



ИНСТИТУТ ПО ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ
БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ



Димитър Георгиев Славчев

АВТОРЕФЕРАТ

на дисертация

СЪСТАВНИ ЧИСЛЕНИ МЕТОДИ И СКАЛИРУЕМИ БЛОЧНИ АЛГОРИТМИ

за придобиване на
образователна и научна степен „Доктор“
по научна специалност: 01.01.09 „Изчислителна математика“
в професионално направление: 4.5 „Математика“

Научен ръководител:
чл.-кор. Светозар Маргенов

19 януари 2022 г.

Дисертацията е обсъдена и допусната до защита на разширено заседание на секция „Научни пресмятания с Лаборатория по 3D дигитализация и микроструктурен анализ“ на ИИКТ-БАН, състояло се на 11.01.2022 г.

Дисертационният труд е структуриран в увод, изложение от четири глави, заключение, декларация за оригиналност на резултатите, списък на публикациите по дисертационния труд, апробация на резултатите, приложение и библиография. Дисертационният труд е в обем от 140 страници, 47 фигури и 7 таблици, 90 цитирани литературни източника и 2 приложения.

Защитата на дисертацията ще се състои на 17.05.2022 г. от 14:00 часа в зала 218 на блок 25А на ИИКТ-БАН на открито заседание на научно жури в състав:

1. проф. дмн Светослав Маринов Марков – ИМИ-БАН
2. доц. дн Миглена Николаева Колева – РУ „Ангел Кънчев“
3. доц. д-р Кирил Стоянов Щерев – Имех-БАН
4. проф. д-р Пенчо Генов Маринов – ИИКТ-БАН
5. проф. д-р Иван Димов Лирков – ИИКТ-БАН

Резервни членове:

1. доц. д-р Петър Пенчев Рашков – ИМИ-БАН
2. доц. д-р Станислав Николаев Харизанов – ИИКТ-БАН

Материалите по защитата са на разположение на интересуващите се в стая 216 на ИИКТ-БАН, ул. „Акад. Г. Бончев“, бл. 25А.

Автор: Димитър Славчев

Заглавие: Съставни числени методи и скалируеми блочни алгоритми

Обща характеристика на дисертационния труд

Актуалност на темата

Численото решаване на задачи с голяма размерност изисква използването на високопроизводителни изчислителни компютърни системи, както и специализиран хардуер и софтуер – графични карти, ускорители, високоскоростна комуникация между сървърите на системата, софтуерни стандарти и пакети за комуникация между процесорни ядра и сървъри, софтуерни пакети имплементиращи ефективни числени методи и много други.

Съществуват различни методи за дискретизация на диференциални уравнения. Такива например са мрежовите методи като методът на крайните елементи, методът на граничните елементи и методът на крайните разлики. След тяхното прилагане, диференциалните уравнения се свеждат до системи от линейни алгебрични уравнения. Методът на Гаус е универсален подход за решаване на такива системи. В общия случай той има висока изчислителна сложност – $O(n^3)$, където n е броят на неизвестните [29].

При дискретизация на диференциалното уравнение с метода на граничните елементи, както и при прилагане на метода на крайните елементи за нелокални задачи (например изследваната в тази дисертация *аномална* (дробна) дифузия) [2], получената матрица на коравина е *плътна*. Един възможен подход за намаляване на изчислителната сложност на решението на системи с такива матрици е йерархичната компресия въведена от Хакбуш (Hackbusch) [10]. При нея се използва *структурата* на изходната матрица. Целта е както за да се редуцира заеманата памет, така и да се подобри изчислителната ефективност. Тук под *структура* на плътна матрица се разбира наличието на апроксимация на изходящата матрица с нисък ранг в извъндиагоналните блокове. Това свойство позволява представянето на извъндиагоналните блокове като произведение на по-малки матрици. Съществуват различни разновидности на йерархични матрици, в т.ч. $\mathcal{H}$, $\mathcal{H}^2$ или йерархични полусепарабелни (Hierarchically Semi-Separable) HSS матрици.

Обзор на основни резултати в областта

Огромният напредък във възможностите на съвременните високопроизводителни изчислителни системи засилва още повече ролята на ефективните числени методи и паралелните алгоритми. Суперкомпютърните симулации са определящи за развитието в редица високотехнологични области. Такива например са *in silico* молекулярната биология и проектиране на лекарствени средства, анализа на турбулентни течения, безразрушителният контрол, обработката на тримерни изображения, динамика на флуидите и много други.

След подходяща дискретизация математическите модели обикновено се свеждат до задачи на линейната алгебра, измежду които определяща е ролята на решаването на системи от линейни алгебрични уравнения. За целта се разработват специализирани софтуерни средства.

В общия случай за решаване на системи линейни алгебрични уравнения с плътни матрици се прилагат варианти на метода на Гаус, които използват последователно

изключване на неизвестните. По предположение плътната матрица е хомогенна, тъй като не се предполага, че тя има нулеви елементи. Методът на Гаус има изчислителна сложност $O(n^3)$. В настоящата дисертация е изследван алтернативен подход на базата на йерархична компресия. Целта е намаляване на изчислителната сложност. Тук под *структура* на плътна матрица се разбира наличието на извъндиагонални блокове с нисък ранг. По-точно предполага се че такова свойство е в сила за матрица, която апроксимира изходната. Наличието на подходяща *структура* е в основата на йерархичната компресия и съответните йерархични методи за решаване на системи с плътни матрици за класове от задачи на изчислителната математика. Йерархичната компресия е въведена от Хакбуш в [10], където са изследвани т.н $\mathcal{H}$ -матрици. Други разновидности на йерархична компресия са $\mathcal{H}^2$ -матриците [11] и йерархичните полусепарабелни матрици (HSS) [16]. Теоретична обосновка на методите използващи HSS компресия може да се намери в [28].

Съществена част от дисертацията е посветена на численото решаване на дробно дифузионни задачи. Дробната дифузия (наричана още аномална дифузия) описва нелокални процеси, които се наблюдават в различни физични и социални среди. За разлика от обикновената (локална) дифузия аномалната дифузия включва т.н. бързи преходи или тунелни ефекти. В литературата са публикувани разнообразни примери на математически модели на процеси и явления, които се описват с дробна дифузия. Такива например са: течения в силно нееднородни порести среди, суперпроводимост, дифузия на полимери в суперстудени среди [4]; електродифузия на йони в нервни клетки [15] и диагностика с помощта на фотонна дифузия [25]; обработка на изображения и машинно самообучение [20]; разпространение на вирусни заболявания, компютърни вируси, както и на престъпност [6]. *Дробният* оператор на Лаплас описва аномална дифузия по пространството. Съществуват различни дефиниции на *дробен* лапласиан. Важно е да отбележим, че те не са еквивалентни. Така например в [13] е анализирана разликата между интегралната и спектралната дефиниции (виж също статии [17] и [12] и литературата в тях).

Цели и задачи на дисертацията

Основни цели на дисертацията са:

- Сравнителен анализ на бързодействието и паралелната производителност на често използвани софтуерни пакети прилагащи директна гаусова елиминация за решаване на системи линейни алгебрични уравнения с плътни матрица при използване на централни процесори (CPU) и ускорители (MIC).
- Анализ на бързодействието, паралелното ускорение и точността на приблизителен метод за решаване на системи линейни алгебрични уравнения базиран на йерархична полусепарабелна компресия (HSS) от софтуерния пакет STRUMPACK за системи с подходяща структура на матрицата.
- Разработване на алгоритми за пренареждане на неизвестните при задачи породени от дискретизация с метод на крайните елементи на дробна дифузия с цел подобряване на ефективността на йерархичната полусепарабелна компресия на така изчислената матрица на коравина.

