

**ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KHOA HỌC**

LƯƠNG TRIỀU DUY

**PHƯƠNG PHÁP LẬP SONG SONG
GIẢI MỘT SỐ BÀI TOÁN BIÊN
DỰA TRÊN CHIA MIỀN**

**Chuyên ngành: Toán ứng dụng
Mã số: 60.46.01.12**

LUẬN VĂN THẠC SĨ TOÁN HỌC

Người hướng dẫn khoa học: TS. VŨ VINH QUANG

Thái Nguyên - 2012

MỤC LỤC

LỜI CẢM ƠN.....	1
LỜI NÓI ĐẦU	2
Chương 1: CÁC KIẾN THỨC CƠ BẢN VỀ XỬ LÝ SONG SONG VÀ GIẢI SỐ PHƯƠNG TRÌNH ĐẠO HÀM RIÊNG.....	4
1.1. Các kiến thức cơ bản về lý thuyết xử lý song song	4
1.1.1 Định nghĩa.....	4
1.1.2 Đánh giá các chương trình song song.....	5
1.1.3 Thuật toán song song	7
1.1.4 Các cách tiếp cận trong thiết kế	8
1.1.5 Phân tích và đánh giá thuật toán song song.....	8
1.2 Các kiến thức cơ bản về giải số phương trình đạo hàm riêng.....	11
1.2.1 Phương pháp sai phân	11
1.2.2 Thuật toán thu gọn khối lượng tính toán	14
1.3 Giới thiệu thư viện TK2004	24
1.3.1 Bài toán biên Diricchlet.....	24
1.3.2 Bài toán biên hỗn hợp.....	26
Chương 2: PHƯƠNG PHÁP LẬP SONG SONG DỰA TRÊN TƯ TƯỞNG CHIA MIỀN	29
2.1 Cơ sở của phương pháp chia miền	29
2.2 Một số thuật toán chia miền	30
2.2.1 Thuật toán chia miền Patrick le Talle.....	30
2.2.2 Thuật chia miền J.R.Rice, E.A. Vavalis, Daopi Yang.	32
2.2.3. Thuật toán chia miền Saito – Fujita.....	34
2.3 Các sơ đồ lập song song dựa trên chia miền.....	36
2.3.1 Phương pháp lập chắn lẻ QD1.....	36
2.3.2 Phương pháp song song QD2.....	43

2.4 Mô hình bài toán biên gián đoạn.....	47
2.4.1 Phương pháp lặp tuần tự	48
2.4.2 Phương pháp lặp chặn lẻ QD3	51
PHẦN KẾT LUẬN.....	55
TÀI LIỆU THAM KHẢO	56

DANH MỤC CÁC BẢNG

Bảng 1: $u^*(x, y) = e^x \log(y + 5) - \sin y \log(x + 6)$	40
Bảng 2: $u^*(x, y) = x^3 + ye^x + y^3 + xe^{-y}$	41
Bảng 3: $u^*(x, y) = \sin x \sin y$	42
Bảng 4: $u^*(x, y) = e^x \log(y + 5) - \sin y \log(x + 6)$	46
Bảng 5: $u^*(x, y) = x^3 + ye^x + y^3 + xe^{-y}$	46
Bảng 6: Kết quả thực nghiệm của ví dụ 1	49
Bảng 7: Kết quả thực nghiệm của ví dụ 2	50

DANH MỤC CÁC HÌNH

Hình 1	29
Hình 2	36
Hình 3: Đồ thị nghiệm xấp xỉ.....	40
Hình 4: Đồ thị nghiệm xấp xỉ.....	41
Hình 5: Đồ thị nghiệm xấp xỉ.....	42
Hình 6	43
Hình 7: Đồ thị nghiệm xấp xỉ trong ví dụ 1	49
Hình 8: Đồ thị nghiệm xấp xỉ trong ví dụ 2	50

LỜI CẢM ƠN

Luận văn được thực hiện và hoàn thành tại trường Đại học khoa học- Đại học Thái Nguyên dưới sự hướng dẫn của TS. Vũ Vinh Quang. Qua đây tác giả xin chân thành cảm ơn sâu sắc đến thầy Vũ Vinh Quang, người đã đưa đề tài và tận tình hướng dẫn trong suốt quá trình nghiên cứu của tôi.

Đồng thời tác giả xin được gửi lời cảm ơn đến các thầy cô trong Khoa Toán - Tin trường Đại học Khoa học - Đại học Thái nguyên đã tạo mọi điều kiện thuận lợi để tác giả hoàn thành bản luận văn này. Tác giả cũng gửi lời cảm ơn đến gia đình, các bạn bè trong lớp Cao học K4C đã động viên giúp đỡ tác giả trong suốt quá trình học tập và làm luận văn.

Tác giả

LỜI NÓI ĐẦU

Trong trường hợp tổng quát, việc tìm nghiệm đúng của lớp các bài toán biên mà chủ yếu là phương trình elliptic cấp hai trong trường hợp điều kiện biên hỗn hợp hoặc hỗn hợp mạnh là không thực hiện được. Việc nghiên cứu giải gần đúng các bài toán biên mà tiêu biểu là các bài toán được biểu diễn bằng các phương trình cấp hai đã và đang là một lĩnh vực rất quan tâm của các nhà toán học. Trong nhiều năm qua đã có nhiều công trình nghiên cứu về lĩnh vực này, mục đích chính của các phương pháp là đưa các bài toán vi phân về các bài toán sai phân để tìm nghiệm xấp xỉ thông qua các thuật toán giải các hệ phương trình đại số. Tuy nhiên khi bài toán biên elliptic với các điều kiện biên phức tạp hoặc hàm nghiệm hoặc đạo hàm của nghiệm gián đoạn trong miền đang xét thì các phương pháp truyền thống sẽ gặp khó khăn. Để giải quyết bài toán trong các trường hợp này thì một trong những phương pháp được biết đến là phương pháp lặp với độ phức tạp tính toán thấp trên tư tưởng của phương pháp chia miền và dựa trên phương pháp này, hướng nghiên cứu hiện nay đang rất phát triển đó là nghiên cứu các sơ đồ tính toán song song giải quyết các mô hình bài toán phức tạp.

Luận văn đặt vấn đề tìm hiểu một số phương pháp lặp giải bài toán elliptic cấp hai trên tư tưởng của thuật toán chia miền. Trên cơ sở các thuật toán chia miền, luận văn đề xuất việc xây dựng một số sơ đồ tính toán song song. Luận văn cấu trúc gồm 2 chương.

Chương 1: Đưa ra các kiến thức cơ bản về lý thuyết tính toán song song, phương pháp sai phân và thuật toán thu gọn khối lượng tính toán và hệ thống thư viện TK2004 giải số các bài toán biên elliptic.

Chương 2: Trình bày tóm tắt cơ sở của phương pháp chia miền. Trên cơ sở các phương pháp đã biết, luận văn đưa ra một số sơ đồ tính toán song song

dựa trên tư tưởng chia miền giải bài toán biên elliptic cấp hai. Tiến hành đánh giá độ phức tạp của thuật toán, cài đặt thử nghiệm trên máy tính điện tử để kiểm tra sự hội tụ của các sơ đồ.

Các kết quả thực nghiệm trong luận văn được cài đặt bằng ngôn ngữ Matlab trên máy tính tuần tự PC.

Chương 1

CÁC KIẾN THỨC CƠ BẢN VỀ XỬ LÝ SONG SONG

VÀ GIẢI SỔ PHƯƠNG TRÌNH ĐẠO HÀM RIÊNG

Trong chương này, luận văn sẽ trình bày một số kiến thức cơ bản về lý thuyết xử lý song song, phương pháp sai phân và thư viện TK2004. Đây là các kiến thức quan trọng sẽ được sử dụng trong các chương sau của luận văn. Các kiến thức này được tham khảo từ các tài liệu [1, 3, 4, 5, 6, 9, 10, 11].

1.1. Các kiến thức cơ bản về lý thuyết xử lý song song

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

Hiện nay vấn đề xử lý song song đã được hiện thực hóa nhờ những ưu điểm vượt trội của nó.

1. Hiện nay giá thành của phần cứng (CPU) giảm mạnh, tạo điều kiện để xây dựng những hệ thống có nhiều BXL với giá thành hợp lý.

2. Sự phát triển của công nghệ mạch tích hợp VLSI cho phép tạo ra những hệ phức hợp có hàng triệu transistor trên một chip.

3. Tốc độ xử lý của các BXL theo kiểu Von Neumann đã dần tiến tới giới hạn, không thể cải tiến thêm được do vậy dẫn tới đòi hỏi phải thực hiện xử lý song song .

Một trong các mục đích chính của xử lý song song là nghiên cứu và xây dựng những thuật toán thích hợp để cài đặt trên các máy tính song song, nghĩa là phát triển các thuật toán song song. Câu hỏi tự nhiên là đánh giá một thuật toán song song như thế nào được gọi là thích hợp cho xử lý song song? Đối với thuật toán tuần tự thì chúng ta có thể thống nhất cách đánh giá dựa vào

thời gian thực hiện thuật toán, không gian bộ nhớ và khả năng lập trình, v.v. Đánh giá thuật toán song song thì phức tạp hơn nhiều, ngoài những tiêu chuẩn trên còn phải bổ sung thêm những tham số về số BXL, khả năng của các bộ nhớ cục bộ, sơ đồ truyền thông, và các giao thức đồng bộ hoá, v.v.

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

1.1.2 Đánh giá các chương trình song song

Sau đây chúng ta đưa ra cơ sở của phương pháp đánh giá độ phức tạp của thuật toán song song.

Thời gian thực hiện song song

Để đánh giá được độ phức tạp tính toán của các thuật toán song song, ngoài số bước tính toán chúng ta còn cần đánh giá thời gian truyền thông của các tiến trình. Trong một hệ thống truyền thông, thời gian truyền thông cũng phải được xem trong thời gian thực hiện của thuật toán.

Thời gian thực hiện song song, ký hiệu là t_p gồm hai phần t_{comp} và t_{comm}

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

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

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

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

$$t_{\text{comm}} = t_{\text{startup}} + n * t_{\text{data}}$$

trong đó

+ t_{startup} là thời gian cần thiết để gửi những thông điệp không phải là dữ liệu. Nó bao gồm cả thời gian để đóng gói thông điệp ở nơi gửi và thời gian mở gói ở nơi nhận. Để đơn giản chúng ta giả thiết thời gian này là hằng số.

+ t_{data} là thời gian cần thiết để chuyển một từ dữ liệu (một mục dữ liệu) từ nơi gửi tới nơi nhận, được giả thiết là hằng số và n là số từ dữ liệu được trao đổi trong hệ thống.

Ví dụ: Giả sử cần thực hiện cộng n số trên hai máy tính, trong đó mỗi máy cộng $n/2$ số với nhau và tất cả các số đó được lưu ở máy tính thứ nhất. Kết quả của máy tính thứ hai khi được tính xong sẽ được chuyển về máy tính thứ nhất để nó cộng hai kết quả bộ phận với nhau. Bài toán này được phát biểu như sau:

1. Máy tính thứ nhất gửi $n/2$ số cho máy tính thứ hai
2. Cả hai máy tính cộng $n/2$ số một cách đồng thời
3. Máy tính thứ hai chuyển kết quả tính được về máy tính thứ nhất
4. Máy tính thứ nhất cộng hai kết quả để có kết quả cuối cùng.

Thời gian tính toán (ở bước 2 và 4):

$$t_{\text{comp}} = n/2 + 1$$

Thời gian truyền thông (ở bước 1 và 3):

$$\begin{aligned}
 t_{\text{comm}} &= (t_{\text{startup}} + n/2 * t_{\text{data}}) + (t_{\text{startup}} + t_{\text{data}}) \\
 &= 2*t_{\text{startup}} + (n/2 + 1) * t_{\text{data}}
 \end{aligned}$$

Độ phức tạp tính toán là $O(n)$ và độ phức tạp truyền thông cũng là $O(n)$, do vậy độ phức tạp nói chung của thuật toán trên cũng là $O(n)$.

1.1.3 Thuật toán song song

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

Để thiết kế được các thuật toán song song cần phải trả lời các câu hỏi sau:

- Việc phân chia dữ liệu cho các tác vụ như thế nào?
- Dữ liệu được truy cập như thế nào, những dữ liệu nào cần phải chia sẻ?
- Phân các tác vụ cho các tiến trình (bộ xử lý) như thế nào?
- Các tiến trình được đồng bộ ra sao?

Các nguyên lý chính trong thiết kế thuật toán song song:

1. Các nguyên lý lập lịch: Sử dụng các thuật toán lập lịch cho các bộ xử lý để giảm tối thiểu thời gian tính toán.

2. Nguyên lý hình ống: Nguyên lý này được áp dụng khi bài toán xuất hiện một dãy các thao tác $\{T_1, T_2, \dots, T_n\}$, trong đó T_{i+1} thực hiện sau khi T_i kết thúc.

3. Nguyên lý chia để trị: Chia bài toán thành những phần nhỏ hơn tương đối độc lập với nhau và giải quyết chúng một cách song song.

1.1.4 Các cách tiếp cận để thiết kế thuật toán song song

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

1. Thực hiện song song hoá những thuật toán tuần tự, biến đổi những cấu trúc tuần tự để tận dụng được những khả năng song song tự nhiên của tất cả các thành phần trong hệ thống xử lý.

