sau:

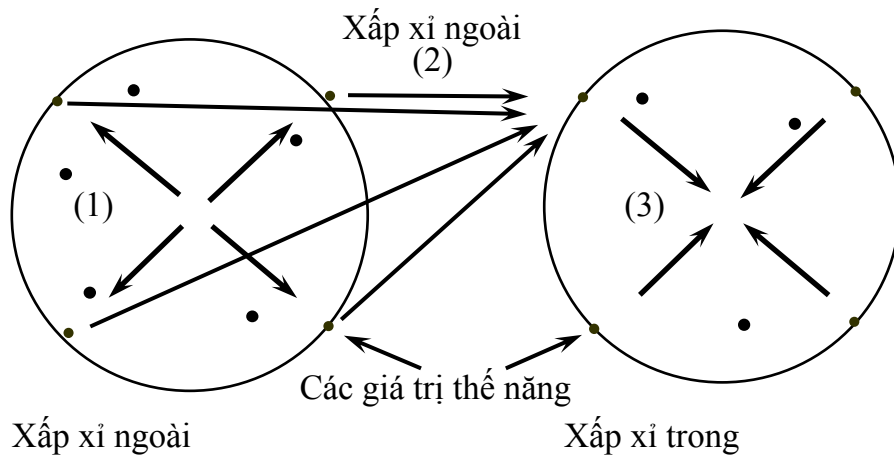
$$\Phi(\vec{r}) \approx \sum_{i=1}^K \sum_{n=0}^p (2n+1) \left(\frac{a}{r}\right)^{n+1} P_n\left(\frac{\vec{s}_i \cdot \vec{r}}{r}\right) \Phi(a \cdot \vec{s}_i) w_i \quad (2.20)$$

đối với  $r \geq a$  (khai triển ngoài) và

$$\Phi(\vec{r}) \approx \sum_{i=1}^K \sum_{n=0}^p (2n+1) \left(\frac{r}{a}\right)^n P_n\left(\frac{\vec{s}_i \cdot \vec{r}}{r}\right) \Phi(a \cdot \vec{s}_i) w_i \quad (2.21)$$

với  $r \leq a$  (khai triển trong). Ở đây  $w_i$  là các giá trị trọng lực hằng số và  $p$  là số các số hạng không bị làm tròn. Trong các phần sau chúng ta nói tới  $p$  như bậc của khai triển

Phương pháp Anderson sử dụng phương trình (2.20) và (2.21) tương ứng cho các biến đổi M2M và L2L. Các thủ tục của các pha khác là giống như trong FMM gốc. Hình 10 mô tả ý tưởng cơ bản phương pháp của Anderson:

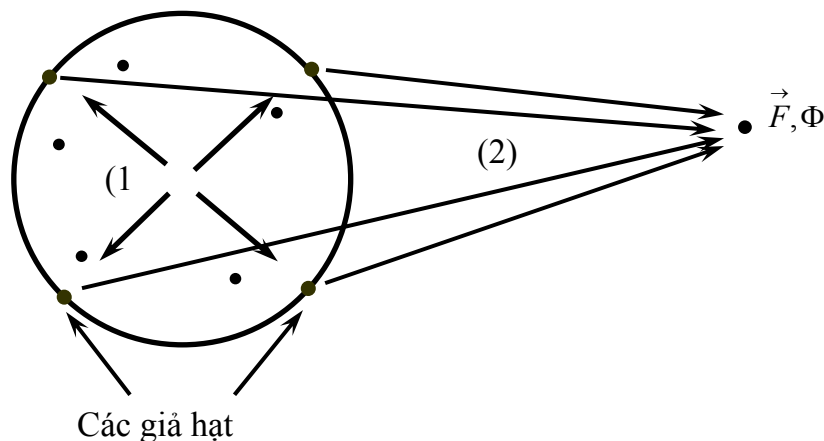


**Hình 10: Phương pháp của Anderson**

Ưu điểm chính trong phương pháp của Anderson trong thực hành là việc cài đặt khá đơn giản. Thuật toán FMM cổ điển trong hệ tọa độ 3 chiều sử dụng các công thức khá phức tạp để cài đặt các thao tác như dịch chuyển tâm của khai triển đa cực (M2M), chuyển đổi khai triển đa cực thành khai triển địa phương (M2L) và dịch chuyển tâm của khai triển địa phương (L2L). Trong phương pháp của Anderson, tất cả các thao tác dịch chuyển và biến đổi được tiến hành bằng cách tính các thế năng tại các điểm lấy mẫu trên mặt cầu. Vì thế cơ sở toán học của phương pháp được giới hạn trong hai công thức (2.20) và (2.21).

### 2.2.2 Phương pháp giả hạt của Makino

Trong phương pháp của Anderson, khai triển đa cực của thế năng gây ra bởi một nhóm hạt được biểu diễn bằng các giá trị thế năng trên một mặt cầu bao quanh các hạt đó. Thế năng tại một điểm bất kỳ bên ngoài mặt cầu sẽ được tính bằng tích phân trên mặt cầu đó. Trong thực hành, tích phân nói trên được tính xấp xỉ bằng tổng trên các hạt mẫu nằm trên mặt cầu. Makino [16] đã đề xuất một cải tiến khác khá giống với phương pháp của Anderson. Ý tưởng cơ bản của phương pháp này là sử dụng một số lượng nhỏ các giả hạt trong biểu diễn khai triển đa cực. Thay vì sử dụng phân phối thế năng trên mặt cầu, Makino đã sử dụng phân phối khối lượng. Phương pháp này của Makino khi kết hợp với máy chuyên dụng GRAPE (GRAPE, Makino và Taiji 1998 [15]; Sugimoto 1999 [20]) đã tăng tốc độ tính lực từ 100-1000 lần so với các máy thông thường.



Hình 11: Phương pháp giả hạt của Makino

### a. Trong hệ tọa độ 2 chiều

Trong hệ tọa độ 2 chiều, khai triển đa cực của thế năng hấp dẫn do một hạt gây ra được cho bởi:

$$\Phi_{z_0}(z) = m \log(z - z_0) = m \log(z) - m \sum_{k=1}^{\infty} \frac{(z_0/z)^k}{k} \quad (2.22)$$

ở đây  $z_0$  và  $z$  là vị trí của hạt và vị trí của điểm tính thế năng trong mặt phẳng phức, và  $m$  là khối lượng của hạt. Công thức trên sẽ hội tụ nếu  $|z| > |z_0|$ .

Nếu chúng ta có  $N$  hạt với khối lượng  $m_i$  tại các vị trí  $z_i$  ( $|z_i| < a$ ), công thức tính thế năng bên ngoài đường tròn bán kính  $a$  là:

$$\Phi(z) = M \log(z) - \sum_{k=1}^{\infty} \frac{\alpha_k}{k} (a/z)^k \quad (2.23)$$

với  $M$  là tổng khối lượng của các hạt và hệ số  $\alpha_k$ :

$$\alpha_k = \sum_{i=1}^N m_i (z_i/a)^k \quad (2.24)$$

Ta cần xác định phân phối khối lượng của các giả hạt trên đường tròn bán kính  $a$ . Các hệ số của khai triển đa cực sẽ được xác định bằng công thức:

$$\alpha_k = (r/a)^k \int_0^{2\pi} e^{ik\theta} \rho(\theta) d\theta \quad (2.25)$$

ở đây  $\rho$  là mật độ của khối lượng tại tọa độ cực  $(r, \theta)$ . Vì thế từ các hệ số khai triển  $\alpha_k$ ,  $\rho(\theta)$  có thể tính từ chuỗi Fourier:

$$\rho(\theta) = \frac{1}{2\pi} \left( \frac{a}{r} \right)^k \sum_{k=0}^{\infty} \alpha_k e^{-ik\theta} \quad (2.26)$$

Makino đã xấp xỉ khối lượng  $m$  liên tục này bằng  $2p+1$  điểm rời rạc tại  $\theta_j = 0, 2\pi/(2p+1), 4\pi/(2p+1) \dots$ , khối lượng  $m_j$  được cho bởi công thức:

$$m_j = \frac{1}{2p+1} \left( \frac{a}{r} \right)^k \sum_{k=0}^p \alpha_k e^{-ik\theta_j} \quad (2.27)$$

Do tính chất của chuỗi Fourier, các phân phối khối lượng  $m_j$  này sẽ biểu diễn chính xác các khai triển đa cực tới cấp  $p$ . Để tính thế năng bên ngoài đường tròn, ta lấy tổng các thế năng do những hạt này gây ra:

$$\Phi(z) = \sum_{j=1}^{2p+1} m_j \log(z - z_j) \quad (2.28)$$

trong đó  $z_j = re^{2ij\pi/(2p+1)}$ .

Đối với pha M2M ở cả hai thuật toán cây và FMM, chúng ta vẫn phải tính khai triển đa cực hoặc biểu diễn dưới dạng hạt tại tâm của một nút từ những nút con của nó. Vì các nút con đã được biểu diễn dưới dạng hạt nên chúng ta có thể sử dụng phương trình (2.24) để thu được hệ số khai triển của các nút cha.

Trong cải tiến của mình, thay vì việc sử dụng các hệ số khai triển đa cực, Makino đã tính khối lượng của các giả hạt trực tiếp từ khối lượng của các hạt vật lý (hoặc từ khối lượng của các giả hạt trong các nút con). Kết hợp hai phương trình (2.24) và (2.27) ta có công thức tính trực tiếp  $m_j$  từ  $m_i$  như sau:

$$m_j = \frac{1}{2p+1} \sum_{k=0}^p \sum_{i=1}^n m_i (z_i / z_j)^k = \frac{1}{2p+1} \sum_{i=1}^n m_i \frac{1 - (z_i / z_j)^{p+1}}{1 - z_i / z_j} \quad (2.29)$$

Cuối cùng cần phải xác định thuật toán cho pha M2L và L2L. Chúng ta có thể sử dụng phương pháp của Anderson hoặc khai triển đa cực. Đối với khai triển địa phương, phương pháp của Anderson dễ cài đặt hơn so với khai triển đa cực.

## b. Trong hệ tọa độ 3 chiều

Các công thức trong hệ tọa độ 3 chiều về cơ bản là giống với trường hợp 2 chiều ngoại trừ việc chúng ta dùng điều hòa cầu thay cho  $z^k$ . Biểu thức của hệ số khai triển  $\alpha_l^m$  là:

$$\alpha_l^m = \sum_{i=1}^N m_i r_i^l Y_l^{-m}(\theta_i, \phi_i) \quad (2.30)$$

trong đó  $m_i$  là khối lượng của hạt thứ  $i$  và  $(r_i, \theta_i, \phi_i)$  là tọa độ cực của nó. Hàm số  $Y_l^m(\theta, \phi)$  là điều hòa cầu bậc  $l$ , có công thức là:

$$Y_l^m = (-1)^m \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) e^{im\phi} \quad (2.31)$$

và  $P_l^m$  là hàm số Legendre kết hợp giữa cấp  $l$  và bậc  $m$ . Với các hệ số khai triển  $\alpha_l^m$ , ta tính được thế năng tại điểm  $(r, \theta, \phi)$  là:

$$\Phi(r, \theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{\alpha_l^m}{r^{l+1}} Y_l^m(\theta, \phi) \quad (2.32)$$

Mục đích của chúng ta là tính phân phối khối lượng  $\rho(\theta, \phi)$  trên một hình cầu bán kính  $a$  mà thỏa mãn:

$$\alpha_l^m = \int_S \rho(\theta, \phi) Y_l^{-m}(\theta, \phi) dS \quad (2.33)$$

trong đó là kí hiệu của bề mặt hình cầu. Vì điều hòa cầu chứa một hệ trục giao nên  $\rho$  được biểu diễn bằng công thức sau:

$$\rho = \sum_{l=0}^{\infty} \sum_{m=-l}^l \alpha_l^m Y_l^{*-m} \quad (2.34)$$

Nếu chúng ta sử dụng  $K$  điểm trên hình cầu, khối lượng của chúng là:

$$m_j = \frac{1}{4\pi K} \sum_{l=0}^{\infty} p \sum_{m=-l}^l \alpha_l^m Y_l^{*-m} \quad (2.35)$$

Khai triển chuỗi này phải được cắt cụt ở một giá trị hữu hạn như chúng ta đã thấy trong trường hợp 2 chiều. Như đã nói trong phương pháp của Anderson, bậc “cắt cụt” phải là  $[t/2]$  nếu  $K$  điểm tạo thành một  $t$ -design.