- Числено решаване на елиптични и параболични задачи от областта на аномалната дифузия описана с интегралната формулировка на дробен лапласиан и дискретизирана по пространството с метод на крайните елементи.

Методология на изследването

В дисертацията се анализира ефективността, в смисъл на бързодействие, паралелното ускорение и точността (за приблизителните решения) на блочни методи за решаване на плътни системи линейни алгебрични уравнения. За целта се използват софтуерни пакети, в които са приложени изследваните блочни методи.

При задачата, разгледана в Глава 2 се използва паралелната програма, разработена в [23], за дискретизиране на зададена задача и генериране на системата линейни алгебрични уравнения. При задачите за дробна дифузия, разгледана в Глави 3 и 4, се използва MatLab програмата разработена в [2] от Акоста и др. за генериране на системата линейни алгебрични уравнения. Разработени са програми на MatLab, които изчисляват използваните пренареждания за тази задача (Приложение А) и матрица на масата (Приложение Б). Софтуерните библиотеки със свободен достъп са компилирани върху използваните компютърни системи.

Структура на съдържанието

В Увода е дадена мотивацията за настоящата работа. Накратко са описани използваните методи и решаваните задачи.

Глава 1 има въвеждащ характер и описва използваните блочни методи за решаване на плътни системи линейни уравнения, както и оценка за изчислителните им сложности. В Раздел 1.1 е описан накратко универсалният директен метод гаусова елиминация и базираната на него LU факторизация. В Раздел 1.4 са разгледани йерархични методи за решаване на системи линейни уравнения разработени за решаване на системи със *структурирани* матрици (плътни и разреждени). Описани са и предимствата на разглеждания метод базиран на HSS компресия – по-ниска оценка на изчислителната сложност за задачи с подходяща *структура* на матрицата.

В Глава 2 са представени числени резултати за обтичането на крилни профили на Жуковски. Получената система с *плътна* матрица се използва за бенчмарк при сравнителният анализ на използваните блочни алгоритми.

В Глава 3 е разгледана задача за двумерна *аномална* дифузия моделирана с *дробния* оператор на Лаплас. За дискретизация по пространството се прилага метод на крайните елементи.

В Глава 4 е разгледана параболична задача за двумерна по пространството *аномална* дифузия.

В Заключение са представени обобщаващи бележки за основните резултати получени в дисертацията. Формулирани са научните и научно-приложните приноси. Даден е списък на публикуваните статии и на изнесените доклади на научни форуми върху които се базира тази работа.

Глава 1 Методи за решаване на системи линейни уравнения с плътни матрици

Много задачи от изчислителната практика се решават числено чрез свеждане до система от линейни алгебрични уравнения. Така например при прилагане на *метода на граничните елементи* или при дискретизация с крайни елементи на уравнения с *дробна* степен на оператора на Лаплас (*дробна* дифузия), се получава система с плътна матрица.

1.1 Преки методи

Методът на Гаус е универсален метод за решаване на системи от линейни алгебрични уравнения. Той е основа на повечето преки методи. Така например LU факторизацията (още наречена декомпозиция) се базира на последователно изключване на неизвестните по метода на Гаус. LU факторизацията е базов метод реализиран във високопроизводителните софтуерни библиотеки на изчислителната линейна алгебра.

1.2 Метод на Гаус

Методът на Гаус за решаване на системата от линейни алгебрични уравнения $Ax = b$ включва прав и обратен ход: (i) Чрез еквивалентни преобразувания матрицата на системата се привежда в горна триъгълна (прав ход, елиминация); (ii) Рекурсивно в обратна последователност се изключват извъндиагоналните елементи в i -тия ред на матрицата за $i = n - 1, n - 2, \dots, 1$ (обратен ход, заместване).

Нека означим с $\tilde{A}$ разширената матрица $\tilde{A} = (A|b)$. Тогава на първата стъпка на правия ход се умножава първия ред на $\tilde{A}$ по $-\frac{a_{i1}}{a_{11}}$ и се прибавя към реда с номер i , за всяко $i = 2, \dots, n$. Правият ход завършва след изпълняване на $n - 1$ такива стъпки. В резултат получаваме горната триъгълна матрица $\tilde{A}^{(n-1)}$.

На обратния ход горната триъгълна матрица $\tilde{A}^{(n-1)}$ се преобразува в диагонална матрица, като за целта се изпълняват $n - 1$ стъпки. На първата стъпка последния ред на разширената матрица се умножава по $\frac{a_{nn}^{(i-1)}}{a_{i-1,n}^{(n-1)}}$ и се събира с i -тия ред за $i = 1, \dots, n - 1$. На втората стъпка се взема $(n - 1)$ -вия ред, като след нейното изпълнение се анулират извъндиагоналните елементи в $(n - 1)$ -вия стълб. Така, след $n - 1$ стъпки на обратния ход, матрицата от коефициенти се свежда до диагонална матрица. Решението на системата се получава след разделяне на i -тия ред със съответния диагонален елемент.

Така в трансформирания вектор $b^{(n-1, n-1)} = x$ получаваме решението на системата. Изчислителната сложност на метода на Гаус се определя от правия ход [29]:

$$\mathcal{N}_{\text{Гаус}} \sim \frac{2n^3}{3} = O(n^3).$$

1.3 LU факторизация

LU факторизацията е изразяване на матрицата A като произведение на две триъгълни матрици $A = LU$. Тук L е долна триъгълна матрица с единици по главния диагонал, а

U е горна триъгълна матрица. Тази факторизация се изчислява с помощта на модифициран метод на Гаус и се използва във високопроизводителните библиотеки (LAPACK, MKL, ACML, PLASMA, ATLAS, и други) за решаване на системи линейни уравнения.

Правият ход на метода на Гаус може да се запише във вида

$$\underbrace{L_{n-1}L_{n-2}\dots L_2L_1}_{\tilde{L}}A = U,$$

където $L_1, L_2, \dots, L_{n-1}$ са долни триъгълни матрици с единици по главния диагонал. Непосредствено се проверява, че $\tilde{L}$ и $L = \tilde{L}^{-1}$ са също долни триъгълни матрици с единици по главния диагонал. Така получаваме:

$$\tilde{L}A = U \iff A = LU, \quad L = \tilde{L}^{-1}.$$

След факторизацията на A , системата от линейни алгебрични уравнения се свежда до решаване на две системи с триъгълни матрици. Полагаме

$$L \underbrace{Ux}_y = b,$$

след което се:

1. Решава системата $Ly = b$ с право заместване;
2. Решава системата $Ux = y$ с обратно заместване.

Изчислителната сложност на факторизацията е $O\left(\frac{2}{3}n^3\right)$, докато правото и обратното заместване са със сложност $O(n^2)$.

1.4 Йерархични матрици. Методи за решаване на системи линейни уравнения с помощта на йерархична полусепарабелна компресия

Йерархичните матрици се използват за апроксимация на разреждени по данни (data-sparse) матрици. Под разреждени по данни се разбират матрици, които имат *структура*, позволяваща апроксимация с помощта на компресирани матрици, които се записват с помощта на по-малък брой елементи. В общия случай, разредените по данни матрици не удовлетворяват условието да имат $O(n)$ ненулеви елемента. Хакбуш въвежда понятието „йерархични матрици“ в [10], като разработва теория и алгоритми за работа с т.н. $\mathcal{H}$ -матрици.