2. Thiết kế những thuật toán song song mới phù hợp với kiến trúc song song.

3. Xây dựng những thuật toán song song từ những thuật toán song song đã được xây dựng cho phù hợp với cấu hình tập ô và môi trường song song thực tế.

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

1. Kiến trúc nào phù hợp cho bài toán?

2. Những bài toán loại nào sẽ xử lý hiệu quả trong kiến trúc song song cho trước?

1.1.5 Phân tích và đánh giá thuật toán song song

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

Định nghĩa: Một thuật toán có độ phức tạp tính toán $f(x) = O(g(x)) \Leftrightarrow$ Tồn tại số C dương và số nguyên x_0 sao cho $0 \leq f(x) \leq C * g(x)$, với mọi số lượng dữ liệu vào $x \geq x_0$.

Hiển nhiên $g(x)$, $f(x)$ là hai hàm có giá trị dương. $O(1)$ ký hiệu cho một hằng số bất kỳ.

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

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

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

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

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

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

thuật toán song song không bị giới hạn chỉ có nghĩa sử dụng khi chúng có thể chuyển đổi về thuật toán song song bị giới hạn.

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

Định nghĩa 1: Một thuật toán song song có độ phức tạp tính toán $O(T)$ với P bộ xử lý khi nó thực hiện nhiều nhất là $O(T * P)$ phép toán cơ sở (định lý Brent).

Định nghĩa 2: Một thuật toán song song có độ phức tạp tính toán $O(T)$ sử dụng rất nhiều bộ xử lý để thực hiện $O(e)$ phép toán cơ sở khi cài đặt với P bộ xử lý thì sẽ có độ phức tạp thời gian là $O(\lceil e/P \rceil + T)$.

Định nghĩa 3: Một thuật toán song song có độ phức tạp tính toán $O(T)$ với P bộ xử lý có thể cài đặt với $\lceil P/p \rceil$, $1 \leq p \leq P$ bộ xử lý thì sẽ có độ phức tạp thời gian là $O(p * T)$.

Định nghĩa 2 chỉ ra rằng khi số bộ xử lý được sử dụng giảm xuống trong một phạm vi nhất định thì thuật toán tiếp tục làm việc nhưng thời gian thực hiện sẽ tăng lên.

Định nghĩa 3 khẳng định rằng có cách để cài đặt thuật toán song song khi số các bộ xử lý được sử dụng bị giảm xuống.

Mức độ hiệu quả của thuật toán được thể hiện ở mức độ song song của thuật toán.

Mức độ song song của thuật toán là số lượng cực đại các phép toán độc lập có thể thực hiện đồng thời ở mỗi thời điểm thực hiện của thuật toán.

Ký hiệu $P(W)$ là độ song song của thuật toán, thì thuật toán hiệu quả giải để giải bài toán có cỡ W là những thuật toán chỉ cần sử dụng nhiều nhất $P(W)$ bộ xử lý.

Ngoài ra, để đánh giá được thuật toán song song chúng ta còn phải xét tới hệ số gia tốc của nó.

Hệ số gia tốc của thuật toán song song sử dụng p bộ xử lý được xác định như sau:

$$S_p = T_s / T_p$$

trong đó

+ T_s là thời gian thực hiện tính toán trên một bộ xử lý.

+ T_p là thời gian thực hiện tính toán trên p bộ xử lý.

Với giả thiết là bộ xử lý tuần tự và bộ xử lý song song là như nhau.

1.2 Các kiến thức cơ bản về giải số phương trình đạo hàm riêng

Trong mục này, luận văn trình bày một số kiến thức liên quan đến việc giải số phương trình đạo hàm riêng bao gồm cơ sở của phương pháp lưới, thuật toán thu gọn khối lượng tính toán của Samarskij-Nicolaev.

1.2.1 Phương pháp sai phân

Lưới sai phân:

$$\text{Xét bài toán} \quad \begin{cases} -\Delta u = f, x \in \Omega \\ u = g, x \in \partial\Omega \end{cases} \quad (1.1)$$

trong đó $\Omega = \{(x, y) \in R^2, a \leq x \leq b, c \leq y \leq d\}$, chọn 2 số nguyên $N > 1$ và $M > 1$, đặt $h = (b - a)/N$ gọi là bước lưới theo x , $k = (d - c)/M$ gọi là bước lưới theo y . Đặt $x_i = a + ih, y_j = c + jk, i = 0..N, j = 0..M$. Mỗi điểm (x_i, y_j) gọi là một nút lưới ký hiệu là (i, j) . Tập tất cả các nút trong ký hiệu là Ω_{hk} . Nút ở trên biên Γ gọi là nút biên. Tập tất cả các nút biên ký hiệu là Γ_{hk} , tập $\bar{\Omega}_{hk} = \Omega_{hk} \cup \Gamma_{hk}$ gọi là một lưới sai phân trên $\bar{\Omega}$.

Hàm lưới: Mỗi hàm xác định tại các nút của lưới gọi là một hàm lưới, giá trị của hàm lưới $u(x, y)$ tại nút lưới (i, j) viết tắt là $u_{i,j}$. Mỗi hàm $u(x, y)$ xác định tại mọi $(x, y) \in \bar{\Omega}$ tạo ra hàm lưới u xác định bởi $u_{i,j}$

Bài toán sai phân: Ký hiệu $Lu = f$ là phương trình mà nghiệm là tập các hàm số hai biến x, y có các đạo hàm riêng đến cấp m liên tục trong $\bar{\Omega} = \Omega \cup \Gamma$ giả sử bài toán có nghiệm $u \in C^4(\bar{\Omega})$, khi đó:

$$\max_{(x,y) \in \bar{\Omega}} \left| \frac{\partial^4 u}{\partial x^4}(x, y) \right| \leq C_1 = \text{const}, \max_{(x,y) \in \bar{\Omega}} \left| \frac{\partial^4 u}{\partial y^4}(x, y) \right| \leq C_2 = \text{const}.$$

Do đó theo công thức Taylor ta có:

$$u(x_{i+1}, y_j) = u(x_i) + h \frac{\partial u}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u}{\partial x^3} + o(h^4)$$

hay

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} = \frac{\partial^2 u}{\partial x^2} + o(h^2)$$

Một cách tương tự

$$u(x_i, y_{j+1}) = u(x_i, y_j + k) = u(x_i, y_j) + k \frac{\partial u}{\partial y} + \frac{k^2}{2!} \frac{\partial^2 u}{\partial y^2} + \frac{k^3}{3!} \frac{\partial^3 u}{\partial y^3} + o(k^4)$$

$$u(x_i, y_{j-1}) = u(x_i, y_j - k) = u(x_i, y_j) - k \frac{\partial u}{\partial y} + \frac{k^2}{2!} \frac{\partial^2 u}{\partial y^2} - \frac{k^3}{3!} \frac{\partial^3 u}{\partial y^3} + o(k^4)$$

Do đó:

$$\frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} = \frac{\partial^2 u}{\partial y^2} + o(k^2)$$

Vậy ta có:

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} = \Delta u + o(h^2 + k^2).$$

Ta đặt:

$$\Delta_{hk} u \equiv \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2}$$

Khi đó chúng ta có:

$$\Delta_{kh} u = \Delta u + (h^2 + k^2)$$

Số hạng $o(h^2 + k^2)$ là một số vô cùng bé bậc hai. Ta nói toán tử Δ_{kh} xấp xỉ toán tử, điều đó cho phép thay phương trình vi phân bằng phương trình sai phân:

$$\Delta_{hk} = f_{ij}, f_{ij} = f(x, y), (x, y) \in \Omega_{hk}$$

Tức là:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = f(x_i, y_j), (x_i, y_j) \in \Omega_{hk} \quad (1.2)$$

đồng thời thay điều kiện biên bằng điều kiện:

$$u_{ij} = g(x_i, y_j), (x_i, y_j) \in \Gamma_{hk} \quad (1.3)$$

Ta được bài toán sai phân hoàn chỉnh: tìm hàm lưới u_{ij} tại các nút (i, j) thỏa mãn hệ phương trình sai phân (1.2) và điều kiện biên (1.3). Như vậy việc tìm nghiệm xấp xỉ của bài toán vi phân (1.1) với độ chính xác cấp hai được đưa về giải bài toán sai phân (1.2) với điều kiện (1.3) bằng phương pháp đại số.

1.2.2 Thuật toán thu gọn khối lượng tính toán

Được đề xuất bởi Samarski-Nicolaev.

Bằng các phép biến đổi đơn giản về vec tơ và ma trận, các bài toán sai phân luôn luôn được đưa về hệ phương trình vec tơ 3 điểm thuộc một trong các dạng sau đây:

1. Bài toán biên thứ nhất

Xét bài toán biên thứ nhất đối với phương trình vec tơ ba điểm

$$-Y_{j-1} + CY_j - Y_{j+1} = F_j, 1 \leq j \leq N-1, Y_0 = F_0, Y_N = F_N \quad (1.4)$$

Trong đó Y_j là vec tơ cần tìm, C là ma trận vuông, F_j là vec tơ cho trước.

Khi đó (1.4) được viết dưới dạng

$$-Y_{j-1} + C^{(0)}Y_j - Y_{j+1} = F_j^{(0)}, 1 \leq j \leq N-1, Y_0 = F_0, Y_N = F_N \quad (1.5)$$

Bước khử thứ nhất:

Từ các phương trình đầu của (1.5) ta khử các Y_j với j lẻ. Muốn vậy ta viết 3 phương trình liên tiếp:

$$\begin{aligned} -Y_{j-2} + C^{(0)}Y_{j-1} - Y_j &= F_{j-1}^{(0)}. \\ -Y_{j-1} + C^{(0)}Y_j - Y_{j+1} &= F_j^{(0)}. \\ -Y_j + C^{(0)}Y_{j+1} - Y_{j+2} &= F_{j+1}^{(0)}. \end{aligned}$$

Nhân 2 vế của phương trình thứ hai với $C^{(0)}$ vào bên trái rồi cộng cả 3 phương trình lại ta được

$$-Y_{j-2} + C^{(1)}Y_j - Y_{j+2} = F_{j-1}^{(1)}, j = 2, 4, \dots, N-2, Y_0 = F_0, Y_N = F_N \quad (1.6)$$

Trong đó:

$$C^{(1)} = (C^{(0)})^2 - 2F, F_j^{(1)} = F_{j-1}^{(0)} + C^{(0)}F_j^{(0)} + F_{j+1}^{(0)}, j = 2, 4, \dots, N-2.$$

Nhận xét rằng hệ (1.6) chỉ chứa các Y_j với j chẵn, số vectơ ẩn Y_j là $\frac{N}{2} - 1$. Do đó nếu giải được hệ này thì các với j lẻ sẽ tìm được từ phương trình

$$C^{(0)}Y_j = F_j^{(0)} + Y_{j-1} + Y_{j+1}, j = 1, 3, \dots, N-1 \quad (1.7)$$

Như vậy hệ (1.5) tương đương với hệ (1.6) và (1.7)

Bước khử thứ hai:

Ở bước khử này ta sẽ tiến hành khử các Y_j hệ (1.6) với j là bội của 2 nhưng không là bội của 4. Muốn vậy ta viết 3 phương trình liên tiếp của (1.6)

$$\begin{aligned} -Y_{j-4} + C^{(1)}Y_{j-2} - Y_j &= F_{j-2}^{(1)} \\ -Y_{j-2} + C^{(1)}Y_j - Y_{j+2} &= F_j^{(1)}, (j = 4, 8, \dots, N-4) \\ -Y_j + C^{(1)}Y_{j+2} - Y_{j+4} &= F_{j+2}^{(1)} \end{aligned}$$

Nhân 2 vế của phương trình thứ hai với $C^{(1)}$ vào bên trái rồi cộng cả 3 vế phương trình lại ta được

$$-Y_{j-4} + C^{(2)}Y_j - Y_{j+4} = F_j^{(2)}, j = 4, 8, \dots, N-4, Y_0 = F_0, Y_N = F_N \quad (1.8)$$

Trong đó:

$$C^{(2)} = (C^{(1)})^2 - 2F, F_j^{(2)} = F_{j-2}^{(1)} + C^{(1)}F_j^{(1)} + F_{j+2}^{(1)}, j = 2, 4, \dots, N-2.$$

Hệ (1.8) chỉ chứa $\frac{N}{2} - 1$ vectơ ẩn Y_j , trong đó j là bội của 4. Nếu giải được hệ này thì các Y_j với j là bội của 4 sẽ tìm được từ phương trình 2 nhưng không là bội của 4 sẽ tìm được từ phương trình:

$$C^{(1)}Y_j = F_{j-2} + Y_{j+2} + F_j^{(1)}, j = 2, 6, 10, \dots, N-2$$

Cứ tiếp tục quá trình khử này. Kết quả là sau bước khử thứ l ta nhận được một hệ gồm $\frac{N}{C^{(1)}} - 1$ ẩn Y_j , trong đó j là bội của 2^l .

$$-Y_{j-2^l} + C^{(1)}Y_j - Y_{j+2^l} = F_j^{(1)}, j = 2^l, 2.2^l, 3.2^l, \dots, N - 2^l, Y_0 = F_0, Y_N = F_N \quad (1.9)$$

và nhóm các phương trình:

$$C^{(k-1)}Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, j = 2^{k-1}, 3.2^{k-1}, \dots, N - 2^{k-1}, k = l, l-1, \dots, 1 \quad (1.10)$$

trong đó các ma trận $C^{(k)}$ và các vec tơ vế phải $F_j^{(k)}$ được tính theo các công thức truy toán:

$$C^{(k)} = (C^{(k-1)})^2 - 2E, F_j^{(k)} = F_{j-2^{k-1}}^{(k-1)} + C^{(k-1)}F_j^{(k-1)} + F_{j+2^{k-1}}^{(k-1)}, \\ j = 2^k, 2.2^k, 3.2^k, \dots, N - 2^k, k = 1, 2, 3, \dots \quad (1.11)$$