Như đã trình bày trong trường hợp 2 chiều, chúng ta có thể biến đổi trực tiếp các vị trí và khối lượng của các hạt vật lý thành vị trí và khối lượng của các giả hạt. Biểu thức chuẩn sẽ là một tổng xichma gồm 3 phần trên  $i$ ,  $l$ , và  $m$ . Để đơn giản giản hóa công thức này chúng ta sử dụng một định lý về điều hòa cầu. Định lý đó là:

$$P_l(\cos \gamma) = \frac{4\pi}{2l+1} \sum_{m=-l}^l Y_l^m(\theta, \phi) Y_l^{-m}(\theta', \phi') \quad (2.36)$$

trong đó  $\gamma$  là góc giữa hai vector có hướng là  $(\theta, \phi)$  và  $(\theta', \phi')$ . Sử dụng định lý ta tính được các khối lượng  $m_j$  như sau:

$$m_j = \sum_{i=1}^N m_i \sum_{l=0}^p \frac{2l+1}{K} \left( \frac{r_i}{a} \right)^l P_l(\cos \gamma_{ij}) \quad (2.37)$$

trong đó  $\gamma_{ij}$  là góc giữa hai vector tương ứng với hai hạt vật lý  $i$  và giả hạt  $j$ .

Như vậy thế năng gây ra bởi các hạt vật lý sẽ được tính xấp xỉ bằng thế năng gây ra bởi các giả hạt này.

Phương trình (2.37) đưa ra một giải pháp cho trường hợp khai triển ngoài, ứng dụng trong pha M2M của thuật toán FMM. Đối với pha M2L là một pha chiếm phần lớn thời gian tính toán của thuật toán, phương pháp của Makino chưa đề cập đến, và trong thực hành, công thức khai triển trong của Anderson được sử dụng.

Với cách tiếp cận tương tự Chau, Kawai và Ebisuzaki [13] đã đạt được giải pháp đối với khai triển trong:

$$m_j = \sum_{i=1}^N m_i \sum_{l=0}^p \frac{2l+1}{K} \left( \frac{a}{r_i} \right)^{l+1} P_l(\cos \gamma_{ij}) \quad (2.38)$$

Tóm lại, phương pháp giả hạt  $P^2M^2$  (Pseudo-Particle Multipole Method) của Makino đưa ra có hai ưu điểm so với phương pháp FMM chuẩn sử dụng khai triển đa cực một cách trực tiếp. Ưu điểm thứ nhất là sự đơn giản. Như chúng ta thấy, các công thức biến đổi sử dụng trong  $P^2M^2$  là đơn giản hơn rất nhiều so với các công thức của FMM. Sự đơn giản ở đây có nghĩa là sự thay đổi trong cài đặt và sự song song hóa là dễ dàng, mặc dù chi phí tính toán của hai phương pháp là gần tương đương đối với cùng một độ chính xác.

Một ưu điểm nữa là  $P^2M^2$  tận dụng được ưu thế của máy tính chuyên dụng GRAPE. GRAPE là một máy tính có bộ xử lý theo kiểu pipeline để tính lực giữa các hạt. Mặc dù thuật toán cây đã được cài đặt trên máy tính GRAPE, nhưng nó khó đạt được độ chính xác cao do máy tính GRAPE chỉ có thể tính được khai triển đơn cực. Với  $P^2M^2$  chúng ta có thể tính toán các cấp cao hơn sử dụng máy tính GRAPE, vì các số hạng được biểu diễn bằng sự phân bố của các giả hạt

## 2.3 Tổng kết chương

Trong chương 2 của khóa luận chúng ta đã tìm hiểu về cơ sở toán học, cài đặt của thuật toán FMM cổ điển và các cải tiến của Anderson và Makino. Thuật toán FMM giống như tên của nó thể hiện là một thuật toán nhanh được sử dụng để tăng tốc độ tính lực trong bài toán mô phỏng động lực phân tử. Nhưng trong thuật toán FMM gốc, do các công thức toán học phức tạp nên việc cài đặt FMM trên máy tính còn hạn chế và kém phổ biến hơn so với thuật toán cây. Anderson và Makino với các cải tiến của mình đã làm cho việc cài đặt FMM trở nên dễ dàng hơn và đạt hiệu năng tốt hơn so với thuật toán gốc. Trong chương 3 của khóa luận, chúng ta sẽ tìm hiểu về cách ứng dụng phương pháp SVD trong bài toán mô phỏng động lực phân tử.

## Chương 3. ÁP DỤNG PHƯƠNG PHÁP SVD TRONG MÔ PHỎNG ĐỘNG LỰC PHÂN TỬ

### 3.1 Phương pháp SVD

Phân tích SVD [17, 18], là một phương pháp đại số rất mạnh và hữu dụng với nhiều ứng dụng trong xử lý tín hiệu số và khoa học thống kê. Phương pháp SVD được sử dụng nhiều trong nhiều bài toán có liên quan đến việc tính toán ma trận mà nếu áp dụng các phương pháp thông thường như phương pháp khử Gauss hay phương pháp phân tích  $LU$  sẽ cho kết quả với sai số lớn.

Phương pháp SVD dựa trên định lý sau đây trong đại số tuyến tính: bất kỳ ma trận  $A$  kích thước  $M \times N$  nào mà số  $M \geq N$ , có thể được viết dưới dạng tích của một ma trận  $U$  trực giao theo cột có kích thước  $M \times N$ , một ma trận chéo  $W$  có kích thước  $N \times N$  với các số trên đường chéo là không âm, và ma trận chuyển vị của một ma trận trực giao  $V$  có kích thước  $N \times N$ :

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \begin{pmatrix} w_1 & & & \\ & w_2 & & \\ & & \dots & \\ & & & w_N \end{pmatrix} \begin{pmatrix} V^T \end{pmatrix} \quad (3.1)$$

Ma trận  $U$  và ma trận  $V$  trực giao theo nghĩa: các cột của chúng là trực giao:

$$\sum_{i=1}^M U_{ik} U_{in} = \delta_{kn} \quad \begin{cases} 1 \leq k \leq N \\ 1 \leq n \leq N \end{cases} \quad (3.2)$$

$$\sum_{j=1}^N V_{jk} V_{jn} = \delta_{kn} \quad \begin{cases} 1 \leq k \leq N \\ 1 \leq n \leq N \end{cases} \quad (3.3)$$

trong đó  $\delta_{kn} = 1$  nếu  $k = n$  và bằng 0 nếu  $k \neq n$ . Hoặc biểu diễn dưới dạng ma trận:

$$\begin{pmatrix} U^T \end{pmatrix} \begin{pmatrix} U \end{pmatrix} = \begin{pmatrix} V^T \end{pmatrix} \begin{pmatrix} V \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix} \quad (3.4)$$

Vì  $V$  là ma trận vuông nên  $V$  đồng thời là ma trận trực giao theo hàng:  $V.V^T = 1$ .



Nếu ký hiệu  $u_i$  và  $v_i$  tương ứng là các cột của  $U$  và  $V$ , thì phương trình (3.1) được viết dưới dạng các tích ngoài như sau:

$$A = \sum_{i=1}^n w_i u_i v_i^T \quad (3.5)$$

Các giá trị  $w_i$  được gọi là các giá trị kỳ dị của  $A$ . Không mất tính tổng quát, chúng ta quy ước rằng các giá trị kỳ dị này được sắp xếp theo thứ tự giảm dần:  $w_1 \geq w_2 \geq \dots \geq w_n \geq 0$ . Chỉ số của giá trị kỳ dị khác 0 cuối cùng trong dãy được ký hiệu là  $\rho_0$ , tức là  $w_i = 0$  với  $i > \rho_0$  và  $w_i > 0$  với  $i \leq \rho_0$ .

### 3.1.1 SVD của ma trận vuông

Nếu ma trận  $A$  là vuông, có kích thước  $N \times N$  thì  $U$ ,  $V$  và  $W$  tất cả đều là ma trận vuông cùng cỡ. Chúng ta dễ dàng tìm được ma trận nghịch đảo của chúng:  $U$  và  $V$  là các ma trận trực giao nên các ma trận nghịch đảo của chúng chính là các ma trận chuyển vị tương ứng;  $W$  là ma trận chéo vì thế ma trận nghịch đảo của  $W$  là ma trận chéo với các phần tử là nghịch đảo của các phần tử  $w_j$ . Từ phương trình (3.1) suy ra ma trận nghịch đảo của  $A$  là:

$$A^{-1} = V \cdot [\text{diag}(1/w_j)] \cdot U^T \quad (3.6)$$

Công thức (3.6) sẽ không còn đúng nếu các tồn tại một giá trị  $w_j$  bằng 0. Khi cài đặt trên máy tính, nếu giá trị  $w_j$  gần tới 0, thì nghịch đảo của giá trị này sẽ vượt qua giá trị cho phép của máy tính.

Ta định nghĩa số điều kiện của ma ma trận  $A$  là tỉ lệ giữa số  $w_j$  lớn nhất với số  $w_j$  bé nhất. Ma trận  $A$  được gọi là suy biến nếu số điều kiện này là vô hạn và được gọi là có điều kiện xấu (*ill-conditioned*) nếu số điều kiện là quá lớn theo nghĩa là số nghịch đảo của nó vượt quá giá trị sai số của máy tính (khoảng  $10^{-6}$  với độ chính xác đơn và  $10^{-12}$  với độ chính xác kép).

Với các ma trận suy biến, có hai khái niệm quan trọng là *range* và *nullspace*. Xét một hệ phương trình tuyến tính biểu diễn dưới dạng ma trận sau đây:

$$A \cdot x = b \quad (3.7)$$

trong đó  $A$  là ma trận vuông,  $b$  và  $x$  là các vector. Phương trình (3.7) định nghĩa  $A$  giống như một ánh xạ tuyến tính từ không gian vector  $x$  tới không gian vector  $b$ . Nếu ma trận  $A$  là suy biến thì tồn tại một không gian con của  $x$ , gọi là *nullspace* được ánh

xạ tới 0,  $Ax = 0$ . Số chiều của *nullspace* (Số các vector  $x$  được lập tập tuyến tính trong đó) được gọi là *nullity* của  $A$ .

*Range* của  $A$  là một không gian con của không gian vector  $b$  sao cho với mỗi vector  $b$  thuộc không gian vector này, tồn tại vector  $x$  sao cho  $Ax = b$ . Số chiều của không gian con *Range* này được gọi là hạng (rank) của  $A$ .

Đối với ma trận vuông  $A \in R^{n,n}$  ta có:  $rank(A) + nullity(A) = n$ . Nếu ma trận  $A$  không suy biến thì  $rank(A) = n$ .