Методите използващи йерархични матрици са част от по-общата група от методи за решаване на системи чрез т.н. структурирани матрици. В [3] е направен обзор на съществуващите методи използващи такива матрици, включително и йерархични полусепарабелни матрици. В настоящата дисертация е изследвана ефективността на алгоритми на базата на този клас методи.

STRUMPACK (STRUctured Matrices PACKage) е паралелна софтуерна библиотека, която използва йерархична полусепарабелна компресия за решаване на системи от линейни алгебрични уравнения с плътни матрици [21]. Алгоритъмът включва три стъпки:

1. *Йерархична полусепарабелна компресия (апроксимация) на матрицата на системата.* При наличието на определени предположения изчислителната сложност на тази стъпка е $O(r^2n)$, където r е максималният ранг на извъндиагоналните блокове на апроксимиращата матрица изчислен в процеса на компресия. В общият случай сложността е $O(rn^2)$
2. *ULV-подобна факторизация.* На тази стъпка се факторизира компресираната матрица. За целта се прилага вариант на метода на Гаус, подобен на използвания при представянето на LU факторизацията. Най-напред се изключват $O(r)$ неизвестни, след което се изключват останалите $O(n-r)$. Изчислителната сложност на тази стъпка е $O(r^2n)$.
3. *Решение.* Тази стъпка използва компресираната и факторизирана матрица от коефициенти на системата и дясната страна за намиране на решението. Изчислителната сложност на тази стъпка е $O(rn)$.

Така общата изчислителна сложност на метода е $O(r^2n)$. Както ще видим по-късно, тази оценка е в сила при определени предположения.

1.5 Йерархична полусепарабелна компресия

В този раздел ще разгледаме накратко йерархичните полусепарабелни матрици (Hierarchically Semi-Separable – HSS). Те са въведени от Мартинсон в [16]. В [21] са описани алгоритмите използвани в STRUMPACK за решаване на системи от линейни уравнения с плътни матрици. Йерархичната компресия може да се приложи върху всяка неособена матрица, но е *ефективна* само, ако изходящата матрица A има подходяща *структура* – т.е. извъндиагоналните ѝ блокове имат нисък ранг. Под *ефективна* компресия разбираме апроксимация на матрицата, което води до съществено намаляване на изчислителната сложност на операциите с компресираната матрицата, както и на паметта необходима за нейното съхраняване.

Означаваме HSS компресираната апроксимация на матрицата A с H . Алгоритъмът може да се опише по следния начин:

1. Разделяме матрицата A на четири блока. Предполагаме, че извъндиагоналните блокове имат нисък ранг (и могат да се декомпозират по сингулярни стойности (Singular Value Decomposition – SVD) или друга факторизация, която изчислява ранг):

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} = \begin{bmatrix} D_1 & U_1^{\text{big}} B_{1,2} V_2^{\text{big}*} \\ U_2^{\text{big}} B_{2,1} V_1^{\text{big}*} & D_2 \end{bmatrix}.$$

Матриците U , B и V се наричат *генератори*. Ако извъндиагоналните блокове имат нисък ранг, матриците U са „високи и тънки“, B са малки и квадратни (или близки до квадратни) и матриците V са „ниски и широки“. Съотношението на броя на колоните и редовете зависи от ранга на извъндиагоналните блокове. D са непроменените диагонални блокове на изходящата матрица A . Означението „big“ ще бъде обяснено по-долу в точка 3.

2. Предполагаме, че диагоналните блокове D също имат извъндиагонални блокове с нисък ранг. Те се компресират по аналогичен начин, като процесът продължава рекурсивно. Второто ниво на рекурсивна компресия има вида:

$$A = \begin{bmatrix} \begin{bmatrix} D_1 & U_1^{\text{big}} B_{1,2} V_2^{\text{big}*} \\ U_2^{\text{big}} B_{2,1} V_1^{\text{big}*} & D_2 \end{bmatrix} & U_3^{\text{big}} B_{3,6} V_6^{\text{big}*} \\ U_6^{\text{big}} B_{6,3} V_3^{\text{big}*} & \begin{bmatrix} D_4 & U_4^{\text{big}} B_{4,5} V_5^{\text{big}*} \\ U_5^{\text{big}} B_{5,4} V_4^{\text{big}*} & D_5 \end{bmatrix} \end{bmatrix}$$

3. Съществува рекурсивна зависимост между *генераторите* на различните нива компресия. Това обяснява и използването на означенията „big“. В сила са следните зависимости:

$$U_3^{\text{big}} = \begin{bmatrix} U_1^{\text{big}} & 0 \\ 0 & U_2^{\text{big}} \end{bmatrix} U_3 \quad \text{и} \quad V_3^{\text{big}} = \begin{bmatrix} V_1^{\text{big}} & 0 \\ 0 & V_2^{\text{big}} \end{bmatrix} V_3 \quad (1)$$

Третото ниво на рекурсивна HSS компресия се записва във вида:

$$A = \begin{bmatrix} \begin{bmatrix} D_1 & U_1^{\text{big}} B_{1,2} V_2^{\text{big}*} \\ U_2^{\text{big}} B_{2,1} V_1^{\text{big}*} & D_2 \end{bmatrix} & \begin{bmatrix} U_1^{\text{big}} & 0 \\ 0 & U_2^{\text{big}} \end{bmatrix} U_3 B_{3,6} V_6^* \begin{bmatrix} V_4^{\text{big}*} & 0 \\ 0 & V_5^{\text{big}*} \end{bmatrix} \\ \begin{bmatrix} U_4^{\text{big}} & 0 \\ 0 & U_5^{\text{big}} \end{bmatrix} U_6 B_{6,3} V_3^* \begin{bmatrix} V_1^{\text{big}*} & 0 \\ 0 & V_2^{\text{big}*} \end{bmatrix} & \begin{bmatrix} D_4 & U_4^{\text{big}} B_{4,5} V_5^{\text{big}*} \\ U_5^{\text{big}} B_{5,4} V_4^{\text{big}*} & D_5 \end{bmatrix} \end{bmatrix} \quad (2)$$

Генераторите с означения „big“ могат да не се изчисляват извън най-високите нива на рекурсивната компресия. U се изчислява от U_τ и от U^{big} в по-високите нива на компресия. В последното ниво на компресия $U = U^{\text{big}}$.

В общия случай равенство 2 не е точно, а приблизително. Това означава, че в резултат на HSS компресията получаваме апроксимация на A , която означаваме с H , т.е. $H \approx A$.

За целта се избира подходящ трешхолд (прагова стойност) ε , който е необходим при изчисляването на *генераторите*. Когато се използва по-голям трешхолд, се получават по-малки *генератори* и съответно компресираната матрица заема по-малко памет и позволява по ефективни операции с нея, но това е за сметка на точността. При избор на по-малък трешхолд е точно обратното.

Както ще покажем в следващите глави, подреждането на неизвестните при асемблиране на матрицата A влияе съществено на ефективността на HSS компресията. Ако матрицата A се пренареди произволно, това с голяма вероятност може да унищожи каквато и да е подходяща за този метод *структура*.