Từ các bước khử trên suy ra rằng sau $n-1$ bước khử ($l = n-1$) ta thu được hệ gồm một phương trình đối với biến

$$Y_2^N - 1 = Y_{N/2} \\ C^{(n-1)}Y_j = F_j^{(n-1)} + Y_{j-2^{n-1}} + Y_{j+2^{n-1}} = F_j^{(n-1)} + Y_0 + Y_N, Y_0 = F_0, Y_N = F_N \quad (1.12)$$

Với vế phải đã biết. Vì vậy từ (1.12) ta có thể tìm được $Y_{N/2}$ và tất cả các ẩn còn lại được tìm liên tiếp từ các phương trình

$$C^{(k-1)}Y_j = F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}} \\ Y_0 = F_0 \\ Y_N = F_N \\ j = 2^{k-1}, 3.2^{k-1}, 5.2^{k-1}, \dots, N - 2^{k-1}, \\ k = n, n-1, \dots, 1 \quad (1.13)$$

Các công thức trên đã mô tả phương pháp rút gọn hoàn toàn giải. việc tính các $F_j^{(k)}$ theo công thức truy toán có thể dẫn đến việc tích lũy sai số nếu như chuẩn của ma trận $C^{(k-1)}$ lớn hơn 1. Ngoài ra các ma trận $C^{(k)}$ nói chung

là các ma trận đầy đủ, thậm chí cả ma trận ban đầu là $C^{(0)} = C$ là ma trận 3 đường chéo. Điều này dẫn đến tăng khối lượng tính toán khi tính các $F_j^{(k)}$ theo (1.13). Để khắc phục những khó khăn trên, thay cho $F_j^{(k)}$ ta sẽ tính các vec tơ $p_j^{(k)}$ và $q_j^{(k)}$ liên hệ với nhau theo công thức sau:

$$F_j^{(k)} = C^k p_j^{(k)} + q_j^{(k)}, j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \dots, N - 2^k, k = 0, 1, 2, \dots, n - 1 \quad (1.14)$$

trong đó ta chọn $p_j^{(0)}$ và $q_j^{(0)} = F_j, j = 1, 2, \dots, N - 1$. Bằng các công thức toán học, có thể thấy mối quan hệ mà $p_j^{(k)}$ và $q_j^{(k)}$ phải thỏa mãn như sau.

$$C^{(k)} p_j^{(k)} + q_j^{(k)} = C^{(k+1)} [q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)}] + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, \\ j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \dots, N - 2^k, k = 1, 2, \dots$$

Ta sẽ chọn $p_j^{(k)}$ và $q_j^{(k)}$ thỏa mãn $q_j^{(k)} = 2p_j^{(k)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}$ khi đó kết hợp với công thức $C^{(k)} + 2E = [C^{k+1}]^2$ ta có:

$$C^{(k)} p_j^{(k)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + C^{(k-1)} p_j^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}$$

Đặt

$$S_j^{(k-1)} = p_j^{(k)} + p_j^{(k-1)},$$

suy ra $S_j^{(k-1)}$ phải thỏa mãn

$$C^{(k-1)} S_j^{(k-1)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}$$

Như vậy ta thu được thuật toán sau đây để xác định các vec tơ $p_j^{(k)}$ và $q_j^{(k)}$

$$\begin{aligned} C^{(k-1)} S_j^{(k-1)} &= q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, S_j^{(k-1)} = p_j^{(k)} + p_j^{(k-1)}, \\ q_j^{(k)} &= 2q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)} \\ q_j^{(0)} &= F_j, p_j^{(0)} = 0, j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \dots, N - 2^k, k = 0, 1, 2, 3, \dots, n - 1. \end{aligned} \quad (1.15)$$

Ký hiệu $t_j^{(k-1)} = Y_j - p_j^{(k-1)}$, ta sẽ thấy rằng Y_j có thể được tính từ các công thức sau:

$$C^{(k-1)} t_j^{(k-1)} = q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, Y_j = p_j^{(k-1)} + t_j^{(k-1)},$$

$$Y_0 = F_0; Y_N = F_N, j = 2^k, 2.2^k, 3.2^k, \dots, N - 2^k, k = n, n-1, \dots, 1 \quad (1.16)$$

Nhận xét rằng các quá trình (1.15) và (1.16) luôn cần tính ma trận nghịch đảo $[C^{(k+1)}]^{-1}$. Bằng các phép biến đổi sơ cấp từ các mối quan hệ của ma trận $C^{(k)}$ và đa thức Chebysev $C^{(k)} = 2T_{2^k} \frac{1}{2} C$, ta có $C^{(k-1)} = \prod_{l=1}^{2^{k-1}} C_{l,k-1}$

trong đó

$$C_{l,k-1} = C - 2 \cos \frac{(2l-1)\pi}{2^k} E.$$

Như vậy chẳng hạn ta có phương trình

$$C^{k-1} v = \phi, \quad (1.17)$$

thì với việc giải lần lượt các phương trình

$$C_{l,k-1} v_l = v_{l-1}, l = 1, 2, \dots, 2^{k-1}, v_0 = \phi$$

sẽ cho ta nghiệm của (1.17) là $v = v_{2^{k-1}}$

Tóm lại qua các bước phân tích trên đây ta có thuật toán rút gọn hoàn toàn giải bài toán biên thứ nhất như sau

• Quá trình xuôi

Bước 1.1 Cho các giá trị ban đầu

$$p_j^{(0)}, q_j^{(0)} = F_j, j = 1, 2, 3, \dots, N-1$$

Bước 1.2 Với $k=1$ giải phương trình

$$Cp_j^{(1)} = q_j^{(0)} \text{ và tính}$$

$$q_j^{(1)} = 2p_j^{(1)} + q_{j-1}^{(0)} + q_{j+1}^{(0)}, j = 2, 4, 6, \dots, N-2$$

Bước 1.3 Với $k = 2, 3, \dots, n-1$ xác định các vec tơ

$$v_j^{(0)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, j = 2^k, 2.2^k, 3.2^k, \dots, N-2^k.$$

Sau đó với mỗi $l = 1, 2, \dots, 2^{k-1}$ và với mỗi $j = 2^k, 2.2^k, 3.2^k, \dots, N-2^k$, giải phương trình

$$C_{l,k-1} v_j^l = v_j^{(l-1)}$$

Khi đó

$$p_j^{(k)} = p_j^{(k-1)} + v_j^{(2^{k-1})}, q_j^k = 2p_j^k + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, j = 2^k, 2.2^k, 3.2^k, \dots, N-2^k$$

•Quá trình ngược

Bước 2.1 Cho các giá trị ban đầu $Y_0 = F_0, Y_N = F_N$

Bước 2.2 Với $k = n, n-1, \dots, 2$ tính

$$v_j^{(0)} = q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, j = 2^{k-1}, 3.2^{k-1}, \dots, N-2^{k-1}.$$

Sau đó, với mỗi $l = 1, 2, \dots, 2^{k-1}$ và với mỗi $j = 2^{k-1}, 3.2^{k-1}, \dots, N-2^{k-1}$, giải phương trình

$$C_{l,k-1} v_j^l = v_j^{(l-1)}$$

$$\text{Khi đó: } Y_j = p_j^{(k-1)} + v_j^{(2^{k-1})}, j = 2^k, 2.2^k, 3.2^k, \dots, N-2^k$$

Bước 2.3 Với $k=1$, giải phương trình

$$CY_j = q_j^{(0)} + Y_{j-1} + Y_{j+1}, j = 1, 3, 5, \dots, N-1$$

2. Bài toán biên thứ hai

Xét bài toán thứ hai

$$\begin{cases} Y_0 = F_0, j = 0 \\ -Y_{j-1} + CY_j + Y_{j+1} = F_j, 1 \leq j \leq N-1 \\ -2Y_{N-1} + CY_N = F_N \end{cases} \quad (1.18)$$

trong đó $N = 2^n, n > 0$ để giải bài toán (1.18) ta cũng thực hiện các bước khử lần lượt như đã được trình bày ở bài toán biên thứ nhất. Sau n phép khử ta nhận được phương trình

$$\begin{aligned} Y_0 &= F_0^{(n)}, \\ -2Y_0 + C^{(n)}Y_N &= F_N^{(n)} \end{aligned} \quad (1.19)$$

Và nhóm các phương trình

$$\begin{aligned} C^{(k-1)}Y_j &= F_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, \\ j &= 2^{k-1}, 3 \cdot 2^{k-1}, \dots, N - 2^{k-1}, k = n, n-1, \dots, 1 \end{aligned}$$

Trong đó $F_j^{(k)}$ và $C^{(k)}$ được xác định bởi công thức truy toán sau đây

$$\begin{aligned} F_0^{(k)} &= F_0 \\ F_j^{(k)} &= F_{j-1^{k-1}}^{(k-1)} + C^{(k-1)}F_{j-1} + F_{j+2^{k-1}}^{(k-1)} \\ j &= 2^k, 2 \cdot 2^k, \dots, N - 2^k \\ F_N^{(k)} &= 2F_{N-2^{k-1}}^{(k-1)} + C^{(k-1)}F_N^{(k-1)} \\ C &= [C^{(k-1)}]^2 - 2E \end{aligned}$$

Kí hiệu:

$$F_j^{(k)} = C^{(k)} p_j^{(k)} + q_j^{(k)}, j = 2^k, 2 \cdot 2^k, \dots, N - 2^k, k = 0, 1, 2, \dots, n$$

Bằng các phép biến đổi đơn giản và cách chọn $p_j^{(k)}$ và $q_j^{(k)}$ thích hợp, ta nhận được quá trình sau để xác định các vec tơ $p_j^{(k)}$ và $q_j^{(k)}$ với $j \neq N$

$$\begin{aligned} C^{(k-1)} S_j^{(k-1)} &= q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)} \\ p_j^{(k)} &= p_j^{(k-1)} + S_j^{(k-1)} \\ q_j^{(k)} &= 2p_j^{(k-1)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)} \\ j &= 2^k, 3 \cdot 2^k, \dots, N - 2^k, k = 0, 1, 2, \dots, n-1 \\ q_j^{(0)} &= F_j, p_j^{(0)} = 0 \end{aligned}$$

Tương tự, với $j=N$, ta có:

$$\begin{aligned} C^{(k-1)} S_N^{(k-1)} &= q_N^{(k-1)} + 2p_{N-2^{k-1}}^{(k-1)} \\ p_N^{(k)} &= p_N^{(k-1)} + S_N^{(k-1)} \\ q_N^{(k)} &= 2p_N^{(k-1)} + 2q_{N-2^{k-1}}^{(k-1)} \\ q_N^{(0)} &= F_N, p_N^{(0)} = 0 \end{aligned}$$

Trong đó

$$Y_j = p_j^{(k-1)} + t_j^{(k-1)}, j = 2^{k-1}, 3 \cdot 2^{k-1}, \dots, N - 2^{k-1}, k = n, n-1, \dots, 1$$

Trong đó $t_j^{(k-1)}$ là nghiệm của phương trình.

$$C^{(k-1)} t_j^{(k-1)} = q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}$$

Tóm lại, ta có thuật toán sau đây giải bài toán biên thứ hai.

• Quá trình xuôi

Bước 1.1 Xác định các giá trị ban đầu

$$p_j^{(0)} = 0, q_j^{(0)} = F_j, j = 1, 2, \dots, N$$

Bước 1.2 Với $k = 1, 2, \dots, n - 1$ xác định các vec tơ:

$$v_j^{(0)} = q_j^{(k-1)} + p_{j-2^{k-1}}^{(k-1)} + p_{j+2^{k-1}}^{(k-1)}, j = 2^k, 2.2^k, 3.2^k, \dots, N - 2^k.$$

Sau đó, với $l = 1, 2, \dots, 2^{k-1}$ và với mỗi $j = 2^k, 2.2^k, 3.2^k, \dots, N - 2^k$, giải phương trình

$$C_{l,k-1} v_j^l = v_j^{(l-1)}$$

Khi đó:

$$p_j^{(k)} = p_j^{(k-1)} + v_j^{(2^{k-1})}, q_j^{(k)} = 2p_j^{(k-1)} + q_{j-2^{k-1}}^{(k-1)} + q_{j+2^{k-1}}^{(k-1)}, j = 2^k, 2.2^k, 3.2^k, \dots, N - 2^k$$

Bước 1.3 Với $k = 1, 2, \dots, n - 1$ xác định các vec tơ:

$$v_N^{(0)} = q_N^{(k-1)} + 2p_{N-2^{k-1}}^{(k-1)}$$

Sau đó, với $l = 1, 2, \dots, 2^{k-1}$, giải phương trình

$$C_{l,k-1} v_N^l = v_N^{(l-1)}$$

Khi đó

$$p_N^{(k)} = p_N^{(k-1)} + v_N^{(2^{k-1})}, q_N^{(k)} = 2p_N^{(k-1)} + 2q_{N-2^{k-1}}^{(k-1)}$$

• Quá trình ngược

Bước 2.1 Xác định Y_N . Xác định vec tơ

$$v_N^{(0)} = q_N^{(n)} + 2Y_0$$

Sau đó, với $l = 1, 2, \dots, n$, giải phương trình

$$C_{l,n} v_N^l = v_N^{(l-1)}$$

Khi đó:

$$Y_N = p_N^{(n)} + v_N^{(n)}$$

Bước 2.2 Xác định $Y_j, j = 1, 2, \dots, N - 1$

Với $k = n, n - 1, \dots, 1$, xác định các vec tơ:

$$v_j^{(0)} = q_j^{(k-1)} + Y_{j-2^{k-1}} + Y_{j+2^{k-1}}, j = 2^{k-1}, 3 \cdot 2^{k-1}, \dots, N - 2^{k-1}.$$

Sau đó, với $l = 1, 2, \dots, 2^{k-1}$ và với mỗi $j = 2^{k-1}, 2 \cdot 2^{k-1}, \dots, N - 2^{k-1}$, giải phương trình

$$C_{l,k-1} v_j^l = v_j^{(l-1)}$$

Khi đó:

$$Y_j = p_j^{(k-1)} + v_j^{(2^{k-1})}, j = 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \dots, N - 2^k$$

Trong các tài liệu của Samaski-Nicolaev đã chứng minh được độ phức tạp của các thuật toán là $O(M \times N \log N)$.