Bằng phương pháp SVD, chúng ta xác định được cơ sở trực giao cho *nullspace* và *range* của  $A$ . Tập các cột  $j$  của ma trận  $U$  tương ứng với các số  $w_j$  khác không tạo thành cơ sở trực giao của *range*, với quy ước nói trong phần đầu,  $\{u_1, \dots, u_{\rho_0}\}$  sẽ tạo thành cơ sở trực giao của  $range(A)$ ; tập các cột  $j$  của  $V$  có các số  $w_j$  bằng 0 tức là tập  $\{v_{\rho_0+1}, \dots, v_n\}$  tạo thành cơ sở trực giao của *nullspace*.

### 3.1.2 Giải hệ phương trình tuyến tính

#### a. Cách giải hệ phương trình tuyến tính bằng SVD

Như đã trình bày, phương pháp SVD có thể được sử dụng để giải hệ phương trình tuyến tính. Xét hệ phương trình dưới dạng ma trận ở (3.7):

$$Ax = b$$

trong đó  $A$  là ma trận vuông,  $x$ ,  $b$  là các vector. Nếu  $A$  là không suy biến (các giá trị  $w_i$  khác 0 trong phân tích SVD), phương trình (3.7) sẽ có nghiệm duy nhất là:

$$x = A^{-1}b$$

$A^{-1}$  là ma trận nghịch đảo của ma trận  $A$  được tính bằng công thức (3.6).

Nhưng nếu ma trận  $A$  là ma trận suy biến (tồn tại một hay nhiều các giá trị kỳ dị  $w_i$  bằng 0 hoặc rất nhỏ), hệ phương trình sẽ không chỉ có nghiệm duy nhất. Chúng ta xét hai trường hợp:

Nếu hệ phương trình là thuần nhất tức là  $b = 0$ , hệ có thể được giải trực tiếp nhờ SVD: các cột của ma trận  $V$  tương ứng với giá trị kỳ dị  $w_j = 0$  sẽ là nghiệm của hệ.

Nếu hệ phương trình là không thuần nhất, tức là vector  $b$  ở vế phải của (3.7) khác vector 0, thì hệ sẽ vô nghiệm (nếu  $b \notin range(A)$ ) hoặc có vô số nghiệm (nếu

$b \in \text{range}(A)$ , khi đó tổ hợp tuyến tính của bất kỳ các vector nằm trong không gian nullspace của  $A$  và một nghiệm đặc biệt  $x_0$  của hệ sẽ sinh ra một nghiệm khác).

Trong trường hợp muốn tìm một vector nghiệm đặc biệt  $x$  sao cho: độ dài  $|x|^2$  bé nhất, có thể sử dụng phân tích SVD theo cách, thay thế các giá trị  $1/w_j$  bởi 0 nếu  $w_j = 0$ , sau đó tính:

$$x = V.[\text{diag}(1/w_j)].(U^T b) \quad (3.8)$$

**Chứng minh:** Xem xét  $|x + x'|$ , ở đây  $x'$  nằm trong nullspace của  $A$ . Gọi  $W^{-1}$  là ma trận nghịch đảo của ma trận  $W$  sau khi đã thay  $1/w_j = 0$  nếu  $w_j = 0$ . Ta có:

$$|x + x'| \stackrel{(1)}{=} |V.W^{-1}.U^T.b + x'| \stackrel{(2)}{=} |V.(W^{-1}.U^T.b + V^T.x')| \stackrel{(3)}{=} |W^{-1}.U^T.b + V^T.x'| \quad (3.9)$$

đẳng thức (1) của (3.9) được suy ra từ (3.8) và các đẳng thức (2), (3) được suy ra do tính trực giao ma trận  $V$ .

Xét hai số hạng của vế phải trong (3.9), ta thấy số hạng đầu có phần tử thứ  $j$  khác không khi và chỉ khi  $w_j \neq 0$ , trong khi số hạng thứ hai, do  $x'$  nằm trong nullspace của  $A$  nên có thành phần thứ  $j$  khác không khi và chỉ khi  $w_j = 0$ . Do đó độ dài của vector  $(x + x')$  bé nhất khi  $x'$  là vector không. (Điều phải chứng minh)

Nếu vector  $b$  không nằm trong không gian range của ma trận suy biến  $A$ , hệ phương trình (3.7) sẽ vô nghiệm. Nhưng ta có thể sử dụng công thức (3.8) để tính nghiệm xấp xỉ  $x$ . Vector này sẽ không thỏa mãn phương trình ma trận (3.7). Nhưng nó sẽ cho nghiệm gần đúng theo nghĩa bình phương tối thiểu tức là với  $x$  xác định bằng công thức (3.8), ta có:

$$r = |Ax - b| \quad (3.10)$$

sẽ có giá trị cực tiểu. Số  $r$  được gọi là phần dư hay sai số của nghiệm.

Việc chứng minh định lý trên là tương tự (3.9): Giả sử ta thêm cộng thêm vào vector  $x$  một vector  $x'$  bất kỳ. Khi đó  $Ax - b = Ax - b + b'$  với  $b' = Ax'$ . Rõ ràng là  $b'$  nằm trong range của  $A$ . Ta có:

$$\begin{aligned} |Ax - b + b'| &= |(U.W.V^T).(V.W^{-1}.U^T.b) - b + b'| \\ &= |(U.W.W^{-1}.U^T - 1).b + b'| \\ &= |U.[(W.W^{-1} - 1).U^T.b + U^T.b']| \\ &= |(W.W^{-1} - 1).U^T.b + U^T.b'| \end{aligned} \quad (3.11)$$

Trong biến đổi trên  $(W.W^{-1} - 1)$  là một ma trận chéo với các thành phần  $j$  khác 0 khi và chỉ khi  $w_j = 0$ , trong khi  $U^T.b'$  có các thành phần  $w_j$  khác 0 khi và chỉ khi  $w_j \neq 0$ , vì  $b'$  nằm trong range của ma trận  $A$ . Do đó giá trị cực tiểu sẽ đạt được với  $b'=0$  (điều phải chứng minh).

### b. Vấn đề chọn tham số “gần 0” trong phương pháp SVD

Trong phương trình (3.7), giả sử ma trận  $A$  có một tất cả các giá trị kỳ dị  $w_j$  khác không, nhưng có một vài giá trị rất nhỏ, sao cho ma trận  $A$  có điều kiện xấu, tức là tỉ lệ của giá trị kỳ dị  $w_j$  bé nhất và giá trị kỳ dị  $w_j$  lớn nhất nhỏ hơn sai số của máy tính. Khi đó phương pháp phân tích  $LU$  và phương pháp khử Gauss có thể đưa ra nghiệm cho hệ phương trình, nhưng khi nhân  $A$  với vector nghiệm của chúng, kết quả sẽ có sai số rất lớn so với vế phải  $b$  của hệ. Trong trường hợp đó, chúng ta sẽ “không hóa” các giá trị kỳ dị  $w_j < \varepsilon$  với  $\varepsilon$  là một giá trị cho trước, tức là gán các giá trị  $w_j = 0$  trong trường hợp  $w_j < \varepsilon$  và sử dụng công thức (3.8) để tính ra vector nghiệm xấp xỉ. Vector nghiệm xấp xỉ này sẽ tốt hơn (theo nghĩa phần dư  $|Ax - b|$  nhỏ hơn) so với phương pháp trực tiếp và phương pháp SVD khi ta chưa xử lý các giá trị kỳ dị gần 0.

Trong áp dụng SVD trong thực hành, chúng ta phải xác định rõ tham số “gần 0” bằng bao nhiêu để không hóa các giá trị kỳ dị, và giá trị của phần dư  $|Ax - b|$  bằng bao nhiêu là chấp nhận được. Vấn đề chọn tham số gần 0 chưa được chỉ ra bằng công thức hay lý thuyết, do đó tùy từng ứng dụng tham số gần 0 sẽ được chọn dựa trên thực nghiệm.

### 3.1.3 Cài đặt phương pháp SVD trên máy tính

Trong khóa luận, chúng tôi sử dụng lại bản cài đặt SVD bằng ngôn ngữ C trong tài liệu tham khảo [19]. Có hai thủ tục chính được sử dụng là:

**(1) void svdcmp(float\*\* a, int m, int n, float w[], float \*\*v)**

*Input:* Ma trận  $A$  được xác định bằng mảng  $a[1...m][1...n]$ ,  $m, n$  tương ứng là số dòng và cột của ma trận.

*Output:* Phân tích SVD,  $A = U.W.V^T$ , trong đó ma trận  $a[1...m][1...n]$  là ma trận  $U$  trong phân tích, ma trận chéo  $W$  được ghi vào mảng  $w[1...n]$  các phần tử trên đường

chéo, và ma trận  $v[1..n][1..n]$  là ma trận  $V$  (không phải là ma trận chuyển vị của  $V$ ) trong khai triển.

(2) `void svbksb(float** u, float w[], int m, int n, float b[], float x[])`

Thủ tục này để giải phương trình dạng  $Ax = b$ , ở đây ma trận  $A$  được xác định bằng mảng hai chiều  $a[1..m][1..n]$ , mảng  $w[1..n]$  và  $v[1..n][1..n]$ . Đây là các output trả về từ thủ tục `svdcmp()`.  $b[1..m]$  là vector vế phải của phương trình.  $x[1..n]$  là vector nghiệm.

Trước khi áp dụng thủ tục `svbksb()` để thu được nghiệm của hệ phương trình, cần gọi thủ tục `zero_small_value(float** w, int n, double threshold)` để xử lý các giá trị gần 0 trong ma trận chéo  $W$ . Việc chọn giá trị `threshold` là tùy từng ứng dụng và được chọn dựa trên thực nghiệm.

Chi tiết về các thủ tục trên có trong phần Phụ lục A của khóa luận.

## 3.2 Ứng dụng của phương pháp SVD trong inner $P^2M^2$

### 3.2.1 Cài đặt thuật toán FMM trên máy GRAPE

#### a. Chức năng của máy GRAPE

Chức năng chủ yếu của GRAPE là để tính lực  $\vec{f}(\vec{r}_i)$  trên hạt  $i$  tại vị trí  $\vec{r}_i$ , và thế năng  $\Phi(\vec{r}_i)$ . Mặc dù có nhiều các loại máy GRAPE khác nhau cho các ứng dụng khác nhau trong vật lý thiên văn (astrophysics) và động lực phân tử (Molecular dynamics) nhưng các chức năng cơ bản của chúng là giống nhau.

Lực  $\vec{f}(\vec{r}_i)$  và thế năng  $\Phi(\vec{r}_i)$  được biểu diễn bằng công thức:

$$\vec{f}(\vec{r}_i) = \sum_{j=1}^N \frac{m_j (\vec{r}_i - \vec{r}_j)}{r_s^3} \quad (3.12)$$

và

$$\Phi(\vec{r}_i) = \sum_{j=1}^N \frac{q_j}{r_s} \quad (3.13)$$

với  $N$  là số lượng các hạt trong hệ,  $\vec{r}_j$  và  $m_j$  là vị trí và khối lượng của hạt  $j$ , và  $r_s$  được gọi là khoảng cách đã được “mềm hóa” giữa hạt  $i$  và hạt  $j$  được định nghĩa:  $r_s = |\vec{r}_i - \vec{r}_j|^2 + \varepsilon^2$  ở đây  $\varepsilon$  là tham số làm mềm.