За определени класове задачи е възможно така да се пренаредят неизвестните, че да се подобри съществено *структурата* на матрицата на системата. Така например в [19] са разгледани няколко метода за клъстеризация при използването на хребетообразна регресия с ядро (Kernel Ridge Regression). В Глава 3 ще предложим и анализираме няколко метода за пренареждане на неизвестните за система от линейни алгебрични уравнения, получена при дискретизация на елиптична задача с *дробна* степен на оператора на Лаплас (дробно-дифузионна задача).

1.6 Компресия със случайни извадки

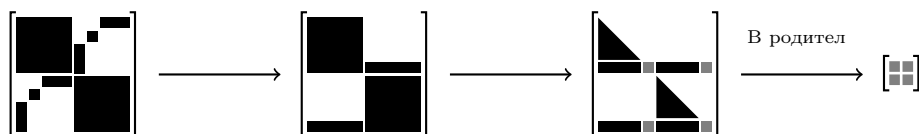
Алгоритъмът за HSS компресия в STRUMPACK се основава на използването на случайни извадки (randomized sampling), който прилага умножаване на множество от случайни вектори с изходната матрица A . Този метод е предложен от Мартинсон в [16]. В алгоритъма не се изисква да разполагаме с матрицата A в явен вид. Вместо това е нужна само функция за умножение на A с вектор. Премумствата на този подход, както и адаптивен алгоритъм за случайни извадки, са разгледани от Горман и др. в [9]. Използването на случайни извадки е полезно също така при интегрирането на HSS ядра в солвъри за задачи с разреждени матрици [8].

В общия случай изчислителната сложност на умножението на плътна матрица с вектор е $O(n^2)$. Това води до сложност на HSS компресирането $O(rn^2)$. За определени класове задачи r е много по-малко от n . Така например за двумерни задачи на Поасон (МКЕ) r е константа, а за тримерни задачи на Хелмхолц (МГЕ) расте бавно при нарастване на n . Ако разполагаме с бърз алгоритъм за умножение на компресираната матрица с вектор, сложността на компресията може да се намали до $O(r^2n)$.

1.7 ULV-подобна факторизация и решение

Компресираната матрица H в HSS форма може да се факторизира със специален вид LU факторизация, наречена ULV факторизация [5]. Тази факторизация използва ортогонални трансформации за последователно изключване на първите $n-r$ неизвестни. Останалите r неизвестни се изключват с LU факторизация. В STRUMPACK е реализирана ULV-подобна факторизация, при която вместо ортогонални трансформации се използва HSS структурата на компресираната матрицата H .

Процесът на ULV-подобна факторизация е илюстриран на Фигура 1.



Фигура 1: ULV-подобна факторизация.

След прилагане на ULV-подобната факторизация системата от линейни алгебрични уравнения $Ax = b$ се свежда до решаване на две системи с триъгълни матрици. Изчислителната сложност на тази последна стъпка е $O(rn)$ [21].

Глава 2 Метод на граничните елементи за числено решаване на двумерна задача за обтичане на крилни профили

В тази глава се разглежда числен метод за компютърна симулация на ламинарен поток около крилни профили на Жуковски. В дисертацията е приложен методът описан в [18], като е разработена програмна реализация за обтичане на каскада от крилни профили от идеален флуид. Методът е базиран на колокация на сплайни с на части

линейна интерполация. В [23] е представен паралелен код на програмния език C за решаване на разглежданата задача.

След дискретизиране на интегралните уравнения по метода на граничните елементи се получава система от линейни уравнения с плътна матрица. Резултатите от прилагане на изследваните в дисертацията методи и алгоритми се сравняват с резултати получени по метода на последователно изключване (метод на Гаус) реализиран в няколко популярни софтуерни пакета. Върху CPU процесори в сравнителния анализ на производителността използваме Intel Math Kernel Library (MKL) и Parallel Linear Algebra for Scalable Multi-core Architectures (PLASMA), докато за Intel Xeon Phi копроцесорите (накратко наричани MIC от името на архитектурата Many Integrated Core) MKL производителността се сравнява и с Matrix Algebra on GPU and Multicore Architectures (MAGMA) за MIC архитектура (наричан за кратко MAGMA MIC).

2.1 Постановка на задачата

2.2 Метод на граничните елементи за пресмятане на токовата функция на идеален флуид в неограничена двумерна област

Нека $\Omega \subset \mathbb{R}^2$ е неограничена многосвързана област с достатъчно гладка вътрешна граница S . За разглежданата задача функцията на тока Ψ удовлетворява уравнението на Лаплас

$$\nabla^2 \Psi \equiv \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0 \quad (3)$$

в $\Omega \subset \mathbb{R}^2$ и може да се представи във вида

$$\Psi(P) = -\frac{1}{4\pi} \int_S \gamma(\sigma) \ln(r^2(P, Q)) d\sigma_Q + \Psi_\infty(P) + C_0, \quad P \in \Omega. \quad (4)$$

където $r^2(P, Q) = (x - \xi)^2 + (y - \eta)^2$, $P = (x, y)$, $Q = (\xi, \eta)$ и $d\sigma_Q$ е мярка върху S . Първият член на дясната страна съответства на прост слой с плътност $\gamma(\sigma)$, $\Psi_\infty(P)$ е хармонична функция, добавена, за да бъдат удовлетворени условията на външната граница, т.е. при $P \rightarrow \infty$. При тези предположения за полето на скоростите $\vec{C} = (u, v)$,

$$u = \frac{\partial \Psi}{\partial y}, \quad v = \frac{\partial \Psi}{\partial x}$$

са в сила уравненията

$$u = \frac{1}{2\pi} \int_S \gamma(\sigma) \frac{y - \eta}{r^2} d\sigma, \quad v = -\frac{1}{2\pi} \int_S \gamma(\sigma) \frac{x - \xi}{r^2} d\sigma.$$

2.3 Обтичане на крилни профили

В този раздел ще разглеждаме задачата за обтичане на крилни профили на Жуковски. Приемаме, че флуидното течение в безкрайност е с хомогенна скорост $\vec{C}_\infty = (1, 0)$. Тук с S са означени контурите на крилните профили. За разглежданата задача токовата функция Ψ удовлетворява уравнението на Лаплас (3). Крилните профили S

са непроницаеми. Следователно е в сила граничното условие $\Psi|_S = K = \text{const}$. За да удовлетворим условията в $\vec{C}_\infty$, избираме такова Ψ_∞ , че

$$\vec{C}_\infty = \left(\frac{\partial \Psi_\infty}{\partial y}, -\frac{\partial \Psi_\infty}{\partial x} \right).$$

В конкретната задача използваме

$$\Psi_\infty(P) = \gamma_\infty(P).$$

Така интегралното уравнение (4) се записва във вида

$$\gamma(P) - \frac{1}{4\pi} \int_S \gamma(\sigma) \ln(r^2(P, Q)) d\sigma_Q + C = 0. \quad (5)$$

За получаване на единствено решение на интегралното уравнение се използва условието на Рунге-Кута $\gamma(A) = 0$, където A са върховете на острите ъгли на крилните профили.