1.3 Giới thiệu thư viện TK2004

Trên cơ sở của thuật toán thu gọn khối lượng tính toán của Samarski-Nicolaev, sau đây luận văn sẽ giới thiệu các kết quả xây dựng thư viện TK2004 giải số các bài toán biên elliptic. Các kết quả đã được tham khảo trong tài liệu [1].

1.3.1 Bài toán biên Dirichlet

Cho Ω là hình chữ nhật $\Omega = \{x = (x_1, x_2) \in R^2 : 0 < x_1 < l_1, 0 < x_2 < l_2\}$

Xét bài toán

$$\begin{cases} -\Delta u + cu = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega \end{cases} \quad (1.20)$$

trong đó $f \in L^2(\Omega); \varphi \in L^2(\Gamma)$

Rời rạc hóa miền $\bar{\Omega}$ bằng lưới Ω_h

$$\Omega_h = \left\{ x = (i, j) = (x_{1i}, x_{2j}); x_{1i} = ih_1, x_{2j} = jh_2; \right. \\ \left. h_1 = \frac{l_1}{M}; h_2 = \frac{l_2}{N}, i = 0, 1, \dots, M; j = 0, 1, \dots, N \right\}$$

Bằng cách biến đổi đơn giản ta có thể đưa bài toán sai phân tương ứng về hệ phương trình vec tơ 3 điểm có dạng như sau:

$$\begin{cases} -Y_{j-1} + CY_j - Y_{j+1} = F_j, j = 1, 2, \dots, N-1 \\ Y_0 = F_0, Y_N = F_N \\ Y_j = (y_{1,j}; y_{2,j}; \dots; y_{M-1,j})^T \\ F_0 = (g_{1,0}; g_{2,0}; \dots; g_{M-1,0})^T \\ Y_N = (g_{1,N}; g_{2,N}; \dots; g_{M-1,N})^T \end{cases} \quad (1.21)$$

Ma trận C có dạng

$$C = \begin{bmatrix} 2(r+1) & -r & 0 & \cdots & 0 & 0 & 0 \\ -r & 2(r+1) & -r & \cdots & 0 & 0 & 0 \\ 0 & -r & 2(r+1) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2(r+1) & -r & 0 \\ 0 & 0 & 0 & \cdots & -r & 2(r+1) & -r \\ 0 & 0 & 0 & \cdots & 0 & -r & 2(r+1) \end{bmatrix}$$

$$F_j = \begin{bmatrix} h_2^2 \varphi_{1,j} + rg_{0,j} \\ h_2^2 \varphi_{2,j} \\ \vdots \\ h_2^2 \varphi_{M-2,j} \\ h_2^2 \varphi_{M-1,j} + rg_{M,j} \end{bmatrix}, \text{ Với } r = \frac{h_2^2}{h_1^2}$$

Trên cơ sở của thuật toán thu gọn khối lượng tính toán của Samarski-Nicolaev, sử dụng ngôn ngữ Matlab giải hệ phương trình (1.21).

Xây dựng hàm $u0000(\varphi, b_1, b_2, b_3, b_4, L_1, L_2, c, M, N, p_1, p_2, q_1, q_2)$ trả lại nghiệm bằng số của bài toán trong đó

- + $M \times N$ là lưới trên hình chữ nhật
- + φ là ma trận giá trị số của hàm về phải
- + b_1, b_2, b_3, b_4 là vec tơ giá trị của điều kiện biên trên các cạnh
- + L_1, L_2 là kích thước hình chữ nhật
- + p_1, p_2, q_1, q_2 là tọa độ của hình chữ nhật trong không gian lưới

1.3.2 Bài toán biên hỗn hợp

Cho miền $\Omega = \{x = (x_1, x_2) \in R^2 : 0 < x_1 < l_1, 0 < x_2 < l_2\}$

Xét bài toán biên hỗn hợp

$$\begin{cases} -\Delta u = \varphi(x); x \in \bar{\Omega}, \\ u = g(x); x \in \Gamma_1 \\ \frac{\partial u}{\partial x} = \mu(x); x \in \Gamma_2 \end{cases} \quad (1.22)$$

trong đó $\varphi \in L^2(\Omega); g \in L^2(\Gamma_1); \mu \in L^2(\Gamma_2)$

$$\begin{aligned} \Gamma_1 &= \{x = (x_1, 0); 0 \leq x_1 \leq l_1\} \cup \\ &\{x = (l_1, x_2); 0 \leq x_2 \leq l_2\} \cup \{x = (0, x_2); 0 \leq x_2 \leq l_2\} \\ \Gamma_2 &= \{x = (x_1, l_2); 0 \leq x_1 \leq l_1\} \end{aligned}$$

Tương tự, ta đưa bài toán sai phân về hệ phương trình vec tơ 3 chiều

$$\begin{cases} Y_0 = F_0, \\ -Y_{j-1} + CY_{j+1} - Y_{j+1} = F_j, 1 \leq j \leq N-1 \\ -2Y_{N-1} + CY_N = F_N, j = N \end{cases} \quad (1.23)$$

Trong đó

$$\begin{aligned} F_0 &= (y_{1,0}, y_{2,0}, \dots, y_{M-1,0})^T; Y_j = (y_{1,j}, y_{2,j}, \dots, y_{M-1,j})^T; j = 1, \dots, N \\ C &= \begin{bmatrix} 2(r+1) & -r & 0 & \dots & 0 & 0 & 0 \\ -r & 2(r+1) & -r & \dots & 0 & 0 & 0 \\ 0 & -r & 2(r+1) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2(r+1) & -r & 0 \\ 0 & 0 & 0 & \dots & -r & 2(r+1) & -r \\ 0 & 0 & 0 & \dots & 0 & -r & 2(r+1) \end{bmatrix} \end{aligned}$$

$$F_j = \begin{bmatrix} h_2^2 \varphi_{1,j} + rg_{0,j} \\ h_2^2 \varphi_{2,j} \\ \vdots \\ h_2^2 \varphi_{M-2,j} \\ h_2^2 \varphi_{M-1,j} + rg_{M,j} \end{bmatrix}$$

$$F_N = \begin{bmatrix} h_2^2 \varphi_{1,N} + 2h_2 \mu_{1,N} + rg_{0,N} \\ h_2^2 \varphi_{2,N} + 2h_2 \mu_{2,N} \\ \vdots \\ h_2^2 \varphi_{M-2,N} + 2h_2 \mu_{M-2,N} \\ h_2^2 \varphi_{M-1,N} + 2h_2 \mu_{M-1,N} + rg_{M,N} \end{bmatrix}$$

Trên cơ sở của thuật toán thu gọn khối lượng tính toán của Samarski-Nicolaev, sử dụng ngôn ngữ Matlab giải hệ phương trình (1.23).

Xây dựng hàm $u0001(\varphi, b_1, b_2, b_3, b_4, L_1, L_2, c, M, N, p_1, p_2, q_1, q_2)$ trả lại nghiệm bằng số của bài toán, trong đó

- + $M \times N$ là lưới trên hình chữ nhật
- + φ là ma trận giá trị số của hàm về phải
- + b_1, b_2, b_3, b_4 là vec tơ giá trị của điều kiện biên trên các cạnh
- + L_1, L_2 là kích thước hình chữ nhật
- + p_1, p_2, q_1, q_2 là tọa độ của hình chữ nhật trong không gian lưới

Hoàn toàn tương tự, trong các trường hợp khi điều kiện biên Neumann trên 2 cạnh, 3 cạnh hoặc 4 cạnh của hình chữ nhật, sử dụng phương pháp sai phân, chúng ta cũng đưa bài toán elliptic về các hệ phương trình vec tơ 3 điểm và sử dụng thuật toán thu gọn khối lượng tính toán xác định nghiệm bằng số trên từng điểm lưới. các hàm được thiết lập tương tự là

$u_{0010}(\dots), u_{0100}(\dots), u_{1000}(\dots), u_{1100}(\dots), \dots, u_{1111}(\dots)$, sẽ cho phép xác định nghiệm số với các bài toán tương ứng. Trong các kí hiệu hàm trên, kí hiệu 0 tương ứng với biên Dirichlet, kí hiệu 1 tương ứng với biên Neumann. Các hệ thống hàm trên đã được xây dựng thành thư viện TK2004 cho phép giải số tất cả các bài toán biên elliptic với điều kiện biên hỗn hợp.

Kết luận: Trong chương 1, luận văn đã đưa ra một số kiến thức cơ bản về vấn đề tính toán song song, cơ sở của việc đánh giá độ phức tạp của thuật toán song song và đặc biệt là đưa ra hệ thống thư viện TK2004 giải số bài toán biên elliptic trên miền chữ nhật. Các kết quả này là nền tảng cho việc nghiên cứu các nội dung trong chương 2 của luận văn.

Chương 2

PHƯƠNG PHÁP LẬP SONG SONG DỰA TRÊN TƯ TƯỞNG CHIA MIỀN

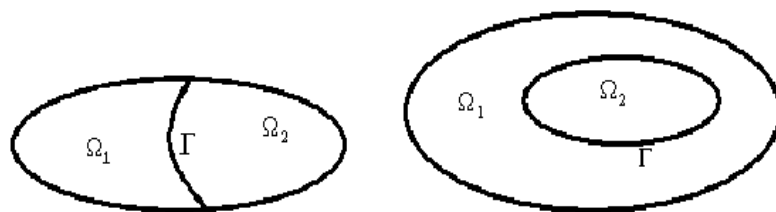
Trong chương này, luận văn sẽ trình bày các kiến thức cơ bản của phương pháp chia miền cùng các sơ đồ lập cơ bản của một số tác giả trên thế giới. Trên cơ sở các phương pháp cơ bản, luận văn sẽ đưa ra một số thuật toán song song giải bài toán biên elliptic, đánh giá độ phức tạp thuật toán và thử nghiệm tính toán số trên máy tính tuần tự. Các kết quả lý thuyết được tham khảo từ các tài liệu [2, 3, 5, 7, 8].

2.1 Cơ sở của phương pháp chia miền

Xét bài toán

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.1)$$

Trong đó Ω là miền d chiều với biên Lipschitz $\partial\Omega$, kí hiệu n là vec tơ pháp tuyến ngoài của miền Ω , f là hàm đã được cho thuộc không gian $L^2(\Omega)$. Giả sử miền Ω được chia thành hai miền con không giao nhau Ω_1 và Ω_2 , kí hiệu $\Gamma = \overline{\Omega_1} \cap \overline{\Omega_2}$ (hình 1)



Hình 1

Khi đó việc giải bài toán (2.1) trong miền Ω được đưa về việc giải bài toán trên hai miền con Ω_1 và Ω_2 . Kí hiệu u_1 là nghiệm của bài toán trong miền Ω_1 , u_2 là nghiệm của bài toán trong miền Ω_2 , khi đó u_1 và u_2 phải thỏa mãn các điều kiện đảm bảo tính liên tục của hàm và đạo hàm khi chuyển tiếp qua biên Γ như sau:

$$\begin{cases} u_1 = u_2, \forall x \in \Gamma, \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n}, \forall x \in \Gamma \end{cases}$$

Dễ dàng thấy rằng để giải được các bài toán trong Ω_1 và Ω_2 , chúng ta cần phải xác định được điều kiện biên của hai bài toán tương ứng trên đường phân chia Γ . Trong những năm qua, trên thế giới đã xuất hiện một số phương pháp chia miền như sau.

2.2 Một số thuật toán chia miền

Trên cơ sở của lý thuyết chia miền tổng quát, trên thế giới đã xuất hiện một số thuật toán chia miền của các tác giả áp dụng đối với trường hợp cụ thể.