Để tính lực  $\vec{f}(\vec{r}_i)$ , cần gửi các dữ liệu cần thiết  $\vec{r}_i, \vec{r}_j, q_j, \varepsilon$  và  $N$  từ máy tính thông thường tới máy GRAPE. GRAPE sẽ tính các lực  $\vec{f}(\vec{r}_i)$  với mọi  $i$  sau đó gửi trả lại kết quả đến máy tính thông thường. Thế năng  $\Phi(\vec{r}_i)$  được tính theo cách tương tự.

## **b. Cài đặt thuật toán FMM trên máy GRAPE**

Thuật toán FMM bao gồm 5 pha (xem phần 2.1.2 a), cụ thể là: tạo cây, chuyển đổi M2M, biến đổi M2L, biến đổi L2L, và tính lực. Pha tính lực bao gồm tính các lực gây ra bởi các hạt ở “gần” và các hạt ở “xa”.

Trong thuật toán FMM gốc, chỉ có phần tính lực trực tiếp do các hạt ở gần gây ra là có thể thực hiện trên máy tính GRAPE. Trong phần này, GRAPE sẽ tính trực tiếp các lực tác dụng trên mỗi hạng có dạng như ở (3.12).

Chau, Kawai, Ebisuzaki [13] đã cải tiến thuật toán FMM gốc để máy GRAPE có thể thực hiện tính toán trong pha M2L, pha chiếm hầu hết thời gian tính toán. Với cải tiến này, GRAPE có thể thực hiện tính toán trong pha M2L thông qua tính giá trị thế năng từ các giả hạt. Trong pha L2L, giá trị thế năng được khai triển địa phương và được dịch chuyển tâm khai triển, sử dụng phương pháp của Anderson.

Sau đây là mô tả ngắn gọn của cài đặt này:

### **i. Tạo cây**

Pha tạo cây không có gì thay đổi. Phần này được thực hiện giống như trong phương pháp FMM gốc.

### **ii. Biến đổi M2M**

Tại pha M2M, tính các vị trí và khối lượng (điện tích) của các giả hạt thay cho việc tính khai triển đa cực như trong thuật toán FMM gốc.

Thủ tục bắt đầu từ các nút lá. Từ vị trí và khối lượng của các hạt vật lý, ta tính các vị trí và khối lượng của các nút lá. Sau đó các nút không phải là nút lá được tính từ các vị trí và các khối lượng của các giả hạt của các nút con. Thủ tục này tiếp tục cho đến khi đạt tới nút gốc. Thủ tục này hoàn toàn được thực hiện trên máy tính thường.

### **iii. Biến đổi M2L**

Pha M2L được thực hiện trên máy tính GRAPE. Sự khác nhau giữa phương pháp FMM gốc và phương pháp này đó là chúng ta không sử dụng công thức để

chuyển khai triển đa cực thành khai triển cục bộ. Chúng ta tính trực tiếp các giá trị thế năng gây ra do các giả hạt trong danh sách tương tác của mỗi nút.

#### iv. Biến đổi L2L

Pha L2L được thực hiện giống như phương pháp của Anderson đã thực hiện. Chúng ta sử dụng công thức khai triển trong (2.21) (phần 2.2.1) để chuyển khai triển địa phương của mỗi nút thành khai triển địa phương của nút con.

#### v. Tính lực (“gần”)

Lực do các hạt ở gần gây ra được tính trực tiếp bằng tính các lực tương tác hạt-hạt. Phần này được thực hiện trên máy chuyên dụng GRAPE.

#### vi. Tính lực (“xa”)

Sử dụng công thức khai triển trong (2.21) (phần 2.2.1), thế năng “xa” trên một hạt ở vị trí  $\vec{r}$  có thể được tính từ tập các giá trị thế năng của nút là chứa các hạt đó. Có nghĩa là lực “xa” được tính bằng đạo hàm của công thức (2.22):

$$-\nabla\Phi(\vec{r}) = \sum_{i=1}^K \sum_{n=0}^p \left( n\vec{r}P_n(u) + \frac{u\vec{r} - \vec{s}_i r}{\sqrt{1-u^2}} \nabla P_n(u) \right) (2n+1) \frac{r^{n-2}}{a^n} g(a\vec{s}_i) w_i \quad (3.14)$$

trong đó  $u = \vec{s}_i \cdot \vec{r} / r$ .

Tất cả các tính toán trong pha này được thực hiện trên máy tính thường.

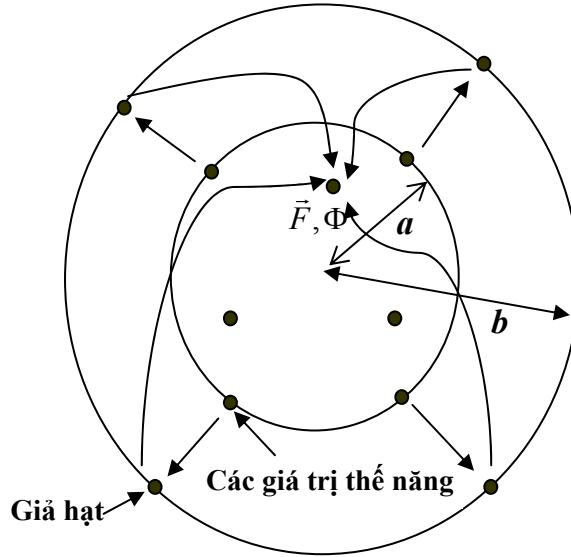
### 3.2.2 Ứng dụng của SVD trong cài đặt inner $P^2M^2$

Cải tiến ở trên đã giải quyết tình trạng thắt cổ chai trong phương pháp FMM, vì pha biến đổi M2L được thực hiện trên máy tính GRAPE. Bây giờ, phần chiếm nhiều thời gian tính toán nhất là tính toán lực “xa”. Công thức (3.14) là phức tạp, vì thế sẽ chiếm thời gian lớn trong toàn bộ thời gian tính toán.

Trong [13], Chau, Kawai, Ebisuzaki đã đưa ra một cách cài đặt khác để tính toán các lực “xa”. Ý tưởng chính của cải tiến này là sử dụng biến đổi A2P để tính phân bố của các giả hạt sinh ra trường thế năng cho bởi công thức khai triển trong của Anderson (công thức 2.21 phần 2.2.1). Khi các phân bố của các giả hạt được tính, pha L2L có thể được thực hiện dùng công thức khai triển trong  $P^2M^2$  (công thức 2.38 phần 2.2.2 b).

$$m_j = \sum_{i=1}^N m_i \sum_{l=0}^p \frac{2l+1}{K} \left( \frac{a}{r_i} \right)^{l+1} P_l(\cos \gamma_{ij})$$

Sau đó pha tính lực sẽ được thực hiện hoàn toàn trên máy GRAPE.



**Hình 12: Tính thế năng và lực từ phân phối khối lượng của các giả hạt**

Trong biến đổi A2P, bước đầu tiên, chúng ta phân phối các giả hạt trên một mặt cầu với bán kính  $b$  sử dụng  $t$ -design cầu. Ở đây,  $b$  lớn hơn bán kính  $a$  của hình cầu trong giá trị thế năng  $g(a\vec{s})$  được định nghĩa trong công thức khai triển của Anderson. Theo công thức (2.38), có thể thay đổi khối lượng của các hạt. Vì thế quan hệ:

$$\sum_{j=1}^K \frac{m_j}{|\vec{R}_j - a\vec{s}_i|} = \Phi(a\vec{s}_i) \quad (3.15)$$

thỏa mãn với mọi  $i = 1, \dots, K$ . Sử dụng một ma trận  $R = \{1/|\vec{R}_j - a\vec{s}_i|\}$  và vector  $\vec{M} = [m_1, m_2, \dots, m_K]^T$  và  $\vec{P} = [\Phi(a\vec{s}_1), \Phi(a\vec{s}_2), \dots, \Phi(a\vec{s}_K)]$ , chúng ta có thể viết lại công thức (3.15) dưới dạng:

$$R\vec{M} = \vec{P} \quad (3.16)$$

Trong bước tiếp theo, chúng ta giải hệ phương trình (3.16) để thu được các khối lượng  $m_j$ . Do sự sai khác lớn của các bán kính  $b$  và  $a$ , nên hệ phương trình (3.16) trở nên gần suy biến với bậc khai triển cao. Với hệ phương trình như vậy, phương pháp khử Gauss cho kết quả không tốt, và như đã nói ở trên áp dụng phương pháp phân tích SVD sẽ cho kết quả với sai số nhỏ hơn.



Tuy nhiên, chúng ta vẫn có thể cải tiến để tăng độ chính xác của các tính toán xấp xỉ. Đó là chọn tham số “gần 0” thích hợp trong phương pháp SVD và chọn bán kính  $b$  nào là tốt. Trong [13], các tác giả đã tìm ra bán kính  $b = 6,0$  bằng thực nghiệm với bậc khai triển  $p = 5,0$  khi các hạt nằm trong một hình hộp lập phương với kích thước bằng 1,0. Anderson ([2]) cũng đã xác định bán kính nhỏ  $a$  thích hợp là khoảng 0,4. Trong khóa luận này, chúng tôi sẽ đưa ra cách khảo sát bằng thực nghiệm để nghiên cứu ảnh hưởng của tham số “gần 0” đối với độ chính xác trong tính toán lực xấp xỉ.

### **3.3 Tổng kết chương**

Trong chương 3 của khóa luận, chúng ta đã tìm hiểu về phương pháp phân tích SVD, các áp dụng của SVD trong tính toán ma trận. Nếu một hệ phương trình tuyến tính là kì dị, hoặc gần kì dị theo nghĩa các giá trị kì dị của ma trận hệ số xấp xỉ giá trị 0, trong khi phương pháp khử Gauss không thể áp dụng hoặc cho kết quả với sai số lớn thì phương pháp SVD đưa ra vector nghiệm với sai số nhỏ nhất theo nghĩa bình phương tối thiểu.

Chương 3 cũng đã trình bày một phương pháp cài đặt FMM trên máy tính GRAPE đã được đưa ra trong [13], các cải tiến sử dụng khai triển inner  $P^2M^2$ , và cách áp dụng phương pháp phân tích SVD để tính phân phối khối lượng của các giả hạt.

Tuy nhiên, vẫn có thể cải tiến cài đặt này hơn nữa. Đó là vấn đề chọn tham số “gần 0” trong SVD trong biến đổi A2P. Vấn đề này sẽ được nghiên cứu qua thực nghiệm.

## Chương 4. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 4.1 Môi trường thực nghiệm

#### 4.1.1 Phần cứng

Máy Pentium IV, chip 2.8 GHz, Ram 512 MB

#### 4.1.2 Phần mềm

Các thử nghiệm được thực hiện bằng ngôn ngữ lập trình C trên hệ điều hành Linux, phiên bản Fedora Core 3. Ưu điểm của ngôn ngữ lập trình C là: đây là một ngôn ngữ rất mạnh với thư viện toán học phong phú, thích hợp cho tính toán khoa học. Để trình bày các kết quả thử nghiệm, chúng tôi sử dụng chương trình *gnuplot*, một chương trình phần mềm mã nguồn mở được phát triển bởi tổ chức GNU, dùng để vẽ đồ thị hàm số, trình bày kết quả thực nghiệm dưới dạng đồ thị, biểu bảng trong không gian 2 và 3 chiều. Ngoài ra chúng tôi có sử dụng lại cài đặt phương pháp FMM do Chau, Kawai, Ebisuzaki [13] thực hiện.