2.4 Дискретизация

За численото решаване на интегралното уравнение (5) прилагаме метода на граничните елементи. Така непрекъснатата задача се свежда до система от линейни алгебрични уравнения, която можем да запишем във вида:

$$(A\gamma)s = f(s).$$

Приближеното решение се търси във вида

$$\gamma_h(S) = \sum_{i=1}^n \gamma_i \phi_i(s).$$

Тук $\{\phi_i(s)\}_{i=1}^n$ е Лагранжевият базис на пространството от на части линейни функции по границата S , върху която е дефинирана мрежата S_h . С $\gamma_i = \gamma_h(s_i)$, $i = 1, \dots, n$ са означени неизвестните стойности на приближеното решение във възлите на мрежата. За реализация на метода на граничните елементи прилагаме метод на колокацията с колокационни точки в средите на елементите от S_h . Следвайки [23] получаваме системата от линейни уравнения

$$\sum_{i=1}^n \gamma_i \Psi_{ji} = f_j, \quad j = 1, 2, \dots, n, \quad (6)$$

където $\Psi_{ji} = \Psi_i(s_j)$, $f(s_j) = f_j$, $\Psi_i(s) = (A\phi_i)(s)$.

В Раздел 2.5 са представени числени експерименти за задачата за обтичане на пет крилни профила разположени вертикално един над друг.

2.5 Анализ на числени експерименти върху компютърни системи с обща памет

Формирането на матрицата D има сложност $O(n^2)$. Сложност $O(n)$ има и изчисляването на коефициентите на челно съпротивление и подемна сила. Фокусът на изследване в тази работа е върху най-тежката изчислителна част – решаването на системата линейни алгебрични уравнения, което с метод на Гаус има сложност $O(n^3)$ [14].

2.6 LU факторизация

2.6.1 CPU процесори с обща памет

PLASMA (Parallel Linear Algebra Software for Multicore Architectures) [7] е софтуерен пакет за решаване на системи линейни уравнения с плътни матрици, който имплементира функциите на стандарта LAPACK.

Ефективността на PLASMA е базирана на силно оптимизирания пакет BLAS (Basic Linear Algebra Subprograms), в който са реализирани основните операции в линейната алгебра – умножение на вектори, матрици и вектори с матрици. За това ниво на изчисления използваме MKL BLAS и ATLAS (Automatically Tuned Linear Algebra Software) BLAS [27].

В Таблица 1 са представени резултатите от проведените числени експерименти за решаване на линейните системи получени при прилагане на метода на граничните елементи за дискретизация на задачата за обтичане на крилни профили. Сравнени последователните и паралелни времена за PLASMA + ATLAS, PLASMA + MKL и MKL за $n = 5\,000$ и $n = 40\,000$, варирайки броя на нишките.

Таблица 1: Последователни и паралелни времена за решаване на системата върху CPU процесори с обща памет

Софтуер		Plasma + ATLAS		PLASMA + MKL		MKL	
Нишки	n	време [s]	ускорение	време [s]	ускорение	време [s]	ускорение
1	5 000	8.42	1.00	5.03	1.00	5.30	1.00
16	5 000	0.67	12.57	0.47	10.69	0.47	11.26
32	5 000	0.88	9.59	0.65	7.76	0.65	8.12
1	40 000	4008.76	1.00	2497.12	1.00	2233.93	1.00
16	40 000	282.94	14.17	166.41	15.01	147.64	15.13
32	40 000	325.17	12.33	169.58	14.73	148.59	15.03

Резултатите показват добро ускорение за всички тествани библиотеки до 16 нишки. Ускорението достига до 15 и е близо до теоретичния максимум 16.

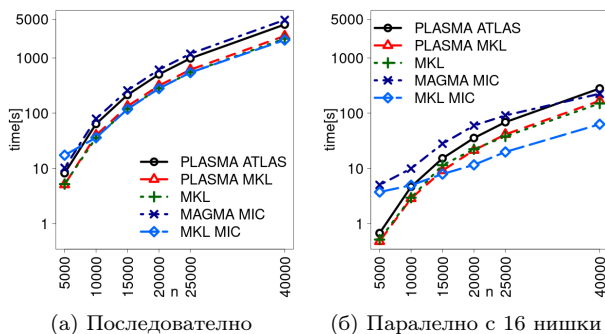
Паралелната ефективност на PLASMA с MKL и MKL е близка, като достига 94% за най-голямата задача ($n = 40\,000$). И двата варианта превъзхождат PLASMA с ATLAS повече от 1.5 пъти.

2.6.2 MIC ускорители

В този раздел анализираме паралелната ефективност на Intel Xeon Phi 7120P копроцесори (MICs). MIC-овете са проектирани за масивни паралелни и векторни изчисления необходими при високопроизводителните изчисления. Използваният модел MIC има

Таблица 2: Последователни и паралелни времена и ускорения за решаване на системата върху МІС копроцесорите

Пакет	Нишки	n	MAGMA МІС		MKL	
			време [s]	ускорение	време [s]	ускорение
1	5 000		11.70	1.00	17.64	1.00
60	5 000		5.81	2.01	2.86	6.16
120	5 000		6.76	1.73	3.55	4.97
240	5 000		5.39	2.17	3.80	4.64
1	40 000		4896.49	1.00	2101.93	1.00
60	40 000		665.53	7.36	154.23	13.63
120	40 000		432.89	11.31	93.80	22.41
240	40 000		208.48	23.49	64.43	32.62



Фигура 2: Сравнение на производителността на софтуерните библиотеки за CPU и МІС.

61 ядра, като всяко ядро може да изпълнява едновременно инструкции от 4 нишки, т.е. могат да се използват максимум 244 нишки. Използваме Offload режим, при който едно от ядрата се запазва за комуникация с CPU процесора и можем да използваме до 60 ядра (240 нишки).

В Таблица 2 са представени резултати от числени експерименти за решаване на системите при $n = 5\,000$ и $n = 40\,000$, варирайки броя на нишките от 1 до 240. Резултатите показват много по-добра производителност на MKL в сравнение с MAGMA MIC. Това може да се дължи на по-добрата комуникация между нишките в MKL. За по-голямата задача постигнатото паралелно ускорение е 32.

На Фигура 2 е сравнена производителността на използваните софтуерни пакети за CPU и МІС архитектурата.

Производителността на PLASMA с MKL за решаване на системите върху CPU е по-добра от тази на MAGMA MIC. Това може да се дължи на по-добра комуникация между нишките.

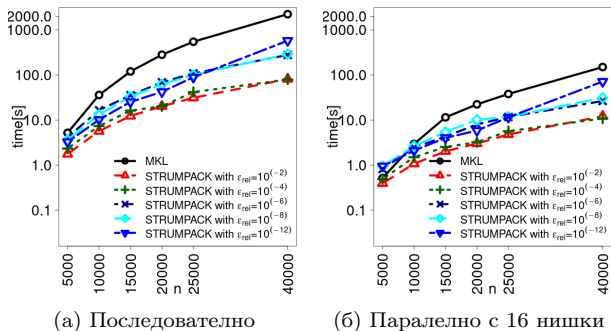
2.7 Йерархична полусепарабелна компресия

Йерархичната полусепарабелна компресия имплементирана в софтуерния пакет STRUMPACK е приблизителна. Това означава, че компресираната матрица H

апроксимира изходящата матрица A . Потребителят задава два прага на грешката – абсолютен ε_{abs} и относителен ε_{rel} [9]. В представените в дисертацията експерименти абсолютният праг е фиксиран на $\varepsilon_{\text{abs}} = 10^{-8}$, като варираме относителния праг $\varepsilon_{\text{rel}} = 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$ и 10^{-12} .