2.2.1 Thuật toán chia miền Patrick le Talle.

Xét mô hình đơn giản

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = 0, x \in \partial\Omega, \end{cases} \quad (2.2)$$

Chia miền $\Omega = \Omega_1 \cup \Omega_2$ bởi biên chung S . Kí hiệu λ là giá trị hàm u trên S , khi đó ta có thể tiến hành giải song song hai bài toán trong miền $\Omega_i, i = 1, 2$.

$$\begin{cases} -\Delta u_i = f, x \in \Omega_i, \\ u_i = 0, x \in \partial\Omega \cap \partial\Omega_i, \\ u_i = \lambda, x \in S \end{cases} \quad (2.3)$$

Trong đó λ phải thỏa mãn điều kiện tổng đạo hàm pháp tuyến trên biên chung phải triệt tiêu tức là

$$\frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2} = 0, x \in S \quad (2.4)$$

Như vậy vấn đề quan trọng nhất là phải xác định giá trị λ .

Để tìm lời giải của bài toán, xuất phát từ giá trị λ , tiến hành giải song song hai bài toán Dirichlet

$$\begin{cases} -\Delta u_i = f, x \in \Omega_i, \\ u_i = 0, x \in \partial\Omega \cap \partial\Omega_i, \\ u_i = \lambda, x \in S \end{cases} \quad (2.5)$$

Tiếp theo giải song song hai bài toán Neumann

$$\begin{cases} -\Delta \Phi_i = 0, x \in \Omega_i, \\ \Phi_i = 0, x \in \partial\Omega \cap \partial\Omega_i, \\ \frac{\partial \Phi_i}{\partial n_i} = \frac{1}{2} \left(\frac{\partial u_1}{\partial n_1} + \frac{\partial u_2}{\partial n_2} \right), x \in S \end{cases} \quad (2.6)$$

Cập nhật lại $\lambda = \lambda - \rho(\Phi_1 + \Phi_2) / 2$, quá trình lặp lại cho đến khi hội tụ.

Thuật toán chia miền Patrick le Talle ở trên mới được trình bày dưới mức vi phân, trong các tài liệu đã biết đã trình bày phương pháp rời rạc hóa thuật toán vi phân trên cơ sở sử dụng sơ đồ lặp Seidel co giãn đưa các kết quả thực nghiệm trên máy tính điện tử với một số bài toán cụ thể. kết quả chứng

tổ thuật toán là đúng đắn và độ chính xác phụ thuộc vào việc sai phân các đạo hàm để nhằm xác định giá trị λ trên mỗi bước lặp.

Nhận xét:

+ Xuất phát từ thuật toán cơ sở, chúng ta có thể thấy: do sơ đồ lặp với mục tiêu là xác định gần đúng giá trị hàm λ trên biên chung đối với cả hai bài toán trong hai miền nên đối với bài toán biên hỗn hợp trong miền hình học phức tạp thì chúng ta phải thực hiện phép chia thành nhiều miền con mới tránh được bài toán biên hỗn hợp mạnh. Điều này sẽ tăng khối lượng tính toán trong các sơ đồ lặp.

+ Đối với thuật toán Patrick le Talle đã đưa ra tư tưởng chuyển bài toán trên biên Ω về việc giải song song hai bài toán trên hai miền Ω_1 và Ω_2 . Tuy nhiên mô hình chỉ dừng lại ở mức độ đơn giản và việc chứng minh sự hội tụ của thuật toán là khó khăn. Đồng thời chúng ta phải giải song song hai bài toán phụ là hai bài toán Neumann, điều này sẽ làm tăng khối lượng tính toán.

2.2.2 Thuật chia miền J.R.Rice, E.A. Vavalis, Daopi Yang.

Xét Ω là đa giác lồi thuộc $\mathbb{R}^d$, $d = 1, 2, \dots$. Với biên $\partial\Omega$, xét bài toán:

Cho $f \in L^2(\Omega)$; $g \in H^{1/2}(\partial\Omega)$, hãy tìm $u \in H^1(\partial\Omega)$

$$\begin{cases} Lu = f, x \in \Omega, \\ u = g, x \in \partial\Omega \end{cases} \quad (2.7)$$

trong đó

$$L_u = -\sum_{i,j=1}^d \frac{\partial}{\partial x_i} (a_{i,j}(x) \frac{\partial u}{\partial x_j}) + a_0(x)u. \quad (2.8)$$

Chia miền Ω thành hai miền con Ω_1 và Ω_2 như sau:

$$\begin{aligned}\overline{\Omega} &= \overline{\Omega_1} \cup \overline{\Omega_2}, \overline{\Omega_1} \cap \overline{\Omega_2}, \\ \partial\Omega_1 \cap \partial\Omega &\neq \phi, \partial\Omega_2 \cap \partial\Omega \neq \phi.\end{aligned}$$

kí hiệu $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ là biên phân chia giữa hai miền. Khi đó bài toán (2.7), (2.8) là tương đương với hai bài toán sau:

$$\begin{cases} Lu_1 = f, x \in \Omega_1, \\ u_1 = g, x \in \partial\Omega_1 \cap \partial\Omega, \\ u_1 = u_2, x \in \Gamma, \\ \frac{\partial u_1}{\partial \nu_1} + \frac{\partial u_2}{\partial \nu_2} = 0, x \in \Gamma. \end{cases} \quad (2.9)$$

$$\begin{cases} Lu_2 = f, x \in \Omega_2, \\ u_2 = g, x \in \partial\Omega_2 \cap \partial\Omega, \\ u_2 = u_1, x \in \Gamma, \\ \frac{\partial u_2}{\partial \nu_2} + \frac{\partial u_1}{\partial \nu_1} = 0, x \in \Gamma. \end{cases} \quad (2.10)$$

Trong đó u_1, u_2 lần lượt là nghiệm trong Ω_1, Ω_2 và ν_1, ν_2 là các vec tơ pháp tuyến ngoài của Γ ứng với Ω_1, Ω_2 .

Định nghĩa thuật toán chia miền như sau

$$\text{Chọn } u_n^{(0)} \in H^1(\Omega_n) \text{ với } u_n^{(0)}|_{\partial\Omega_1 \cap \partial\Omega} = g, n = 1, 2.$$

Xây dựng dãy lặp $u_n^{(k+1)} \in H^1(\Omega)$ với $u_n^{(k+1)}|_{\partial\Omega_n \cap \partial\Omega} = g, k = 1, 2, \dots$ Thỏa mãn

$$\begin{cases} Lu_1^{(2k+1)} = f, x \in \Omega_1, u_1^{(2k+1)} = \alpha u_1^{(2k)} + (1 - \alpha)u_2^{(2k)}, x \in \Gamma \\ Lu_2^{(2k+1)} = f, x \in \Omega_2, u_2^{(2k+1)} = \alpha u_2^{(2k)} + (1 - \alpha)u_1^{(2k)}, x \in \Gamma \\ Lu_1^{(2k+2)} = f, x \in \Omega_1, \frac{\partial u_1^{(2k+2)}}{\partial \nu_1} = \beta \frac{\partial u_1^{(2k+1)}}{\partial \nu_1} + (1 - \beta) \frac{\partial u_2^{(2k+1)}}{\partial \nu_2}, x \in \Gamma, \\ Lu_2^{(2k+2)} = f, x \in \Omega_2, \frac{\partial u_2^{(2k+2)}}{\partial \nu_2} = \beta \frac{\partial u_2^{(2k+1)}}{\partial \nu_2} + (1 - \beta) \frac{\partial u_1^{(2k+1)}}{\partial \nu_1}, x \in \Gamma. \end{cases} \quad (2.11)$$

Với $\alpha, \beta \in (0,1)$ là các tham số lặp được chọn để đẩy nhanh tốc độ hội tụ của sơ đồ lặp, việc chọn các tham số này phụ thuộc vào cách chia miền và bài toán gốc. Về mặt lý thuyết thì khó xác định các giá trị tối ưu, tuy nhiên qua phân tích lý thuyết và thực nghiệm đã chỉ ra rằng trong nhiều trường hợp nên chọn $\alpha = \beta = \frac{1}{2}$. Khi phép lặp hội tụ, giới hạn của dãy $\{u_n^{(k)}\}$ sẽ là nghiệm của bài toán ban đầu.

Nhận xét

+ Xuất phát từ sơ đồ lặp, chúng ta có thể thấy rằng do các bài toán ứng với chỉ số lẻ là nhằm hiệu chỉnh giá trị hàm, các bài toán ứng với chỉ số chẵn là hiệu chỉnh giá trị đạo hàm nên trong trường hợp bài toán biên hỗn hợp trên miền hình học phức tạp thì để tránh gặp phải những bài toán biên hỗn hợp mạnh thì phải sử dụng phép chia thành nhiều miền con. Điều này làm tăng khối lượng tính toán trong các bước lặp.

+ Do việc phân chia thành các bước lặp chẵn và lẻ độc lập nên điều này cũng dẫn tới việc thiết kế thuật toán song song giải bài toán trong trường hợp chia thành nhiều miền con, tuy nhiên chúng ta cũng gặp phải 2 cặp bài toán Dirichlet và Neumann do đó độ phức tạp thuật toán cũng tăng.

+ Việc chứng minh sự hội tụ của dãy lặp bằng lý thuyết là khó khăn. Tuy nhiên sự hội tụ có thể kiểm tra bằng các chương trình thực nghiệm.

2.2.3. Thuật toán chia miền Saito – Fujita

Với tư tưởng hiệu chỉnh giá trị hàm trên biên phân chia, năm 2001, hai nhà toán học Nhật Bản là Saito – Fujita dựa trên cơ sở sơ đồ lặp Dirichlet-Neumann đã đề xuất một phương pháp chia miền giải bài toán biên Elliptic với điều kiện biên Dirichlet.

Cho Ω là miền trong R^2 với biên Lipschitz.

Xét bài toán:

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.12)$$

trong đó $f \in L^2(\Omega); \varphi \in H^{1/2}(\partial\Omega)$. Chia miền $\Omega = \Omega_1 \cup \Omega_2$ bởi biên chung Γ .

Kí hiệu $\{u_1^{(k)}\}, \{u_2^{(k)}\}$ là các dãy hàm hội tụ đến u_1, u_2 một cách tương ứng.

Tư tưởng của phương pháp Saito – Fujita là tìm ra xấp xỉ $g = u|_{\Gamma}$ nhận được bởi sơ đồ lặp sau:

1. Cho trước $g^{(0)}$ xác định trên Γ .
2. Với $g^{(k)}$ xác định trên $\Gamma (k \geq 0)$, tiến hành giải lần lượt hai bài toán

$$\begin{cases} -\Delta u_1^{(k)} = f, x \in \Omega_1, \\ u_1^{(k)} = \varphi, x \in \Gamma_1, \\ u_1^{(k)} = g^{(k)}, x \in \Gamma \end{cases} \quad (2.13)$$

$$\begin{cases} -\Delta u_2^{(k)} = f, x \in \Omega_2, \\ u_2^{(k)} = \phi, x \in \Gamma_2, \\ \frac{\partial u_2^{(k)}}{\partial n^2} = -\frac{\partial u_1^{(k)}}{\partial n^1}, x \in \Gamma \end{cases} \quad (2.14)$$

3. Hiệu chỉnh lại giá trị của $g^{(k)}$ theo công thức

$$g^{(k+1)} = (1 - \theta)g^{(k)} + \theta u_2^{(k)}, x \in \Gamma \quad (2.15)$$

trong đó θ là tham số cần lựa chọn để dãy lặp hội tụ, $0 < \theta < 1$.

Ta thấy rằng, điều kiện liên tục của đạo hàm qua biên phân chia Γ đã thỏa mãn, còn điều kiện liên tục của hàm qua biên phân chia Γ phụ thuộc vào sự hội tụ của dãy lặp (2.15).

Sự hội tụ của phương pháp lặp đã được các tác giả chứng minh chặt chẽ bằng lý thuyết các sơ đồ lặp trên các không gian hàm. Tham số lặp tối ưu được xác định trên các mô hình đơn giản.

Nhận xét

+ Sơ đồ lặp trên chỉ cần giải một bài toán Dirichlet và một bài toán Neumann tại mỗi bước lặp nên khối lượng tính toán là đơn giản hơn các phương pháp trên.

+ Do việc giải lần lượt hai bài toán kế tiếp tại mỗi bước lặp nên cũng có thể xây dựng thuật toán song song trong trường hợp phép chia thành nhiều miền con.

2.3 Các sơ đồ lặp song song dựa trên chia miền

Trên cơ sở các phương pháp lặp cơ bản, sau đây luận văn sẽ đưa ra một số thuật toán song song giải bài toán biên elliptic cấp hai.

2.3.1 Phương pháp lặp chặn lẻ QD1.

Xét bài toán

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.16)$$

Giả sử miền Ω được chia thành $2n+1$ miền con bởi các biên $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_{2n}$.

Ω_1	Ω_2	Ω_3	$\dots$	Ω_{2k-1}	Ω_{2k}	Ω_{2k+1}	$\dots$	Ω_{2n+1}
	Γ_1	Γ_2	Γ_3	Γ_{2k-1}	Γ_{2k}	Γ_{2k+1}		Γ_{2n}

Hình 2

Trên cơ sở của phương pháp chia miền Saito – Fujita chúng ta xác định nghiệm của bài toán theo sơ đồ lặp sau đây.