**Bảng 3: Công cụ sử dụng trong thử nghiệm**

STT	Tên phần mềm (công cụ) sử dụng	Nguồn
1	Ngôn ngữ lập trình C	
2	Hệ điều hành Linux Fedora Core 3	<a href="http://fedora.redhat.com/">http://fedora.redhat.com/</a>
3	Chương trình Gnuplot	<a href="http://gnuplot.info/">http://gnuplot.info/</a>
4	Cài đặt thuật toán FMM	Tài liệu tham khảo [13]

### 4.2 Thử nghiệm phương pháp khai triển đa cực nhanh FMM

#### 4.2.1 Thời gian tính toán của phương pháp FMM

Để thử nghiệm phương pháp FMM, chúng tôi cho số hạt trong hệ thay đổi, sau đó ghi lại thời gian tính toán trong mỗi bước mô phỏng, và so sánh với phương pháp tính toán trực tiếp tương tác hạt-hạt. Trong thử nghiệm cấp khai triển sử dụng cấp 4.

Kết quả thu được như trong bảng sau. Ký hiệu mK có nghĩa là số hạt  $N = m \cdot 1024$  hạt.

**Bảng 4: Thời gian tính toán của FMM với số hạt thay đổi**

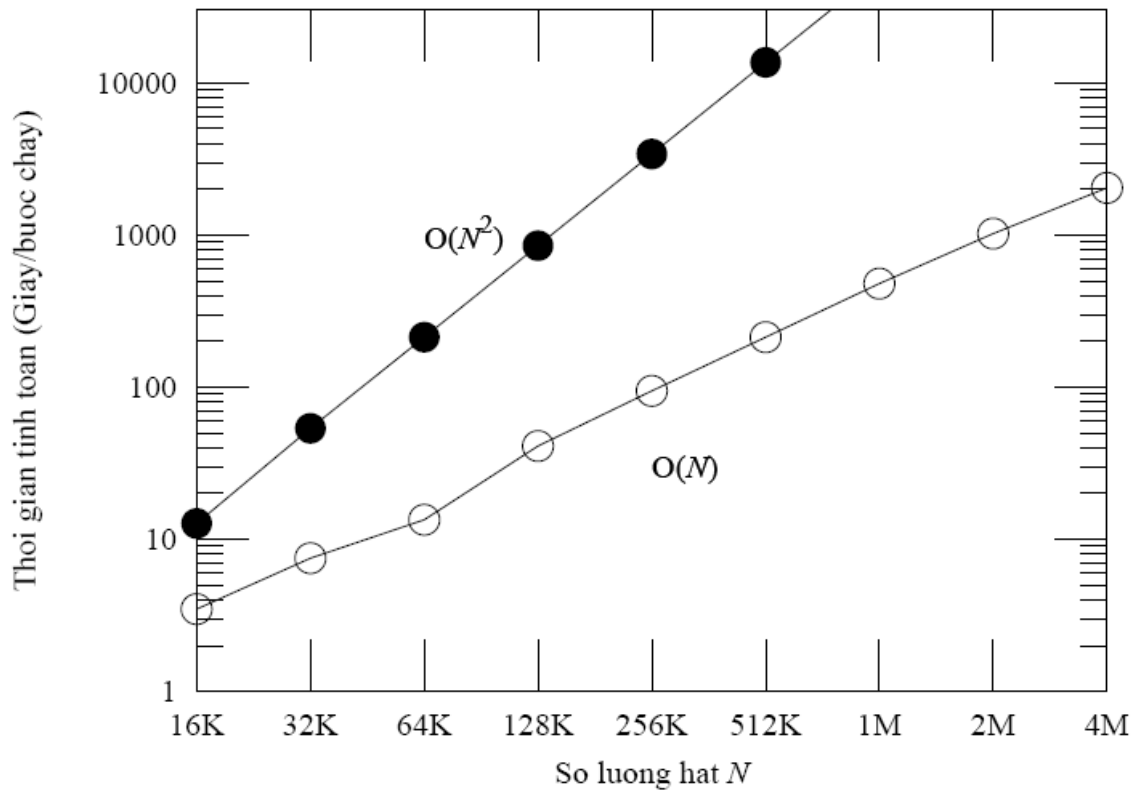
Số hạt (N)	Tổng (s)
1K	0,077240
2K	0,232300
4K	0,544597
8K	1,137916
16K	3,472888
32K	7,503617
64K	13,427825
128K	40,955762
256K	94,450049
512K	213,114363
1024K	479,507316
2048K	1021,350584
4096K	2042,701169

**Bảng 5: Thời gian tính toán trực tiếp với số hạt thay đổi**

Số hạt (N)	Thời gian tính toán (s)
1K	0,05956
2K	0,20723
4K	0,79719
8K	3,18779
16K	12,69657
32K	53,63753
64K	213,04669
128K	852,18676
256K	3408,74704
512K	13634,98816
1024K	54539,95264
2048K	218159,81056
4096K	872639,24224

Vì lý do thời gian (Thời gian cho thử nghiệm với số hạt lớn rất dài), số hạt thử nghiệm trong phương pháp FMM chỉ giới hạn đến 512K hạt, và số hạt thử nghiệm trong phương pháp tính toán trực tiếp chỉ đến 64K hạt. Các số liệu đối với số lượng hạt lớn hơn, được đánh giá trên cơ sở lý thuyết: thuật toán FMM có độ phức tạp  $O(N)$  và thuật toán tính lực trực tiếp có độ phức tạp là  $O(N^2)$ .

Đồ thị sau đây so sánh thời gian tính lực của hai thuật toán.



**Hình 13: Thời gian tính lực của thuật toán trực tiếp (trên) và FMM (dưới)**

#### 4.2.2 Đánh giá kết quả

Từ kết quả tính toán trên có thể rút ra nhận xét:

- Với số lượng hạt nhỏ, thời gian tính toán giữa hai phương pháp FMM và phương pháp tính toán trực tiếp là chênh lệch không đáng kể. Trong trường hợp số hạt ít hơn 5000 hạt, hai phương pháp gần như tương đương về thời gian tính toán.
- Với số lượng hạt lớn, thời gian tính toán của FMM tốt hơn thời gian tính toán của phương pháp trực tiếp với cùng số hạt. Nếu thực thi trên máy tính chuyên dụng GRAPE, thời gian tính toán của FMM có thể nhanh hơn từ 100 đến 1000 lần so với phương pháp trực tiếp.

### 4.3 Thử nghiệm phương pháp SVD trong biến đổi A2P

#### 4.3.1 Độ chính xác của khai triển inner $P^2M^2$ và biến đổi A2P

Như đã trình bày ở chương 3, việc tính các lực bởi các hạt ở xa dùng công thức đạo hàm (3.14) là phức tạp và chiếm nhiều thời gian tính toán. Vì vậy thay vì dùng công thức đạo hàm này, chúng ta dùng biến đổi A2P với công thức khai triển inner  $P^2M^2$  (công thức 2.38). Biến đổi A2P có ưu điểm là dễ cài đặt, bằng biến đổi này pha L2L có thể được thực hiện trên máy tính chuyên dụng GRAPE. Một vấn đề cần nghiên cứu là độ chính xác của biến đổi A2P và khai triển inner  $P^2M^2$ . Ở đây chúng tôi đưa ra kết quả thử nghiệm để đánh giá độ chính xác của các biến đổi này.

##### a. Phương pháp thực nghiệm

Các bước chính của trong thử nghiệm là:

1. Đặt một hạt L tại tọa độ  $(r, \pi, \pi/2)$ , hạt L có khối lượng  $m = 1,0$ . Cho  $r$  chạy từ 1,0 đến 10,0.
2. Lấy phân phối chuẩn hóa của các hạt trong hình lập phương đơn vị có tâm ở gốc tọa độ. Trong thử nghiệm này, chúng tôi sử dụng số hạt là 1000 hạt.
3. Tính toán thế năng, lực gây ra bởi hạt L tại 1000 hạt trên bằng phương pháp tính toán trực tiếp.
4. Tính các giá trị thế năng gây ra bởi hạt L tại các vị trí được định nghĩa bởi  $t$ -design cầu trên bề mặt của một hình cầu có bán kính  $a = 0,4$  có tâm tại gốc tọa độ. Số lượng các điểm cần tính toán phụ thuộc vào cấp khai triển  $p$ . Trong thử nghiệm này, bậc khai triển lần lượt được thử từ 1 đến 8.
5. Sử dụng biến đổi A2P cho khai triển cục bộ thu được ở bước 4, tức là sử dụng công thức (3.16) trong chương 3 để thu được các khối lượng của các giả hạt  $M_j$  trên bề mặt của một hình cầu có bán kính  $b = 6$  và có tâm ở gốc tọa độ. Số lượng và vị trí của các giả hạt phụ thuộc vào bậc khai triển  $p$ .
6. Tính toán lực và thế năng gây ra bởi các giả hạt tại 1000 hạt được phân phối trong bước 2.

7. So sánh kết quả với lực và thế năng tính bằng phương pháp trực tiếp. Đưa ra sai số theo nghĩa trung bình bình phương (root mean square). Sai số trung bình bình phương này được tính bằng công thức:

$$\varepsilon_{\phi} = \sqrt{\sum_i \left| \frac{p_i - p'_i}{p_i} \right|^2} \quad (4.1)$$

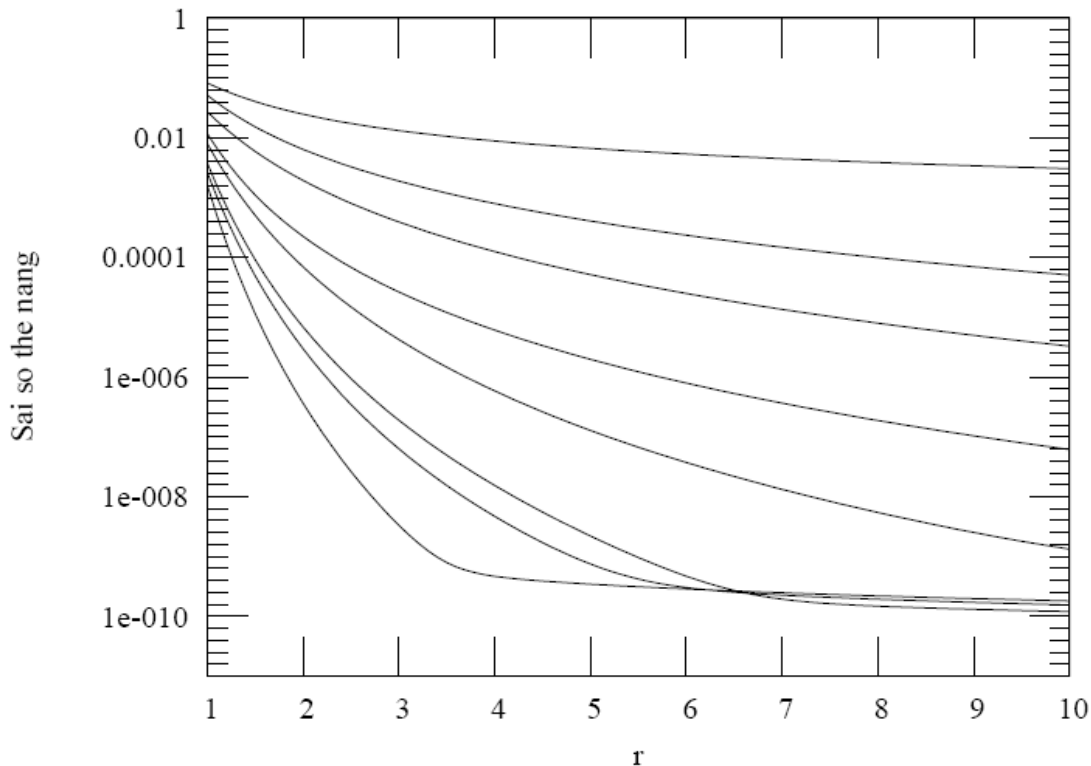
$$\varepsilon_F = \sqrt{\sum_i \frac{|\vec{f}_i - \vec{f}'_i|^2}{|\vec{f}_i|^2}} \quad (4.2)$$

trong đó  $\varepsilon_{\phi}$ ,  $\varepsilon_F$  là các sai số trung bình bình phương của thế năng và lực tính bằng phương pháp xấp xỉ;  $p_i$ ,  $\vec{f}_i$ ,  $p'_i$ ,  $\vec{f}'_i$  lần lượt là thế năng và lực được tính bằng phương pháp trực tiếp và tính xấp xỉ.