2.7.1 Сравнителен анализ на йерархична и LU факторизация върху CPU с обща памет

В този раздел анализираме производителността на метода на полусепарабелна йерархична (HSS) компресия и неговата софтуерна реализация STRUMPACK в сравнение с метода на Гаус и неговата най-добра (на базата на анализа, изложен по-горе) реализация MKL, където алгоритъмът използва блочна LU факторизация. На Фигура 3 са представени последователните (Фигура 3а) и паралелните (Фигура 3б) времена за решаване на системата линейни уравнения. Резултатите потвърждават по-добрата изчислителна сложността на HSS компресията ($O(n^2r)$) в сравнение с LU факторизацията ($O(n^3)$). Ясно се вижда и въздействието на избрания относителен праг на грешката ε_{rel} .



Фигура 3: Производителност на STRUMPACK сравнена с MKL

За последователните експерименти STRUMPACK е много по-ефективен от най-добрия пряк солвър, който използваме – MKL. STRUMPACK показва по-малки паралелни ускорения, съответно от ~ 2 до ~ 5 . Това се дължи на по-сложната рекурсивна структура на HSS компресията.

2.7.2 Анализ на грешката за HSS-базирания солвър

Нека припомним отново, че HSS компресията е приблизителна, т.е. компресираната матрица H е приближение на матрицата A . Полученото с помощта на HSS компресия решение на системата линейни алгебрични уравнения е апроксимация на точното. За оценка на грешката приемаме за референтно решението по метода на Гаус с пряк солвър на базата на LU факторизация (виж Раздел 2.6). Тук анализираме относителната грешка R_{relative} дефинирана както следва:

$$R_{\text{relative}} = \frac{\|x^{\text{Gauss}} - x^{\text{HSS}}\|_{l_2}}{\|x^{\text{Gauss}}\|_{l_2}} = \frac{\sqrt{\sum_{i=1}^n (x_i^{\text{Gauss}} - x_i^{\text{HSS}})^2}}{\sqrt{\sum_{i=1}^n (x_i^{\text{Gauss}})^2}}, \quad (7)$$

където x^{Gauss} е решението получено с гаусовия солвър от пакета MKL, което се използва като референтно, а x^{HSS} е решението получено при използване на йерархична полусепарабелна компресия.

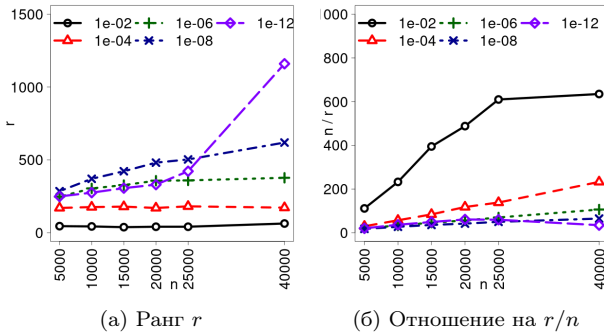
В Таблица 3 са показани относителните грешки R_{relative} , варирайки размерността на задачата $n \in \{5\,005, 10\,005, 15\,005, 20\,005, 25\,005, 40\,005\}$, както и праговете на относителна грешка $\varepsilon_{\text{rel}} \in \{10^{-6}, 10^{-8}, 10^{-12}\}$.

Таблица 3: Относителна грешка R_{relative}

n	ε_{rel}			n	ε_{rel}		
	10^{-6}	10^{-8}	10^{-12}		10^{-6}	10^{-8}	10^{-12}
5005	1.1	0.085	0.00019	15005	0.29	0.23	0.00097
20005	0.28	0.34	0.0038	25005	0.3	1.48	0.013
10005	0.75	0.17	0.00075	40005	0.37	1.59	0.027

Точността и изчислителната ефективност на йерархичния метод зависят от максималния ранг на извъндиагоналните блокове r , който се определя от избраните прагове на грешката и зависи от структурата на изходната матрица A . По-големият ранг r съответства на по-малка относителна грешка R_{relative} , както и на по-дълго време за решаване на системата.

Анализът на представените резултати показва, че за постигане на висока точност на метода може да е необходим много малък праг съответстващ на голям ранг r . В такъв случай методът на HSS компресия може да не бъде достатъчно ефективен.



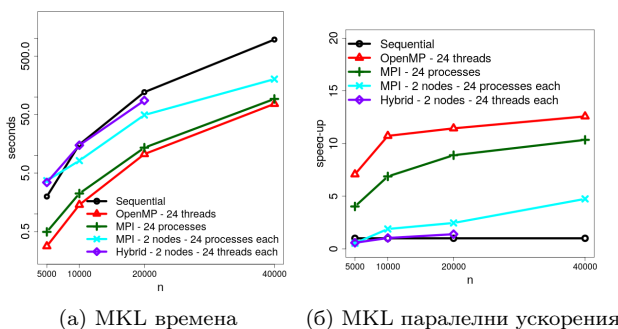
Фигура 4: Максимален извъндиагонал ранг r

2.8 Паралелна скалируемост върху компютърни системи с разпределена памет

Този раздел е посветен на анализ на някои специфични особености и трудности при работа с изчислителни системи с хибридна архитектура. Те имат разпределена памет на нивото на сървърите, от които са изградени и обща памет в рамките на всеки сървър. Представените резултати са публикувани в [24]. Както и в предишния раздел, сравнителният анализ включва паралелните библиотеки MKL и STRUMPACK. За решаване на системата от линейните алгебрични уравнения, получена при дискретизация на задачата за обтичане на профили на Жуковски са използвани един или два сървъра, свързани с етернет. Анализирани са следните варианти на изпълнение на MKL и STRUMPACK: Последователно; Паралелизация с 24 OpenMP нишки; Паралелизация с 24 MPI процеса на 1 сървър; Паралелизация с 48 MPI процеса на 2 сървъра; Хибридна паралелизация с 2 MPI процеса на 2 сървъра, всеки с 24 OpenMP нишки.

2.9 LU факторизация

На Фигура 5 са показани времената за решение на системата от линейни алгебрични уравнения с MKL. Най-добро време се получава с OpenMP при използване на един сървър. Паралелното ускорение на един сървър е най-добро за OpenMP, следвано от MPI (Фигура 5б). Това се обяснява с относително по-бавните етернет комуникации между процесите при използване на повече от един сървър.



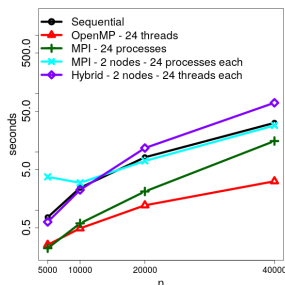
Фигура 5: Паралелни времена и ускорения за решаване на системите с MKL.

2.10 HSS компресия

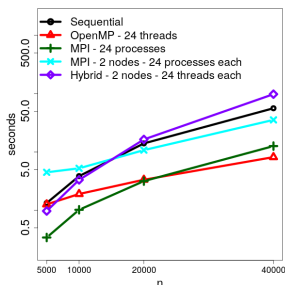
При решаване на системата с използвана HSS компресия се получават по-малки паралелни ускорения отколкото при прекия Гаусов солвър (виж Фигура 6). Това може да се обясни с рекурсивната структура на HSS компресията.