Kí hiệu g_k là giá trị hàm trên đường biên Γ_k . Khi đó việc giải bài toán được thực hiện như sau:

Thuật toán:

Bước 1: Xuất phát từ $g_k^{(0)} = 0$, trên biên Γ_k

Bước 2: Giải các bài toán trên các miền lẻ $(\Omega_1, \Omega_3, \dots, \Omega_{2n+1})$

$$\begin{cases} -\Delta u_1^{(m)} = f, x \in \Omega_1, \\ u_1^{(m)} = \varphi, x \in \partial\Omega_1 \setminus \Gamma_1, \\ u_1^{(m)} = g_1^{(m)}, x \in \Gamma_1 \end{cases} \quad (2.17)$$

$$\begin{cases} -\Delta u_{2k-1}^{(m)} = f, \\ u_{2k-1}^{(m)} = \varphi, x \in \partial\Omega_{2k-1} \setminus (\Gamma_{2k-1} \cup \Gamma_{2k}), \\ u_{2k-1}^{(m)} = g_{2k-1}^{(m)}, x \in \Gamma_{2k-1} \\ u_{2k}^{(m)} = g_{2k}^{(m)}, x \in \Gamma_{2k} \end{cases} \quad (2.18)$$

$$\begin{cases} -\Delta u_{2n+1}^{(m)} = f, \\ u_{2n+1}^{(m)} = \varphi, x \in \partial\Omega_{2n+1} \setminus \Gamma_{2n}, \\ u_{2n+1}^{(m)} = g_{2n}^{(m)}, x \in \Gamma_{2n} \end{cases} \quad (2.19)$$

Bước 3: Giải các bài toán trên các miền chẵn $(\Omega_2, \Omega_4, \dots, \Omega_{2n})$

$$\begin{cases} -\Delta u_{2k}^{(m)} = f, \\ u_{2k}^{(m)} = \varphi, x \in \partial\Omega_{2k} \setminus (\Gamma_{2k} \cup \Gamma_{2k+1}), \\ \frac{\partial u_{2k}^{(m)}}{\partial \nu} = \frac{\partial u_{2k-1}^{(m)}}{\partial \nu}, x \in \Gamma_{2k} \\ \frac{\partial u_{2k}^{(m)}}{\partial \nu} = \frac{\partial u_{2k+1}^{(m)}}{\partial \nu}, x \in \Gamma_{2k+1} \end{cases} \quad (2.20)$$

$$\forall k = 1, 2, 3, \dots, n.$$

Bước 4: Hiệu chỉnh các giá trị $g_k^{(m)}$ theo công thức.

$$\begin{cases} g_{2k-1}^{(m+1)} = (1 - \theta)g_{2k-1}^{(m)} + \theta u_{2k-1}^{(m)}, x \in \Gamma_{2k-1} \\ g_{2k}^{(m+1)} = (1 - \theta)g_{2k}^{(m)} + \theta u_{2k}^{(m)}, x \in \Gamma_{2k} \\ \forall k = 1, 2, \dots, n \end{cases} \quad (2.21)$$

Nhận xét:

+ Sự hội tụ của phương pháp phụ thuộc hoàn toàn vào sự hội tụ của sơ đồ lặp (2.21). Các sơ đồ lặp này có thể chứng minh sự hội tụ tương tự như phương pháp chia miền Saito – Fujita.

+ Theo các bước của thuật toán chúng ta thấy rằng trong bước 2, việc giải $n+1$ bài toán trên các miền lẻ được thực hiện độc lập. Do đó ta có thể sử dụng $n+1$ bộ xử lý độc lập giải $n+1$ bài toán một cách song song. Tương tự trong bước 3, việc giải n bài toán trên các miền chẵn cũng được thực hiện độc lập. Do đó ta có thể sử dụng n bộ xử lý độc lập giải n bài toán một cách song song. Do đó, thuật toán giải trên $2n+1$ miền luôn được thực hiện song song bởi $(n+1)$ bộ xử lý.

+ Cũng theo thuật toán, chúng ta thấy để thực hiện được bước 3 ta cần có dữ liệu của bước 2, nên giữa bước 2 và bước 3 phải được thực hiện tuần tự nên thuật toán trên là thuật toán song song nhưng chưa được thực hiện triệt để.

+ Việc giải các bài toán trên các miền con là các miền đơn giản (hình chữ nhật) nên luôn luôn sử dụng các hàm trong thư viện TK2004.

Đánh giá độ phức tạp của thuật toán

Giả sử chúng ta cần thực hiện K bước lặp để nhận được nghiệm xấp xỉ của bài toán với độ chính xác cấp hai. Khi đó hiển nhiên tại mỗi bước lặp chúng ta cần phải giải song song $n+1$ bài toán Dirichlet sau đó giải song song n bài toán Neumann. Với mỗi bài toán trên theo thuật toán thu gọn khối lượng

tính toán có độ phức tạp $O(MN \log N)$ do đó độ phức tạp thời gian đối với thuật toán là

$$t_{comp1} = 2K \times O(MN \log N)$$

Tại mỗi bước lặp, Các bộ xử lý cần phải gửi N số liệu cho các bộ xử lý lẻ để hiệu chỉnh giá trị trên các biên chung do đó độ phức tạp truyền thông được tính theo công thức

$$t_{comm} = 2Nt_{staup} + Nt_{data}$$

Tại bước hiệu chỉnh, $N+1$ bộ xử lý lẻ phải tính toán các công thức hiệu chỉnh trên các vector N chiều. Do đó số lượng tính toán tại bước này là

$$t_{comp2} = 8KN$$

Như vậy độ phức tạp của toàn bộ thuật toán được xác định bằng công thức

$$O(T) = 2K \times O(MN \log N) + 2Nt_{staup} + Nt_{data} + 8KN \approx O(KMN \log N)$$

Trong đó $M \times N$ là lưới chia hình chữ nhật cơ sở.

Một số kết quả thực nghiệm đối với QD1

Để kiểm tra tính đúng đắn của sơ đồ lặp song song dạng chặn lẻ, chúng tôi sử dụng phương pháp lưới chuyển bài toán vi phân về bài toán dạng sai phân với lưới chia hình chữ nhật là 64×64 lưới điểm, sau đó tiến hành thực hiện sơ đồ lặp song song trên cơ sở các hàm trong thư viện TK2004.

Trong các ví dụ, kí hiệu $u^*(x, y)$ là nghiệm đúng của bài toán, $u^{(m)}(x, y)$ là nghiệm xấp xỉ sau bước lặp thứ m , điều kiện dừng lặp là

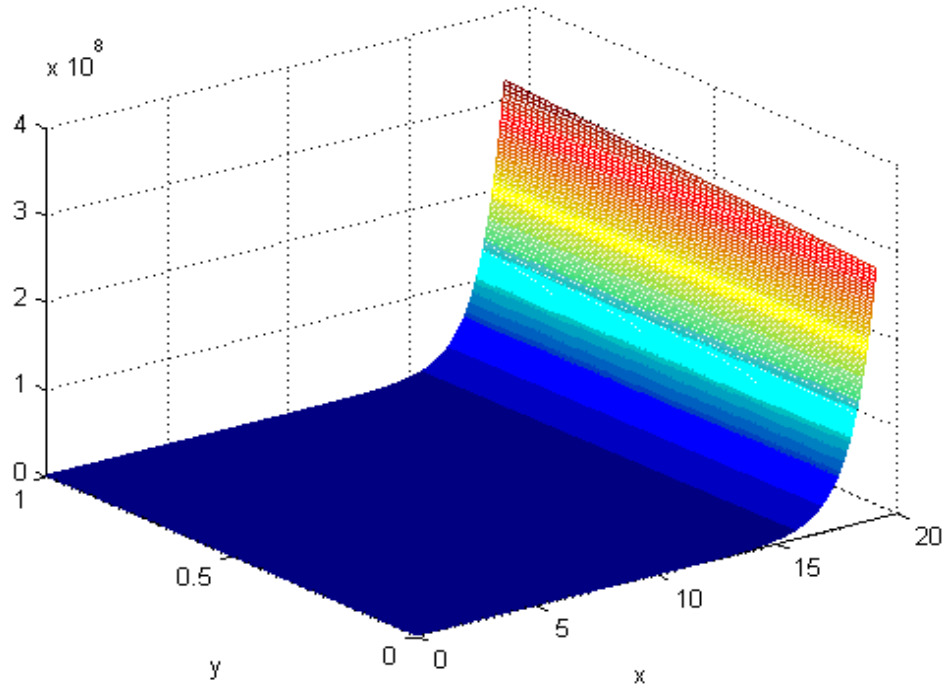
$$\max |u^{(m)}(x, y) - u^*(x, y)| < \varepsilon. \text{ Ngôn ngữ sử dụng lập trình là Matlab trên}$$

máy tính PC. Kích thước các hình chữ nhật cơ sở luôn lấy là $a = 1; b = 1$.

Các kết quả thực nghiệm thu được như sau (Số miền chia là 19)

Bảng 1: $u^*(x, y) = e^x \log(y + 5) - \sin y \log(x + 6)$

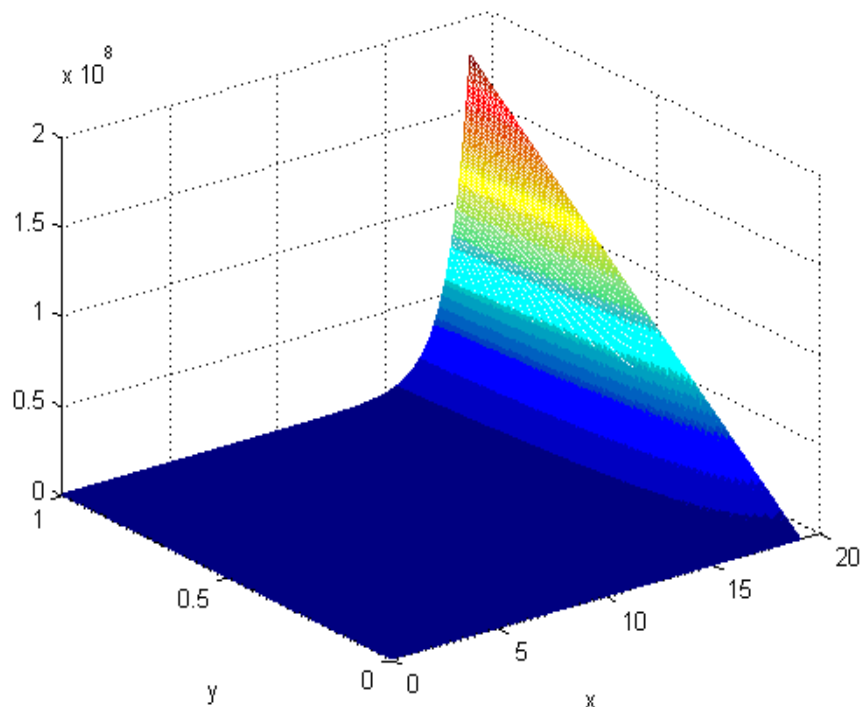
Tham số	Số bước lặp	Sai số
0.2	40	1.10^{-4}
0.3	27	6.10^{-5}
0.4	21	3.10^{-5}
0.5	17	5.10^{-5}
0.6	11	7.10^{-5}
0.7	13	9.10^{-5}
0.8	30	9.10^{-5}



Hình 3: Đồ thị nghiệm xấp xỉ

Bảng 2: $u^*(x, y) = x^3 + ye^x + y^3 + xe^{-y}$

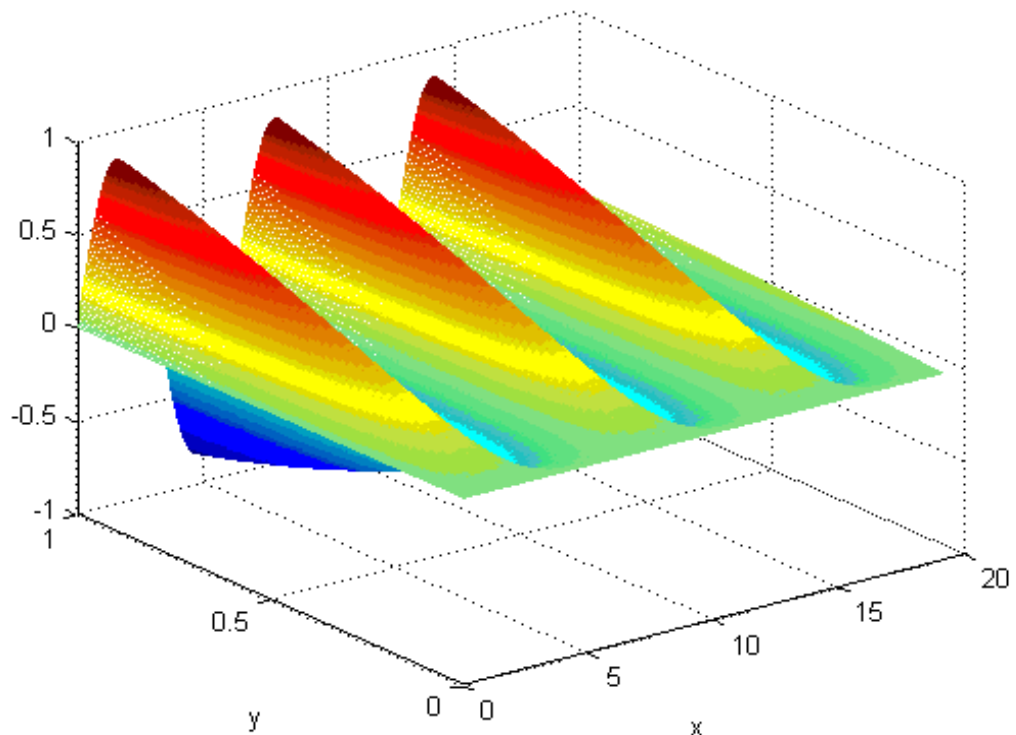
Tham số	Số bước lặp	Sai số
0.2	40	8.10^{-4}
0.3	36	2.10^{-4}
0.4	21	8.10^{-5}
0.5	15	5.10^{-5}
0.6	10	7.10^{-5}
0.7	17	7.10^{-5}
0.8	32	8.10^{-5}



Hình 4: Đồ thị nghiệm xấp xỉ

Bảng 3: $u^*(x, y) = \sin x \sin y$

Tham số	Số bước lặp	Sai số
0.2	40	2.10^{-4}
0.3	25	2.10^{-4}
0.4	20	2.10^{-4}
0.5	17	2.10^{-4}
0.6	10	2.10^{-4}
0.7	15	2.10^{-4}
0.8	35	2.10^{-4}



Hình 5: Đồ thị nghiệm xấp xỉ

Nhận xét:

Qua thực nghiệm có thể thấy sơ đồ lặp song song dạng chuẩn lẻ QD1 là hội tụ với tham số lặp θ được chọn trong khoảng $(0.1 - 0.9)$. Tham số tối ưu được chọn trong khoảng $(0.5-0.7)$.