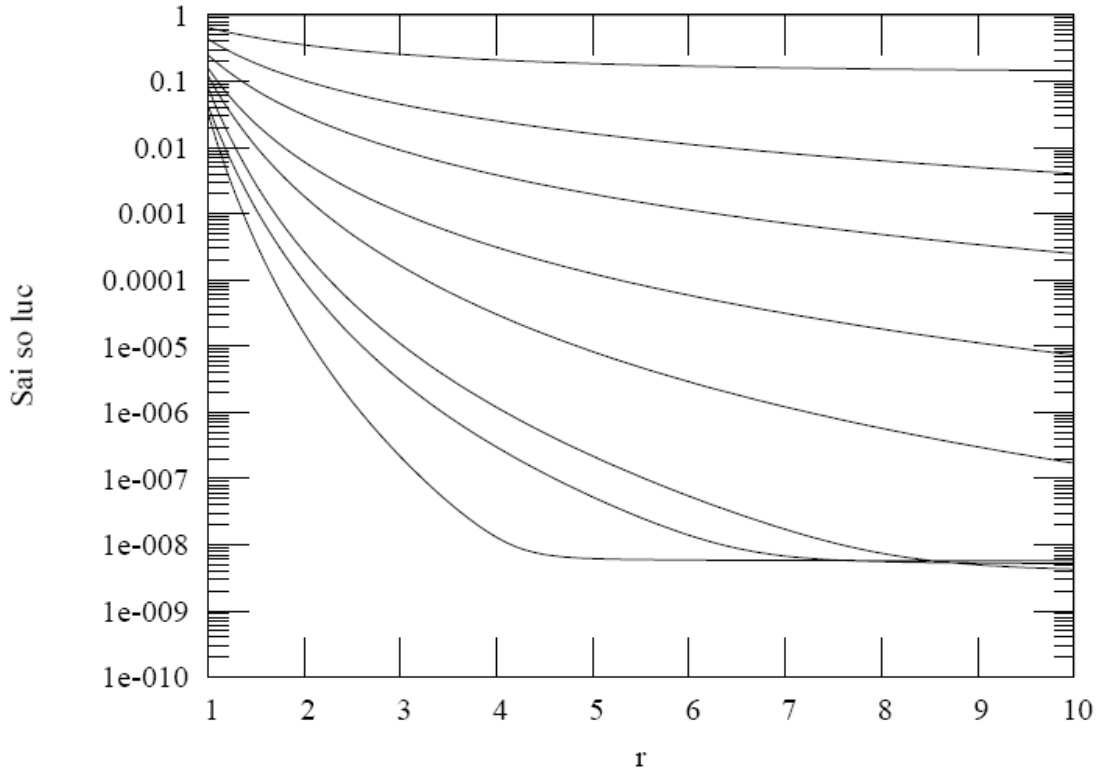
8. Vẽ đồ thị liên hệ giữa  $r$  và độ chính xác của thế năng và lực trong phương pháp tính xấp xỉ.

### b. Kết quả thực nghiệm

Sai số trung bình bình phương của thế năng và lực tương ứng được vẽ trên hình 14 và 15.



**Hình 14:** Sai số trung bình bình phương của thế năng được tính bằng khai triển inner  $P^2M^2$  và biến đổi A2P. Từ trên xuống, 8 đường cong tương ứng với các bậc khai triển  $p = 1, 2, 3, 4, 5, 6, 7, 8$



**Hình 15:** Sai số trung bình bình phương của lực được tính bằng khai triển inner  $P^2M^2$  và biến đổi A2P. Từ trên xuống, 8 đường cong tương ứng với các bậc khai triển  $p = 1, 2, 3, 4, 5, 6, 7, 8$

Từ hình 14, 15, có thể thấy với bậc khai triển  $p = 7$  và  $p = 8$ , sai số bắt đầu ngừng giảm khi  $r \geq 6$ .

#### 4.3.2 Ảnh hưởng của tham số gần không trong phương pháp SVD đến độ chính xác của thuật toán FMM

Như đã trình bày trong phương pháp SVD, việc chọn tham số gần 0 sẽ ảnh hưởng đến độ chính xác trong việc tính nghiệm xấp xỉ. Việc chọn tham số gần 0 phụ thuộc vào ứng dụng. Vì vậy chúng tôi xác định tham số gần 0 theo thực nghiệm.

##### a. Phương pháp thực nghiệm

Trong phương pháp SVD, số điều kiện được định nghĩa là tỉ số giữa số  $w_j$  lớn nhất và số  $w_j$  bé nhất, trong đó  $w_j$  là các phần tử trên đường chéo của ma trận chéo

$W$  thu được từ phân tích SVD. Nếu số điều kiện quá lớn, theo nghĩa nghịch đảo của nó xấp xỉ với độ chính xác cho phép trong tính toán dấu phẩy động của máy tính (xấp xỉ  $10^{-6}$  với độ chính xác đơn và  $10^{-12}$  với độ chính xác kép), ma trận  $A$  được gọi là có điều kiện xấu. Xét hệ cho dưới dạng ma trận:

$$Ax = b$$

trong đó  $A$  là ma trận vuông,  $x$  và  $b$  là các vector.

Trong trường hợp ma trận  $A$  có điều kiện xấu, cách tiếp cận chúng tôi sử dụng trong thực nghiệm là xác định trước một tham số “gần 0”  $\varepsilon$ . Với các số  $w_j$  sao cho  $w_j / \max < \varepsilon$ , ta gán  $w_j = 0$ . Ở đây  $\max$  là giá trị lớn nhất trong các số trên đường chéo của ma trận  $W$  trong phân tích SVD của ma trận  $A$ . Sau khi xử lý các giá trị này, gọi thủ tục *svbksb* (xem chi tiết trong phụ lục A) để giải hệ phương trình.

Các bước thử nghiệm để tìm tham số gần 0 thích hợp như sau:

Từ bước 1 đến bước 4, tương tự như trong phần (4.3.1 a), trong bước 1 chúng tôi chọn bán kính  $r = 6,0$ , trong bước 4 cấp khai triển là từ 1 đến 10.

Trong bước 5, trong phần tính khối lượng của các giả hạt trên hình cầu bán kính  $b = 6,0$ , trước khi gọi thủ tục *svbksb()*, chúng tôi xử lý các giá trị kỳ dị gần không với các tham số gần 0 thay đổi từ  $\varepsilon = 10^{-6}$  giảm dần đến  $10^{-14}$ . Kết quả tính sai số trong bước 7 được ghi vào file để phân tích. Sau đó vẽ đồ thị liên hệ giữa tham số gần 0 với sai số tính toán.

## b. Kết quả thực nghiệm

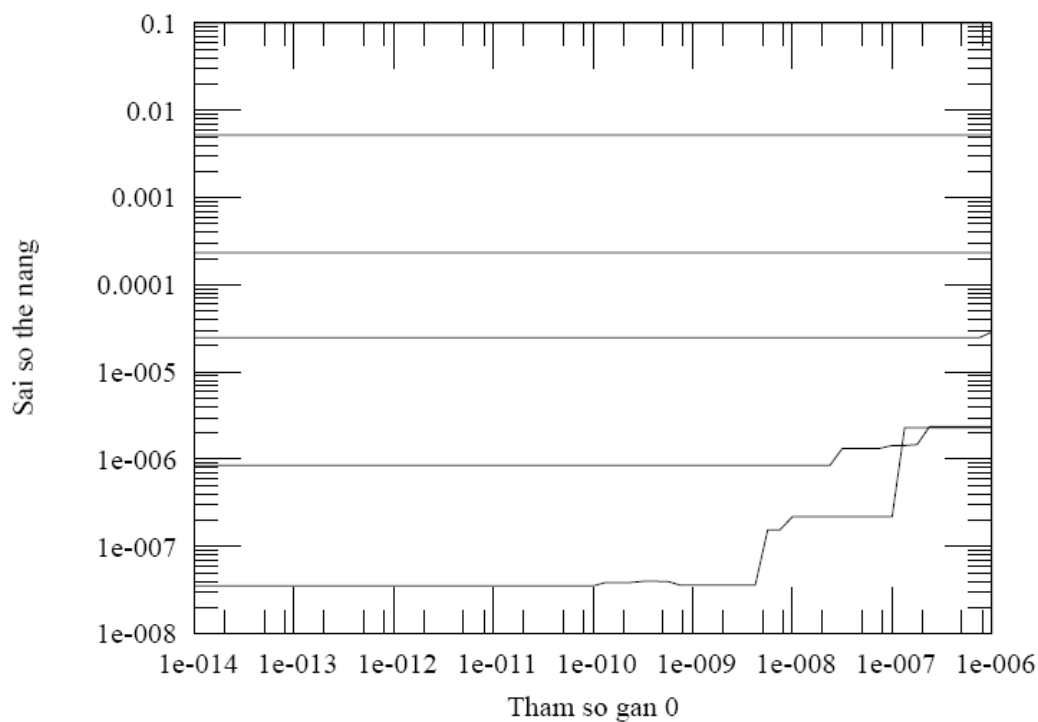
Hình 16, 17, 18, 19 thể hiện mô tả quan hệ giữa tham số gần 0 với sai số của tính thế năng và tính lực xấp xỉ.