При числените експерименти с най-ниския праг на относителна грешка ($\varepsilon_{\text{rel}} = 10^{-8}$) се получава най-добро време за изпълнение при използване на

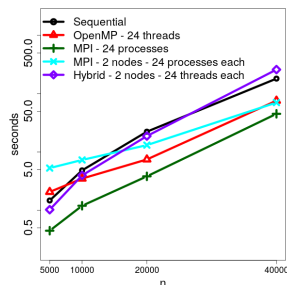
MPI върху един сървър. Това най-вероятно се дължи на факта, че OpenMP паралелизацията в STRUMPACK е направена по-късно в разработката на пакета от MPI.



(a) STRUMPACK времена при $\varepsilon_{rel} = 10^{-2}$



(б) STRUMPACK времена при $\varepsilon_{rel} = 10^{-6}$



(в) STRUMPACK времена при $\varepsilon_{rel} = 10^{-8}$

Фигура 6: Времена и паралелни ускорения.

Поведението на паралелното ускорение се променя съществено при използване на два сървъра. Това се дължи на (относително) бавната връзка между тях – 1000 Mb Ethernet. Можем да направим извода, че ефективността може значително да се подобри при по-бърза комуникационна среда (например InfiniBand), както и при увеличаване на размера на решаваните системи линейни алгебрични уравнения.

2.11 Заключителни бележки

Централно място в представените резултати заема изследването на метода на йерархична полусепарателна компресия (HSS компресия). Експерименталният сравнителен анализ е на базата на неговата реализация в софтуерния пакет STRUMPACK. Той показва по-добро бързодействие от преките Гаусови солвъри, използващи блочна LU факторизация. В същото време получените паралелни ускорения със STRUMPACK са по-малки, което се обуславя от по-сложната йерархична структура на алгоритъма.

Точността и изчислителната ефективност на HSS компресията зависят от праговете на относителна и абсолютна грешка. Това са параметри, които се избират от потребителя. Представеният анализ показва как да получим най-добра ефективност при зададена точност.

Глава 3 Метод на крайните елементи за числено решаване на двумерна стационарна задача за дробна дифузия

Дробните елиптични оператори по пространството на степен $\alpha \in (0, 1)$ описват процеси на *аномална* дифузия. Свързаните с тях гранични задачи са нелокални и, в общия случай, численото решение на такива задачи е изчислително скъп процес. Такъв тип нелокални модели се прилага например в обработката на изображения, финансовата

математика, електро-магнитостатиката, перидинамиката, моделирането на течения в порести среди и много други.

Представените числени експерименти са за моделни задачи в квадратна и кръгла област, като авторефератът се ограничава с резултати само за първата. Дробният лапласиан се дефинира с помощта на потенциал на Риц (Riesz). Теоретичната постановка на задачата и разработеният специализиран метод на крайните елементи за нейното числено решаване са представени в статията на Акоста и съавтори [1]. В [2] авторите описват алгоритмичната реализация на метода.

В Глава 2 бяха анализирани няколко софтуерни пакета реализиращи методи от тип гаусова елиминация. В тази глава е използван само най-ефективният от тях: Intel's Math kernel Library (MKL). Анализирана е производителността на алгоритъма базиран на йерархична полусепарабелна компресия (HSS), реализиран в пакета STRUctured Matrix PACKage (STRUMPACK). Изследвани са няколко метода за пренареждане на неизвестните с цел подобряване ефективността на HSS компресията.

3.1 Постановка на задачата

Дробният лапласиан може да се представи във вида

$$(-\Delta)^\alpha u(x) = C(d, \alpha) \text{ P.V. } \int_{\mathbb{R}^n} \frac{u(x) - u(y)}{|x - y|^{d+2\alpha}}, \quad (8)$$

където P.V. означава главна стойност, d е размерността, $\alpha \in (0, 1)$, а $C(d, \alpha)$ е нормализиращата константа

$$C(d, \alpha) = \frac{2^{2\alpha} \alpha \Gamma(\alpha + \frac{d}{2})}{\pi^{d/2} \Gamma(1 - \alpha)},$$

където Γ е гамма функцията.

В тази глава е използвана интегралната дефиниция на дробен лапласиан, в съответствие с възприетата постановка на задачата в статията на Акоста [2]. Така разглеждаме следната гранична задача за *дробния* оператор на Лаплас

$$\begin{cases} (-\Delta)^\alpha u(x) = f(x), & x \in \Omega \\ u(x) = 0, & x \in \Omega^c. \end{cases} \quad (9)$$

Тук $\Omega \subset \mathbb{R}^d$ е ограничена отворена област, Ω^c е допълнението на Ω в $\mathbb{R}^d$ и $f(x)$, $x \in \Omega$, е дясна част с достатъчна гладкост.

Вариационната формулировка на (9) се получава, като уравнението се умножи с тестова функция и се интегрира по части. Така за слабото решение получаваме уравнението: търси се $u \in H^\alpha(\Omega)$, такова че

$$\frac{C(d, \alpha)}{2} \langle u, v \rangle_{H^\alpha \mathbb{R}^d} = \int_{\Omega} f v, \quad v \in \tilde{H}^\alpha(\Omega). \quad (10)$$

Скаларното произведение на u и v е дефинирано в Хилбертовото пространство $H^\alpha(\Omega)$ с норма $\| \cdot \|_{H^\alpha(\Omega)} = \| \cdot \|_{L^2(\Omega)} + | \cdot |_{H^\alpha(\Omega)}$. Тук $| \cdot |_{H^\alpha(\Omega)}$ е полунормата на Ароншайн-Слободецкий. $\langle u, v \rangle_{H^\alpha \mathbb{R}^d}$ може да се запише във вида

$$\langle u, v \rangle_{H^\alpha \mathbb{R}^d} = \iint_{\mathbb{R}^d \times \mathbb{R}^d} \frac{(u(x) - u(y))(v(x) - v(y))}{|x - y|^{d+2\alpha}} dx dy.$$

Ще обърнем внимание на факта, че интегрирането е върху цялото пространство $\mathbb{R}^d$.

Коректността на вариационната постановка на задачата (10), както и съществуването и единствеността на решение в $\tilde{H}^\alpha(\Omega)$ следват от лемата на Лакс-Милграм (Lax-Milgram).

3.2 Постановка на метода на крайните елементи

Нека $\mathcal{T}$ е допустима триангулация на областта Ω съставена от $N_{\mathcal{T}}$ триъгълни крайни елементи. Разглеждаме крайноелементното пространство $\mathbb{V}_h$ от непрекъснати на части линейни функции върху $\mathcal{T}$. Нека $\{\varphi_1, \dots, \varphi_N\} \subset \mathbb{V}_h$ е лагранжев възлов базис съответстващ на върховете на триъгълниците от $N_{\mathcal{T}}$ означени с $x_1, \dots, x_N$, които не са на границата $\partial\Omega$. Тогава $\varphi_i(x_j) = \delta_j^i$. Нека $T \in \mathcal{T}$ е даден елемент от триангулацията и нека означим с h_T и ρ_T съответно диаметъра и радиуса на вписаната в T окръжност, като $h = \max_{T \in \mathcal{T}} h_T$. Разглеждаме регулярни по форма триангулации, за които съществува $\tau > 0$ независимо от $\mathcal{T}$, такова че

$$h_T \leq \tau \rho_T, \quad \forall T \in \mathcal{T}.$$

При тези предположения за всяко $\alpha \in (0, 1)$ дискретният аналог на вариационната задача (10) има вида

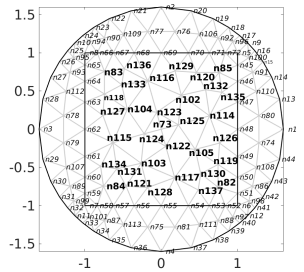
$$\frac{C(d, \alpha)}{2} \langle u_h, v_h \rangle_{H^\alpha(\mathbb{R}^n)} = \int_{\Omega} f v_h, \quad v_h \in \mathbb{V}_h. \quad (11)$$

Тук с $u_h = \sum_j u_j \varphi_j$ е означено численото решение по метода на крайните елементи (МКЕ) на стационарната задача за дробна дифузия (9). Решаването на дискретната вариационна задача се свежда до система от линейни алгебрични уравнения във вида $KU = F$, където $U = (u_j) \in \mathbb{R}^N$ са неизвестните възлови стойности. Матрицата на коравина K е симетрична и положително определена, и следователно системата има единствено решение.