2.3.2 Phương pháp song song QD2

Xét bài toán

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.22)$$

Giả sử miền Ω được chia thành n miền con bởi các biên $\Gamma_1, \Gamma_2, \dots, \Gamma_{n-1}$.

Ω_1	Ω_2	$\dots$	Ω_k	$\dots$	Ω_n
	Γ_1	Γ_2	Γ_{k-1}	Γ_k	Γ_k

Hình 6

Trên cơ sở của phương pháp chia miền Patrick le Talle, chúng ta xác định nghiệm của bài toán theo sơ đồ lặp sau đây.

Kí hiệu g_k là giá trị hàm trên đường biên Γ_k , $k = 1, 2, 3, \dots, n-1$.

Thuật toán:

Bước 1: Xuất phát từ $g_k^{(0)} = 0$, trên biên Γ_k

Bước 2: Giải song song các bài toán trên các miền

$$\begin{cases} -\Delta u_1^{(m)} = f, x \in \Omega_1, \\ u_1^{(m)} = \varphi, x \in \partial\Omega_1 \setminus \Gamma_1, \\ u_1^{(m)} = g_1^{(m)}, x \in \Gamma_1 \end{cases} \quad (2.23)$$

$$\begin{cases} -\Delta u_k^{(m)} = f, x \in \Omega_k \\ u_k^{(m)} = \varphi, x \in \partial\Omega_k \setminus \{\Gamma_{k-1}, \Gamma_k\}, \\ u_k^{(m)} = g_{k-1}^{(m)}, x \in \Gamma_{k-1} \\ u_k^{(m)} = g_k^{(m)}, x \in \Gamma_k \end{cases} \quad (2.24)$$

$$\begin{cases} -\Delta u_n^{(m)} = f, x \in \Omega_n \\ u_n^{(m)} = \varphi, x \in \partial\Omega_n \setminus \Gamma_{n-1}, \\ u_n^{(m)} = g_{n-1}^{(m)}, x \in \Gamma_{n-1} \end{cases} \quad (2.25)$$

Bước 3: Hiệu chỉnh trên Γ_k

$$\begin{cases} g_k^{(m+1)} = (1 - \theta)g_k^{(m)} + \theta\left(\frac{\partial u_k^{(m)}}{\partial \nu} - \frac{\partial u_{k+1}^{(m)}}{\partial \nu}\right) \\ \forall k = 1, 2, \dots, n \end{cases} \quad (2.26)$$

Nhận xét:

+ Sự hội tụ của phương pháp phụ thuộc hoàn toàn vào sự hội tụ của sơ đồ lặp (2.26). Việc chứng minh sơ đồ lặp này hội tụ bằng lý thuyết là khó khăn, tuy nhiên chúng ta có thể khẳng định thông qua các kết quả thực nghiệm.

+ Theo các bước của thuật toán chúng ta thấy rằng trong bước 2, việc giải n bài toán trên các miền con được thực hiện độc lập. Do đó ta có thể sử dụng n bộ xử lý độc lập giải n bài toán một cách song song. Do đó đây chính là quá trình xử lý song song được thực hiện bằng n bộ xử lý độc lập.

+ Cũng theo thuật toán, chúng ta thấy để thực hiện được bước 3 ta cần có dữ liệu của bước 2, nên giữa bước 2 và bước 3 phải được thực hiện tuần tự nên thuật toán trên là thuật toán song song nhưng chưa được thực hiện triệt để.

+ Việc giải các bài toán trên các miền con là các miền đơn giản (hình chữ nhật) nên luôn luôn sử dụng các hàm trong thư viện TK2004.

Đánh giá độ phức tạp của thuật toán

Giả sử chúng ta cần thực hiện K bước lặp để nhận được nghiệm xấp xỉ của bài toán với độ chính xác cấp hai. Khi đó hiển nhiên tại mỗi bước lặp chúng ta cần phải giải song song n bài toán Dirichlet. Với mỗi bài toán trên theo thuật toán thu gọn khối lượng tính toán có độ phức tạp $O(MN \log N)$ do đó độ phức tạp thời gian đối với thuật toán là

$$t_{comp1} = K \times O(MN \log N)$$

Tại mỗi bước lặp, Các bộ xử lý cần phải gửi N số liệu cho các bộ xử lý tổng để hiệu chỉnh giá trị trên các biên chung do đó độ phức tạp truyền thông được tính theo công thức

$$t_{comm} = 2Nt_{staup} + Nt_{data}$$

Tại bước hiệu chỉnh, bộ xử lý tổng phải tính toán các công thức hiệu chỉnh trên các vector N chiều. Do đó số lượng tính toán tại bước này là

$$t_{comp2} = 8KN$$

Như vậy độ phức tạp của toàn bộ thuật toán được xác định bằng công thức:

$$O(T) = K \times O(MN \log N) + 2Nt_{staup} + Nt_{data} + 8KN \approx O(KMN \log N)$$

Trong đó $M \times N$ là lưới chia hình chữ nhật cơ sở.

Một số kết quả thực nghiệm QD2

Để kiểm tra tính đúng đắn của sơ đồ lặp song song QD2, chúng tôi sử dụng phương pháp lưới chuyển bài toán vi phân về bài toán dạng sai phân với lưới chia hình chữ nhật là 64×64 lưới điểm, sau đó tiến hành thực hiện sơ đồ lặp song song trên cơ sở các hàm trong thư viện TK2004.

Trong các ví dụ, kí hiệu $u^*(x, y)$ là nghiệm đúng của bài toán, $u^{(m)}(x, y)$ là nghiệm xấp xỉ sau bước lặp thứ m , điều kiện dừng lặp là $\max |u^{(m)}(x, y) - u^*(x, y)| < \varepsilon$. Kích thước các hình chữ nhật cơ sở luôn lấy là $a = 1; b = 1$. Các kết quả thực nghiệm thu được như sau (Số miền chia là 19):

Bảng 4: $u^*(x, y) = e^x \log(y + 5) - \sin y \log(x + 6)$

Tham số	Số bước lặp	Sai số
0.2	80	1.10^{-4}
0.3	67	6.10^{-5}
0.4	40	7.10^{-5}
0.5	25	5.10^{-5}
0.6	18	8.10^{-5}
0.7	17	8.10^{-5}
0.8	77	2.10^{-4}

Bảng 5: $u^*(x, y) = x^3 + ye^x + y^3 + xe^{-y}$

Tham số	Số bước lặp	Sai số
0.2	80	0.003
0.3	66	2.10^{-4}
0.4	39	7.10^{-5}
0.5	25	5.10^{-5}
0.6	20	6.10^{-5}
0.7	25	8.10^{-5}
0.8	73	2.10^{-4}

Nhận xét:

+ Qua thực nghiệm có thể thấy sơ đồ lặp song song QD2 là hội tụ với tham số lặp θ được chọn trong khoảng $(0.1 - 0.9)$. Tham số tối ưu trong khoảng $(0.5-0.7)$.

+ So với phương pháp lặp QD1 thì số lần lặp đối với QD2 để đạt đến cùng một mức sai số là nhiều hơn. Điều này được giải thích vì trong phương pháp QD2, việc xác định nghiệm của các bài toán lẻ không tận dụng được giá trị nghiệm của các bài toán chẵn do hai bài toán giải song song với nhau. Tuy nhiên ưu điểm của phương pháp là sơ đồ lặp là tính toán theo hệ thống song song hoàn toàn.

2.4 Mô hình bài toán biên gián đoạn

Xét mô hình bài toán

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.27)$$

Trong đó $\Omega \in R^2$, giả sử hàm và đạo hàm của nghiệm phương trình bị gián đoạn qua đường Γ tức là:

$$\begin{cases} u_1 - u_2 = \Psi_1(x), x \in \Gamma, \\ \frac{\partial u_1}{\partial n} - \frac{\partial u_2}{\partial n} = \Psi_2(x), x \in \Gamma. \end{cases} \quad (2.28)$$

Mô hình này thường được xuất hiện trong các bài toán cơ học và vật lý khi môi trường được chế tạo bởi 2 loại vật liệu khác nhau. Nhiệt lượng và thông lượng nhiệt sẽ bị gián đoạn khi qua một mặt phân cách nào đó trong môi trường. Đây cũng là một mô hình hiện nay đang được các nhà khoa học quan tâm.

2.4.1 Phương pháp lặp tuần tự

Xuất phát từ phương pháp chia miền Saito-Fujita việc giải bài toán trên được giải bằng các phương pháp sau đây.

Kí hiệu g là giá trị của hàm u_1 trên biên Γ . Khi đó việc giải bài toán được thực hiện bằng sơ đồ lặp sau đây.

Bước 1: Xuất phát từ $g^{(0)}$, trên biên Γ

Bước 2: $\forall m = 1, 2, \dots$, thực hiện giải các bài toán

$$\begin{cases} -\Delta u_1^{(m)} = f, x \in \Omega_1, \\ u_1^{(m)} = \varphi, x \in \Gamma_1, \\ u_1^{(m)} = g^m, x \in \Gamma \end{cases} \quad (2.29)$$

$$\begin{cases} -\Delta u_2^{(m)} = f, x \in \Omega_2 \\ u_2^{(m)} = \varphi, x \in \Gamma_2, \\ \frac{\partial u_2^{(m)}}{\partial n} = \frac{\partial u_2^{(m)}}{\partial n} - \Psi_2(x), x \in \Gamma \end{cases} \quad (2.30)$$

Bước 3: Hiệu chỉnh trên Γ

$$\begin{cases} g^{(m+1)} = (1 - \theta)g^{(m)} + \theta(u_2^{(m)} + \Psi_1(x)) \\ \forall x \in \Gamma \end{cases} \quad (2.31)$$

Việc chứng minh sự hội tụ của sơ đồ lặp đưa ra là chưa thực hiện được. Tuy nhiên chúng ta có thể kiểm tra được bằng thực nghiệm.

Trong các kết quả thực nghiệm sau đây, chúng tôi cho trước các bước nhảy $\Psi_1(x), \Psi_2(x)$, điều kiện biên $\varphi(x)$ và hàm vế phải $f(x, y)$.

Điều kiện dừng lặp là $\max |u^{(m+1)}(x, y) - u^{(m)}(x, y)| < \varepsilon$, miền

$$\Omega = [0, 2] \times [0, 1], \Gamma = \{x = 1, 0 < y < 1\}.$$

Ví dụ 1.

$$\Psi_1(x, y) = x^4 + y^2$$

$$\Psi_2(x, y) = 4 \sin x - \log(x + y + 2)$$

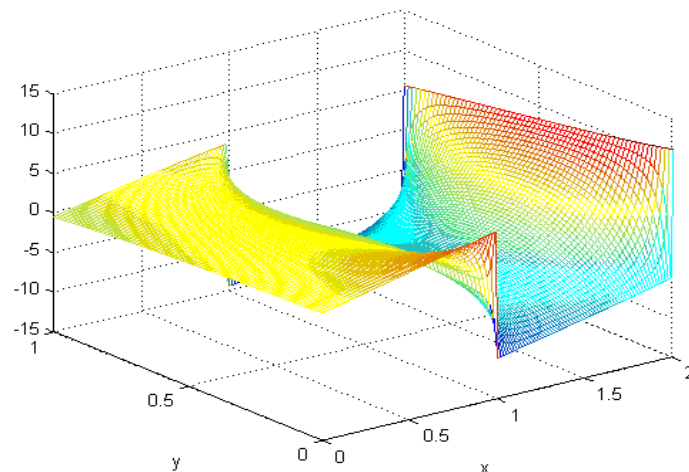
$$\varphi(x, y) = 4 \sin x - y^3 + e^{x-y}$$

$$f = 4 \sin(xy) + e^{x+y}$$

Các kết quả thực nghiệm thu được như sau:

Bảng 6: Kết quả thực nghiệm của ví dụ 1

Tham số	Số bước lặp	Sai số
0.1	40	0.0035
0.2	40	7.10^{-5}
0.3	40	5.10^{-7}
0.4	40	2.10^{-9}
0.5	36	5.10^{-11}
0.6	28	5.10^{-11}
0.7	30	5.10^{-11}
0.8	40	2.10^{-8}
0.9	40	0.0015



Hình 7: Đồ thị nghiệm xấp xỉ trong ví dụ 1

Ví dụ 2.