Từ kết quả thực nghiệm ta có nhận xét:

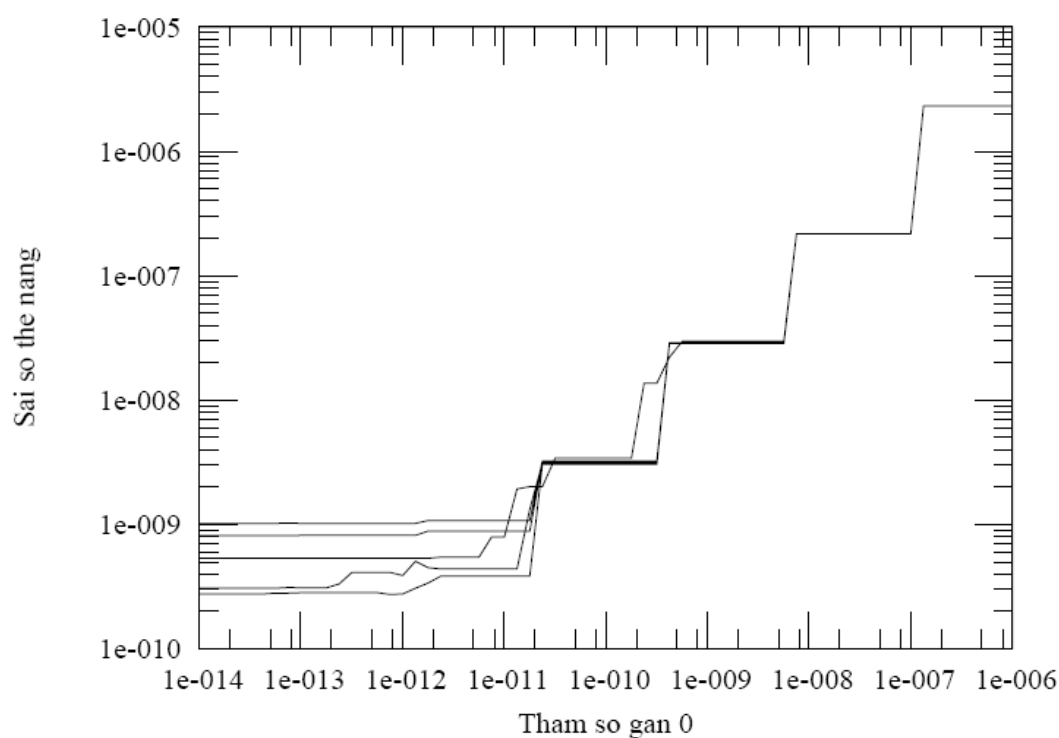
- Với các bậc khai triển thấp (bậc khai triển nhỏ hơn 4), chọn tham số gần 0 không ảnh hưởng đến độ chính xác tính toán.
- Với bậc khai triển cao, thì tham số gần 0 càng nhỏ để thu được độ chính xác tốt. Nếu chọn tham số gần 0 quá nhỏ (khoảng  $1,0e-14$ ), độ chính xác không thay đổi so với phương pháp SVD khi chưa xử lý các giá trị kỳ dị gần 0.

Bảng 6 thể hiện các tham số gần 0 được chọn với từng bậc khai triển. Tham số gần 0 được chọn với bậc khai triển từ 4 trở lên.

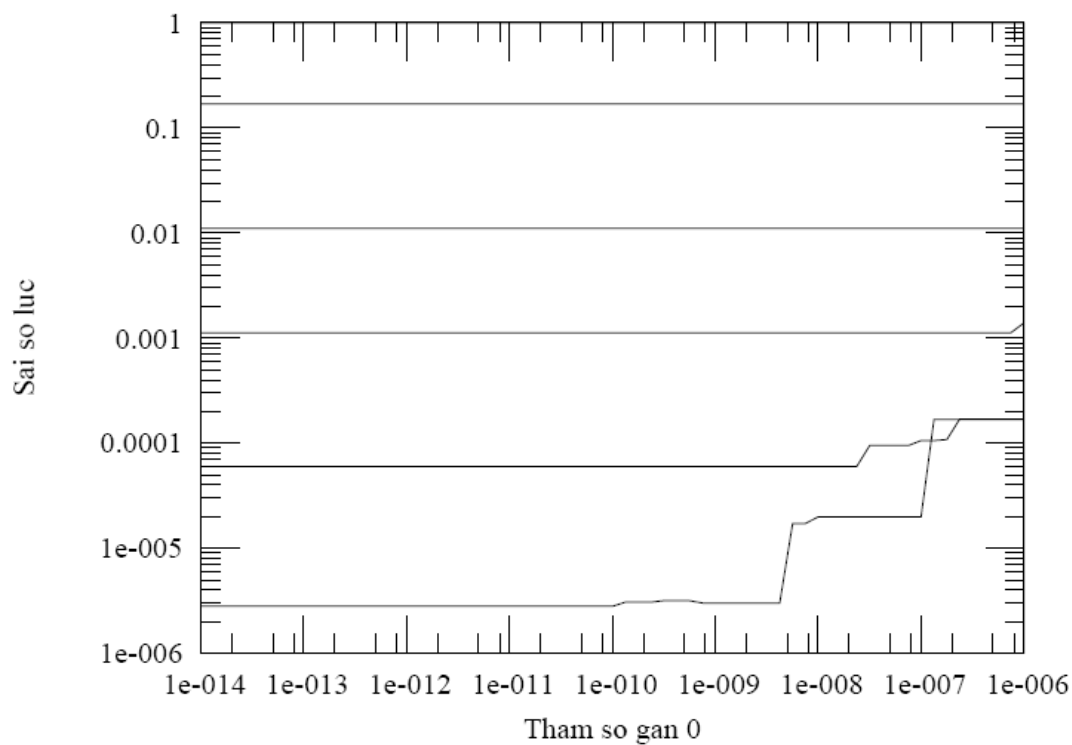




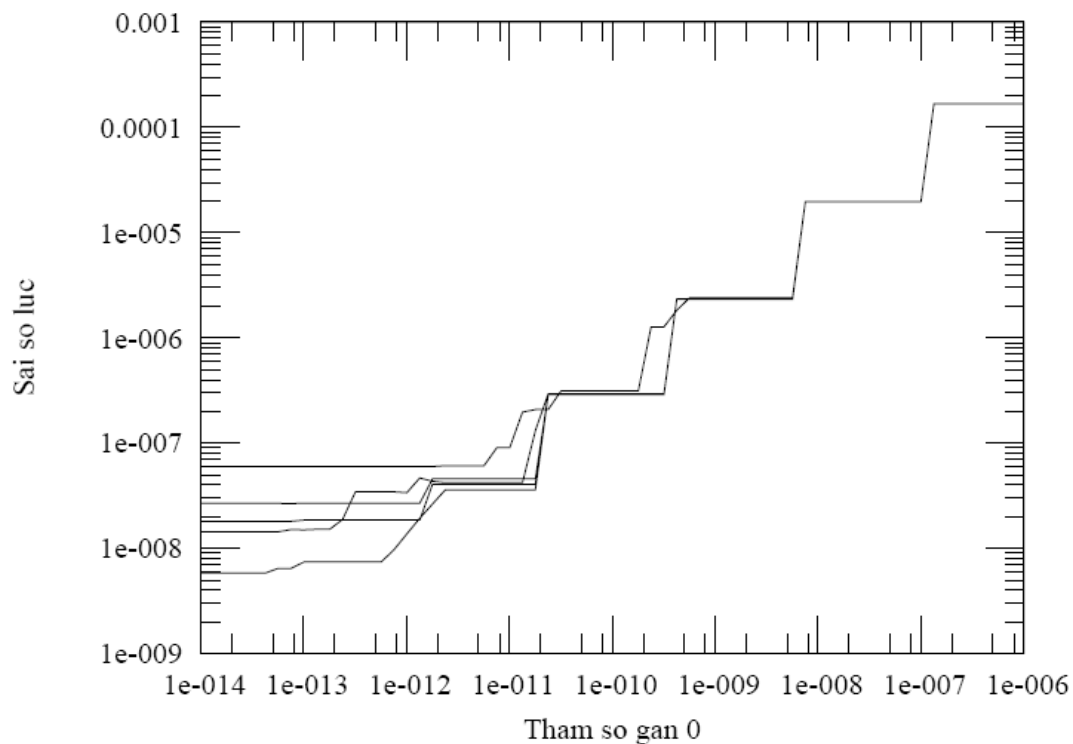
**Hình 16: Sai số trung bình bình phương của thể năng ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 1 đến 5**



**Hình 17 : Sai số trung bình bình phương của thể năng ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 6 đến 10**



**Hình 18: Sai số trung bình bình phương của lực ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 1 đến 5**



**Hình 19 : Sai số trung bình bình phương của lực ứng với các tham số gần 0 khác nhau. Từ trên xuống, các đường tương ứng với bậc khai triển từ 6 đến 10**

**Bảng 6: Tham số gần 0 ứng với các mức khai triển khác nhau**

Bậc khai triển	Tham số gần 0
4	2,0e-08
5	4,0e-09
6	5,0e-12
7	2,0e-13
8	2,0e-13
9	2,0e-13
10	2,0e-13

#### 4.4 Tổng kết chương

Trong chương 4 của khóa luận, chúng tôi đã trình bày các thực nghiệm của mình trong việc đánh giá hiệu năng của thuật toán FMM, độ chính xác của khai triển inner  $P^2M^2$  và biến đổi A2P có sử dụng phương pháp SVD. Ảnh hưởng của phương pháp SVD đến độ chính xác của thuật toán cũng đã được khảo sát. Các tham số “gần 0” đã được tìm bằng thực nghiệm. Tuy nhiên do những hạn chế về phần cứng, các thử nghiệm còn bị giới hạn về số lượng hạt trong hệ. Nếu có điều kiện, chúng tôi hy vọng sẽ được thử nghiệm trên các máy chuyên dụng cho mô phỏng động lực phân tử để đánh giá kết quả với số lượng hạt lớn hơn.

## KẾT LUẬN

### Kết quả đạt được

Trong khóa luận, chúng tôi đã hệ thống hóa một số vấn đề lý thuyết trong bài toán mô phỏng động lực phân tử, các cách tiếp cận trong bài toán tăng tốc độ tính lực trong mô phỏng động lực phân tử. Các nghiên cứu gần đây về phương pháp khai triển đa cực nhanh FMM như các cải tiến của Anderson, Makino cũng đã được trình bày trong khóa luận. Chúng tôi đã nghiên cứu phương pháp phân tích SVD và cách áp dụng phương pháp SVD trong biến đổi A2P và khai triển inner  $P^2M^2$ . Trong khóa luận chúng tôi cũng đưa ra phương pháp đánh giá về độ chính xác của phương pháp biến đổi A2P và khai triển inner  $P^2M^2$ .

Sau một quá trình tìm hiểu và nghiên cứu chúng tôi đã đưa ra được cải tiến trong vấn đề áp dụng phương pháp SVD, đó là việc xử lý các giá trị kì dị gần không để tăng độ chính xác trong tính lực xấp xỉ. Kết quả cho thấy đối với cấp khai triển cao, việc áp dụng cải tiến này cho kết quả tốt.

Cũng trong quá trình nghiên cứu, chúng tôi đã thu được những kiến thức và kỹ năng quý báu về sử dụng ngôn ngữ lập trình C, tính toán dấu phẩy động và kỹ năng sử dụng hệ điều hành Linux.

### Hướng phát triển

Mã cài đặt của thuật toán FMM mà chúng tôi có được viết trên ngôn ngữ C nên rất khó bảo trì và nâng cấp. Vì vậy trong thời gian tới, chúng tôi dự định sẽ chuyển mã này sang ngôn ngữ hướng đối tượng C++.

Trong phương pháp biến đổi A2P, còn một cải tiến khác nữa là thay đổi vị trí các điểm trong  $t$ -design cầu. Chúng tôi đã nghiên cứu cải tiến này, nhưng do những hạn chế về mặt thời gian và lý thuyết nên kết quả thu được chưa tốt. Chúng tôi sẽ tiếp tục nghiên cứu về mặt lý thuyết và thực hành cải tiến này. Ngoài ra, chúng tôi sẽ tìm hiểu thêm về các biến thể của thuật toán FMM đang được nghiên cứu trong thời gian gần đây, như cải tiến của Ying [21, 22] và tiến hành song song hóa thuật toán FMM.

Các ứng dụng mô phỏng động lực phân tử ở trên thế giới là khá phổ biến nhưng ở Việt Nam là còn khá hạn chế. Vì vậy một hướng nghiên cứu tiếp theo của

chúng tôi đó là ứng dụng các phương pháp mô phỏng động lực phân tử để xây dựng các hệ mô phỏng vật lý, hóa học sử dụng trong nghiên cứu khoa học và trong giáo dục.

## Phụ lục A: Cài đặt SVD bằng ngôn ngữ C

### A1. Thủ tục svdcmp()

```
#include <math.h>
#define NRANSI
#include "nrutil.h"

void svdcmp(float **a, int m, int n, float w[], float **v)
{
    float pythag(float a, float b);
    int flag,i,its,j,jj,k,l,nm;
    float anorm,c,f,g,h,s,scale,x,y,z,*rv1;

    rv1=vector(1,n);
    g=scale=anorm=0.0;
    for (i=1;i<=n;i++) {
        l=i+1;
        rv1[i]=scale*g;
        g=s=scale=0.0;
        if (i <= m) {
            for (k=i;k<=m;k++) scale += fabs(a[k][i]);
            if (scale) {
                for (k=i;k<=m;k++) {
                    a[k][i] /= scale;
                    s += a[k][i]*a[k][i];
                }
                f=a[i][i];
                g = -SIGN(sqrt(s),f);
                h=f*g-s;
                a[i][i]=f-g;
                for (j=l;j<=n;j++) {
                    for (s=0.0,k=i;k<=m;k++)
                        s += a[k][i]*a[k][j];
                    f=s/h;
                    for (k=i;k<=m;k++)
                        a[k][j] += f*a[k][i];
                }
                for (k=i;k<=m;k++) a[k][i] *= scale;
            }
        }
        w[i]=scale*g;
        g=s=scale=0.0;
        if (i <= m && i != n) {
            for (k=l;k<=n;k++) scale += fabs(a[i][k]);
            if (scale) {
                for (k=l;k<=n;k++) {
                    a[i][k] /= scale;
                    s += a[i][k]*a[i][k];
                }
            }
        }
    }
}
```

```
f=a[i][1];
g = -SIGN(sqrt(s),f);
h=f*g-s;
a[i][1]=f-g;
for (k=1;k<=n;k++) rv1[k]=a[i][k]/h;
for (j=1;j<=m;j++) {
    for (s=0.0,k=1;k<=n;k++)
        s += a[j][k]*a[i][k];
    for (k=1;k<=n;k++)
        a[j][k] += s*rv1[k];
}
for (k=1;k<=n;k++) a[i][k] *= scale;
}
}
anorm=FMAX(anorm, (fabs(w[i])+fabs(rv1[i])));
}
for (i=n;i>=1;i--) {
    if (i < n) {
        if (g) {
            for (j=1;j<=n;j++)
                v[j][i]=(a[i][j]/a[i][1])/g;
            for (j=1;j<=n;j++) {
                for (s=0.0,k=1;k<=n;k++)
                    s += a[i][k]*v[k][j];
                for (k=1;k<=n;k++)
                    v[k][j] += s*v[k][i];
            }
            for (j=1;j<=n;j++) v[i][j]=v[j][i]=0.0;
        }
        v[i][i]=1.0;
        g=rv1[i];
        l=i;
    }
}
for (i=IMIN(m,n);i>=1;i--) {
    l=i+1;
    g=w[i];
    for (j=1;j<=n;j++) a[i][j]=0.0;
    if (g) {
        g=1.0/g;
        for (j=1;j<=n;j++) {
            for (s=0.0,k=1;k<=m;k++)
                s += a[k][i]*a[k][j];
            f=(s/a[i][i])*g;
            for (k=i;k<=m;k++) a[k][j] += f*a[k][i];
        }
        for (j=i;j<=m;j++) a[j][i] *= g;
    } else for (j=i;j<=m;j++) a[j][i]=0.0;
    ++a[i][i];
}
for (k=n;k>=1;k--) {
    for (its=1;its<=30;its++) {
        flag=1;
```

```
for (l=k;l>=1;l--) {
    nm=l-1;
    if ((float)(fabs(rv1[l])+anorm)== anorm)
    {
        flag=0;
        break;
    }
    if ((float)(fabs(w[nm])+anorm) == anorm)
        break;
}
if (flag) {
    c=0.0;
    s=1.0;
    for (i=1;i<=k;i++) {
        f=s*rv1[i];
        rv1[i]=c*rv1[i];
        if ((float)(fabs(f)+anorm)== norm)
            break;

        g=w[i];
        h=pythag(f,g);
        w[i]=h;
        h=1.0/h;
        c=g*h;
        s = -f*h;
        for (j=1;j<=m;j++) {
            y=a[j][nm];
            z=a[j][i];
            a[j][nm]=y*c+z*s;
            a[j][i]=z*c-y*s;
        }
    }
}
z=w[k];
if (l == k) {
    if (z < 0.0) {
        w[k] = -z;
        for (j=1;j<=n;j++)
            v[j][k] = -v[j][k];
    }
    break;
}
if (its == 30)
    nrerror("no convergence in 30 svdcmp
            iterations");

x=w[l];
nm=k-1;
y=w[nm];
g=rv1[nm];
h=rv1[k];
f=((y-z)*(y+z)+(g-h)*(g+h))/(2.0*h*y);
g=pythag(f,1.0);
f=((x-z)*(x+z)+h*((y/(f+SIGN(g,f)))-h))/x;
c=s=1.0;
```



```
        for (j=1;j<=nm;j++) {
            i=j+1;
            g=rv1[i];
            y=w[i];
            h=s*g;
            g=c*g;
            z=pythag(f,h);
            rv1[j]=z;
            c=f/z;
            s=h/z;
            f=x*c+g*s;
            g = g*c-x*s;
            h=y*s;
            y *= c;
            for (jj=1;jj<=n;jj++) {
                x=v[jj][j];
                z=v[jj][i];
                v[jj][j]=x*c+z*s;
                v[jj][i]=z*c-x*s;
            }
            z=pythag(f,h);
            w[j]=z;
            if (z) {
                z=1.0/z;
                c=f*z;
                s=h*z;
            }
            f=c*g+s*y;
            x=c*y-s*g;
            for (jj=1;jj<=m;jj++) {
                y=a[jj][j];
                z=a[jj][i];
                a[jj][j]=y*c+z*s;
                a[jj][i]=z*c-y*s;
            }
        }
        rv1[l]=0.0;
        rv1[k]=f;
        w[k]=x;
    }
    free_vector(rv1,1,n);
}
#undef NRANSI

float pythag(float a, float b)
/* Tính  $(a^2+b^2)^{1/2}$  mà không bị lỗi tràn số */
{
    float absa, absb;
    absa=fabs(a);
    absb=fabs(b);
    if (absa > absb)
        return absa*sqrt(1.0+SQR(absb/absa));
}
```

```
        else return (absb == 0.0 ? 0.0 :
absb*sqrt(1.0+SQR(absa/absb)));
}
```

## **A2. Thủ tục svbksb()**

```
#include "nrutil.h"
void svbksb(double **u, double w[], double **v, int m, int n,
           double b[], double x[])
{
    int jj,j,i;
    double s,*tmp;
    tmp=dvector(1,n);
    for (j=1;j<=n;j++) {
        s=0.0;
        if (w[j]) {
            for (i=1;i<=m;i++) s += u[i][j]*b[i];
            s /= w[j];
        }
        tmp[j]=s;
    }
    for (j=1;j<=n;j++) {
        s=0.0;
        for (jj=1;jj<=n;jj++) s += v[j][jj]*tmp[jj];
        x[j]=s;
    }
    free_dvector(tmp,1,n);
}
```

## **A3. Thủ tục zero\_small\_values()**

```
void zero_small_values(double* w, int n, double threshold) {
    int i;
    double wmin, wmax;
    wmax= 0.0;
    for(i=1; i<= n; i++)
        if( w[i] > wmax) wmax= w[i];
    //Giá trị threshold được chọn tùy từng ứng dụng
    wmin= wmax*threshold;
    for(i=1; i <= n; i++)
        if(w[i] < wmin) w[i]=0.0;
}
```

## TÀI LIỆU THAM KHẢO

- [1] Michael P. Allen, *Introduction to Molecular Dynamics Simulation*, John von Neumann Institute for Computing (2004), p. 1-28
- [2] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Stat. Comput. vol **13**, (1992), p. 923-947
- [3] A. W. Appel, *An efficient program for many-body simulation*. SIAM J. Sci. Stat. Comput. vol. **16**, n. 1, (1985), p. 85-103
- [4] J. Barnes and P. Hut, *A hierarchical  $O(N\log N)$  force calculation algorithm*, Nature **324**, (1986), p. 446-449
- [5] Rick Beatson, Leslie Greengard, *A short course on fast multipole methods*, Oxford University Press, (1997)
- [6] G. Beylkin, R. Coifman, and V. Rokhlin. *The fast wavelet transform and numerical algorithms*. Comm. Pure and Appl. Math, (1991), **44**, p. 141-183
- [7] L. Greengard, V. Rokhlin, *A fast Algorithm for Particle Simulations*, Journal of Computational Physics (ISSN 0021-9991), vol. **73**, Dec. 1987, p. 325-348
- [8] L. Greengard, V. Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimension*, Acta Numerica 6 (1997), p. 229-269
- [9] W. Hackbush, Z. P. Nowak, *On the fast matrix multiplication in the boundary element method by panel clustering*, Numer. Math., **54**, (1989), p. 463-491
- [10] R. H. Hardin and N. J. Sloane, *MacLaren's improved snub cube and other new spherical designs in three dimensions*, Discrete and Computational Geometry, **15**, (1996), p. 429-441
- [11] Pieter Heres, *An introduction to Fast Multipole Method*, Seminar Scientific Computing, March 6, 2002
- [12] A. Kawai, J. Makino, *A simple Formulation of the Fast Multipole Method: Pseudo-Particle Multipole Method*, eprint arXiv:astro-ph/9812431, (1998), p. 1-13

- [13] N. H. Chau, A. Kawai, T. Ebisuzaki, *A new implementation of fast multipole algorithm on special-purpose computer MDGRAPE-2*, Proceeding of Annual Meeting of Molecular Simulation Society of Japan, Niigata, Dec 16-18, 2002
- [14] A. Kawai, J. Makino, T. Ebisuzaki, *Performance analysis of high-accuracy tree code based on the pseudoparticle multipole method*, The Astrophysical Journal Supplement **151** (2004), p. 13-33
- [15] J. Makino and M. Taiji, *Special purpose computer for scientific simulations – the GRAPE system*, John Wiley and Sons, Chichester, 1998
- [16] J. Makino, *Yet another fast multipole method without multipoles-pseudoparticles multipole method*, Journal of Computational Physics, **151** (1999), p. 910-920
- [17] Thomas Melzer, *SVD and its Application to Generalized Eigenvalue problems*, June 8, 2004, p. 1-15
- [18] William. H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery *Numerical in C, The Art of Scientific computing Second Edition*, Cambridge University Press, 1988-1992, chapter 15
- [19] Robin Schoemaker, *An implementation of the Fast Multipole Methods without multipoles*, Seminar Computing Group Seminar, Spring 2002
- [20] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, and M. Umemura, *A special-purpose computer for gravitational many-body problems*, Nature, 345 (1990), p. 33-35
- [21] L. Ying, G. Biros, D. Zorin, H. Langston, *A new parallel kernel-independent fast multipole method*, Proceedings of the ACM/IEEE SC2003, Phoenix, Arizona, USA, November, 2003, p. 15-21
- [22] L. Ying, G. Biros, D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, Journal of Computational Physics, **196** (2004) p. 591-626
- [23] David C. Young, *Computational Chemistry: A practice guide for Applying Techniques to Real-World Problems*, John Willey & Sons. Inc (2001), p. 60-62
- [24] Website: <http://www.amara.com/papers/nbody.html>