$$\Psi_1(x, y) = 5$$

$$\Psi_2(x, y) = 9$$

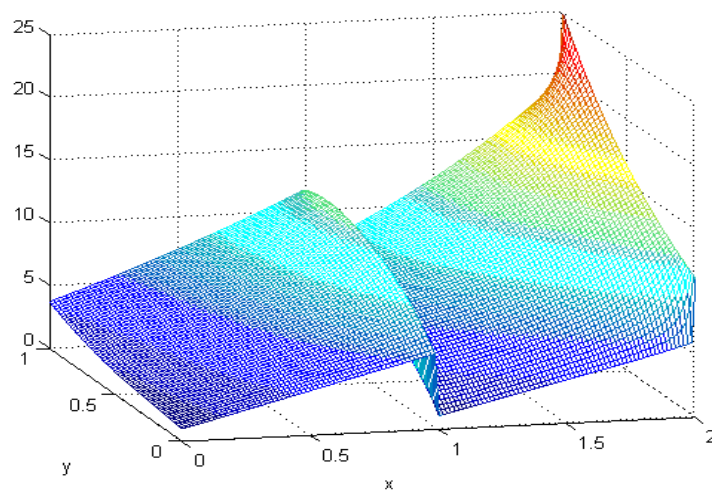
$$\varphi(x, y) = 4 \sin x + y^3 + e^{x+y}$$

$$f = 4 \sin(xy) + e^{x-y}$$

Các kết quả thực nghiệm thu được như sau

Bảng 7: Kết quả thực nghiệm của ví dụ 2

Tham số	Số bước lặp	Sai số
0.1	40	0.0031
0.2	40	6.10^{-5}
0.3	40	6.10^{-7}
0.4	40	2.10^{-9}
0.5	35	9.10^{-11}
0.6	28	4.10^{-11}
0.7	31	5.10^{-11}
0.8	40	7.10^{-8}
0.9	40	0.0045



Hình 8: Đồ thị nghiệm xấp xỉ trong ví dụ 2

Qua thực nghiệm có thể khẳng định phương pháp lặp đối với bài toán biên gián đoạn qua mặt phân cách là hội tụ. Xuất phát từ phương pháp trên, chúng ta có thể đề xuất sơ đồ lặp song song như sau:

2.4.2 Phương pháp lặp chặn lẻ QD3

Xét bài toán

$$\begin{cases} -\Delta u = f, x \in \Omega, \\ u = \varphi, x \in \partial\Omega, \end{cases} \quad (2.32)$$

Giả sử miền Ω được chia thành $2n+1$ miền con $\Omega_1, \dots, \Omega_{2n+1}$ bởi các biên $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_{2n}$ và đồng thời giá trị hàm và đạo hàm gián đoạn khi chuyển tiếp qua các đường biên phân chia tức là

$$\begin{aligned} [u]_{\Gamma_k} &= \Psi_{1k}(x), x \in \Gamma_k \\ \left[\frac{\partial u}{\partial n} \right]_{\Gamma_k} &= \Psi_{2k}(x), x \in \Gamma_k \end{aligned}$$

Xuất từ sơ đồ lặp trên, chúng ta xây dựng sơ đồ lặp sơ đồ lặp dạng chặn lẻ giải bài toán trên như sau.

Kí hiệu $g_k(x)$ là giá trị hàm u_k trên đường biên Γ_k , với $k = 1, 2, \dots, 2n$

Thuật toán:

Bước 1: Xuất phát từ $g_k^{(0)} = 0$, trên Γ_k

Bước 2: $\forall m = 1, 2, \dots$, tiến hành giải các bài toán miền lẻ

$\Omega_1, \Omega_3, \dots, \Omega_{2n+1}$,

$$\begin{cases} -\Delta u_1^{(m)} = f, x \in \Omega_1, \\ u_1^{(m)} = \varphi, x \in \partial\Omega_1 \setminus \Gamma_1, \\ u_1^{(m)} = g_1^{(m)}, x \in \Gamma_1 \end{cases} \quad (2.33)$$

$$\begin{cases} -\Delta u_{2k-1}^{(m)} = f, x \in \Omega_k, \\ u_k^{(m)} = \varphi, x \in \partial\Omega_k \setminus \{\Gamma_{k-1} \cup \Gamma_k\}, \\ u_k^{(m)} = g_{k-1}^{(m)}, x \in \Gamma_{k-1}, \\ u_k^{(m)} = g_k^{(m)}, x \in \Gamma_k. \\ k = 1, 3, \dots, 2n-1 \end{cases} \quad (2.34)$$

$$\begin{cases} -\Delta u_{2n+1}^{(m)} = f, x \in \Omega_{2n+1} \\ u_{2n+1}^{(m)} = \varphi, x \in \partial\Omega_{2n+1} \setminus \Gamma_{2n}, \\ u_{2n+1}^{(m)} = g_{2n}^{(m)}, x \in \Gamma_{2n} \end{cases} \quad (2.35)$$

Bước 3: Với mọi $m = 0, 1, 2, \dots$ tiến hành giải các bài toán trên miền chặn $\Omega_2, \Omega_4, \dots, \Omega_{2n}$

$$\begin{cases} -\Delta u_k^{(m)} = f, x \in \Omega_k \\ u_k^{(m)} = \varphi, x \in \partial\Omega_k \setminus (\Gamma_{k-1} \cup \Gamma_k), \\ \frac{\partial u_k^{(m)}}{\partial \nu} = \frac{\partial u_{k-1}^{(m)}}{\partial \nu} - \Psi_{2,k-1}(x), x \in \Gamma_{k-1} \\ \frac{\partial u_k^{(m)}}{\partial \nu} = \frac{\partial u_{k+1}^{(m)}}{\partial \nu} - \Psi_{2,k+1}(x), x \in \Gamma_k \\ k = 2, 4, \dots, 2n \end{cases} \quad (2.36)$$

Bước 4: Hiệu chỉnh

$$\begin{cases} g_k^{(m+1)} = (1 - \theta)g_k^{(m)} - \theta(u_{k+1}^{(m)}(x) + \Psi_{1k}(x)), \\ g_{k+1}^{(m+1)} = (1 - \theta)g_{k+1}^{(m)} - \theta(u_k^{(m)}(x) + \Psi_{1k+1}(x)) \end{cases} \quad (2.27)$$

Nhận xét:

+ Sự hội tụ của phương pháp phụ thuộc hoàn toàn vào sự hội tụ của sơ đồ lặp (2.26). Việc chứng minh sơ đồ lặp này hội tụ bằng lý thuyết là khó khăn, tuy nhiên chúng ta có thể khẳng định thông qua các kết quả thực nghiệm.

+ Theo các bước của thuật toán chúng ta thấy rằng trong bước 2, việc giải n bài toán trên các miền con được thực hiện độc lập. Do đó ta có thể sử dụng n bộ xử lý độc lập giải n bài toán một cách song song. Do đó đây chính là quá trình xử lý song song được thực hiện bằng n bộ xử lý độc lập.

+ Cũng theo thuật toán, chúng ta thấy để thực hiện được bước 3 ta cần có dữ liệu của bước 2, nên giữa bước 2 và bước 3 phải được thực hiện tuần tự nên thuật toán trên là thuật toán song song nhưng chưa được thực hiện triệt để.

+ Việc giải các bài toán trên các miền con là các miền đơn giản (hình chữ nhật) nên luôn luôn sử dụng các hàm trong thư viện TK2004.

Đánh giá độ phức tạp của thuật toán

Giả sử chúng ta cần thực hiện K bước lặp để nhận được nghiệm xấp xỉ của bài toán với độ chính xác cấp hai. Khi đó hiển nhiên tại mỗi bước lặp chúng ta cần phải giải song song n bài toán Dirichlet. Với mỗi bài toán trên theo thuật toán thu gọn khối lượng tính toán có độ phức tạp $O(MN \log N)$ do đó độ phức tạp thời gian đối với thuật toán là

$$t_{comp1} = K \times O(MN \log N)$$

Tại mỗi bước lặp, các bộ xử lý cần phải gửi N số liệu cho các bộ xử lý tổng để hiệu chỉnh giá trị trên các biên chung do đó độ phức tạp truyền thông được tính theo công thức

$$t_{comm} = 2Nt_{stap} + Nt_{data}$$

Tại bước hiệu chỉnh, bộ xử lý tổng phải tính toán các công thức hiệu chỉnh trên các vectơ N chiều. Do đó số lượng tính toán tại bước này là

$$t_{comp2} = 8KN$$

Như vậy độ phức tạp của toàn bộ thuật toán được xác định bằng công thức

$$O(T) = K \times O(MN \log N) + 2Nt_{stap} + Nt_{data} + 8KN \approx O(KMN \log N)$$

Trong đó $M \times N$ là lưới chia hình chữ nhật cơ sở.

Thuật toán song song QD3 cũng có thể kiểm tra sự hội tụ thông qua các chương trình thực nghiệm tương tự như thuật toán QD2. Các kết quả thực nghiệm khẳng định tham số lựa chọn tối ưu cũng trong khoảng (0.5-0.7).

Kết luận: Trong chương 2, luận văn đã đưa ra một số sơ đồ tính toán song song giải bài toán biên elliptic dựa trên tư tưởng chia miền. Các sơ đồ này đã được kiểm tra bằng các chương trình thực nghiệm thông qua nhiều ví dụ khẳng định sự hội tụ của các sơ đồ.

PHẦN KẾT LUẬN

Nội dung chính của luận văn đã đề cập đến các sơ đồ tính toán song song giải bài toán biên elliptic dựa trên tư tưởng chia miền của các tác giả trên thế giới. Các kết quả chính của luận văn đạt được bao gồm:

1. Luận văn đã tìm hiểu các kiến thức cơ bản về lý thuyết tính toán song song, phương pháp sai phân và thư viện TK2004 giải số bài toán biên elliptic với hệ số hằng trên miền chữ nhật.

2. Tìm hiểu về cơ sở phương pháp chia miền và đặc biệt là phương pháp chia miền của các tác giả Saito-Fujita dựa trên tư tưởng hiệu chỉnh hàm trên biên phân chia.

3. Trên cơ sở của phương pháp Saito-Fujita, Luận văn đưa ra 2 sơ đồ lặp song song QD1 và QD2 giải bài toán biên elliptic với số miền chia là tùy ý, đối với mỗi phương pháp đã đánh giá độ phức tạp của quá trình xử lý song song, tính toán thử nghiệm trên nhiều ví dụ khác nhau. Các kết quả thực nghiệm đã khẳng định sự hội tụ của các sơ đồ tính toán song song QD1 và QD2.

4. Luận văn nghiên cứu mô hình bài toán biên gián đoạn qua mặt phân cách. Trên cơ sở của phương pháp Saito-Fujita, luận văn đưa ra sơ đồ lặp song song dạng chặn lẻ QD3 giải bài toán biên elliptic với số miền chia là tùy ý, đánh giá độ phức tạp của quá trình xử lý song song, thử nghiệm trên máy tính điện tử. Các kết quả thực nghiệm đã khẳng định sự hội tụ của sơ đồ tính toán song song QD3.

Hướng nghiên cứu tiếp theo của luận văn là nghiên cứu đề thuật các thuật toán song song khác dựa trên các tư tưởng chia miền khác nhau áp dụng vào các bài toán phức tạp hơn, đồng thời tiến hành thực nghiệm áp dụng mô hình tính toán song song đối với các bài toán trong thực tế.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Vũ Vinh Quang (2005), Các kết quả về việc ứng dụng thuật toán thu gọn khối lượng tính toán giải các bài toán elliptic với các điều kiện biên hỗn hợp, Hội thảo Khoa học Toàn quốc “Phát triển công cụ Tin học trợ giúp cho giảng dạy, nghiên cứu và ứng dụng Toán học”, Nhà xuất bản Khoa học Kỹ thuật, 247-256.

[2] Vũ Vinh Quang, Trương Hà Hải, Cao Thị Anh Thư (2009), Mô hình tính toán song song giải bài toán biên hỗn hợp mạnh dựa trên chia miền, Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên, T.1(30):25-35.

Tiếng Anh

[3] Dang Quang A, Vu Vinh Quang, “A domain decomposition method for solving an elliptic boundary value problem”, Methods of Complex and Clifford Analysis (Proceedings of 2004 International Conference on Applied Mathematics), SAS International Publication, Delhi, 309-319.

[4] Marchuk G.I. (1982), Methods of Numerical Mathematics, Springer, New York.

[5] Quarteroni A. and Valli A. (1999), Domain Decomposition Methods for Partial Differential Equations, Clarendon Press. Oxford.

[6] Samarskij A. and Nikolaev E. (1989), Numerical Methods for Grid Equations, Vol. 2, Birkhauser, Basel.

[7] Saito N. and Fujita H. (2001), Operator Theoretical Analysis to Domain Decomposition Methods, 12th Int. Conf. on Domain Decomposition Methods, 63-70, [www.ddm.org/DDI 2/saito.pdf](http://www.ddm.org/DDI2/saito.pdf).

[8] Le Tallec P. Et al., Domain decomposition method for large linearly elliptic three dimensional problem, Rapprts de Recherche, No 1182, INRIA, France, 1990.

[9] Barry Wilkingson, Michael Allen (1999), Parallel Programming, Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall New Jersey.

[10] Joseph JaJa (1992), An Introduction to Parallel Algorithms, Addison – Wesley.

[11] M. Sasikumar, Dinesh Shikhare, P. Ravi Prakash (2000), Introduction to Parallel Processing, Prentice – Hall.