В приложения в [1] вариант на МКЕ, интегрирането се редуцира до кръгова област $B \supset \Omega$, такава че разстоянието между границата $\bar{\Omega}$ и допълнението B^c е достатъчно голямо. Добавя се допълнителната триангулация $\tilde{\mathcal{T}}_A$ върху $B \setminus \Omega$, такава че общата триангулация $\tilde{\mathcal{T}} = \mathcal{T} \cup \tilde{\mathcal{T}}_A$ върху B е допустима. На Фигура 7 е показан пример на такава триангулация за квадратна област Ω . Възлите отбелязани с удебелен шрифт съответстват на неизвестните u_j .

Нека означим с $M_{\tilde{\mathcal{T}}}$ броя на елементите в триангулацията върху кръговата област B . Тогава елементите на матрицата на коравина K могат да се запишат във вида

$$K_{ij} = \frac{C(d, \alpha)}{2} \sum_{\ell=1}^{N_{\tilde{\mathcal{T}}}} \left(\sum_{m=1}^{N_{\tilde{\mathcal{T}}}} I_{\ell, m}^{i, j} + 2J_{\ell}^{i, j} \right), \quad \ell, m \in [1, \dots, N_{\tilde{\mathcal{T}}}], \quad (12)$$



Фигура 7: Допустима триангулация за квадратна област Ω с обвиващ кръг B .

където за въведените интеграли I и J са в сила формулите

$$I_{\ell,m}^{i,j} = \int_{T_\ell} \int_{T_m} \frac{(\varphi_i(x) - \varphi_i(y))(\varphi_j(x) - \varphi_j(y))}{|x - y|^{2+2\alpha}} dx dy \quad (13)$$

$$J_\ell^{i,j} = \int_{T_\ell} \int_{B^c} \frac{\varphi_i(x)\varphi_j(x)}{|x - y|^{2+2\alpha}} dy dx. \quad (14)$$

3.3 Пренареждане на неизвестните

Както беше отбелязано в Раздел 1.5 за определени класове задачи е възможно неизвестните да бъдат пренаредени така, че да се подобри съществено *структурата* на матрицата на системата. Представеният в тази глава анализ показва, че за матриците породени от задачата за дробна дифузия пренареждането е необходимо.

На Фигура 9а са показани оригиналните номерации на възлите в триангулацията. Те се получават от MatLab функцията `initmesh` при генериране на крайноелементните мрежи. *Структурата* на матрицата съответстваща на това подреждане не е подходяща за йерархичния солвър от пакета STRUMPACK.

3.3.1 Пренареждане по Y координата – „top“

При това пренареждане съответстващите на възлите от триангулацията на Ω неизвестни се сортират по тяхната Y координата. Новата номерация и структурата на получената матрица са визуализирани на Фигура 9б.

3.3.2 Пренареждане по линии – „stripes“

При това пренареждане новата номерация на възлите от мрежата е по хоризонтални линии. Пример за това пренареждане и структурата на получената матрица са показани на Фигура 9в.

3.3.3 Пренареждане по спирала – „snake“

Този алгоритъм пренарежда неизвестните по спирала наподобяваща навита змия. Това пренареждане и структурата на получената матрица са визуализирани на Фигура 9г.

3.3.4 Пренареждане по метода на вложените сечения

Методът на вложените сечения (Nested Dissection) използва подхода „разделяй и владей“ за разделяне на графа представящ структурата на ненулевите елементи на зададена разрежена симетрична матрица.

Алгоритъмът включва три стъпки:

1. Конструирание на неориентиран граф, съответстващ на триангулацията в който върхове са възлите от мрежата а ребрата са страните на съответните триъгълници.

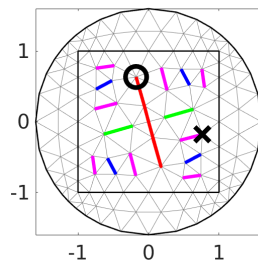
2. Рекурсивно разделяне на графа, използвайки разделители (такива малки подмножества от върхове, че когато се премахнат графът се разделя на два подграфа). Тази процедура продължава рекурсивно.
3. Пренареждане на възлите в съответствие с рекурсивната структура: първо по подграфи и след това по разделители.

Разделителите за тестовата задача в квадратна област са показани на Фигура 8. Пренареждането и получената структура на матрицата са визуализирани на Фигура 9г.

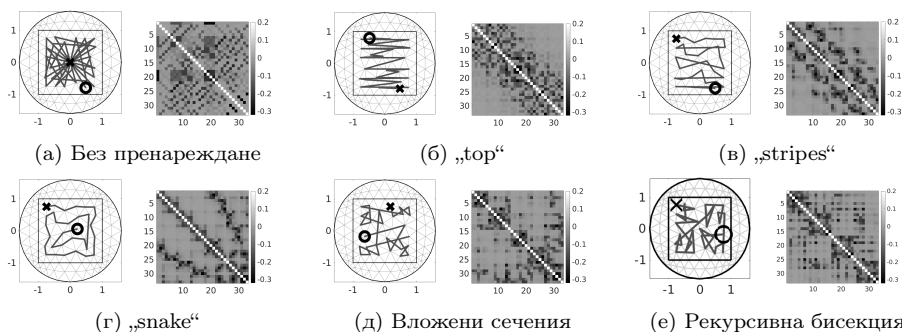
3.3.5 Пренареждане по метода на рекурсивна бисекция

Разглеждаме рекурсивната бисекция като алтернативен подход за разделяне на графа на две части [22]. За разлика от метода на вложените сечения, в този случай графът се разделя чрез премахване на множество от ребра. Процесът продължава рекурсивно, като всеки от двата подграфа се разделя на две по същият начин и така нататък. Частите на всяко ниво на рекурсивното разделяне се номерират последователно.

Рекурсивната бисекция се използва за балансиране на натоварването при решаване на задачи върху изчислителни системи с разпределена памет [22].



Фигура 8: Пренареждане с метод на вложените сечения. С „X“ е означен първия връх, а с „O“ последния.



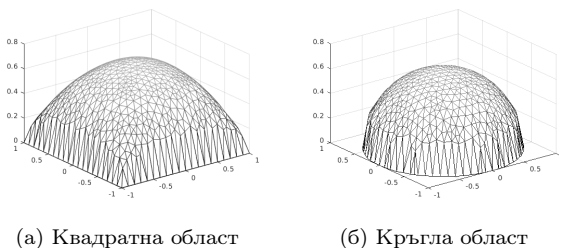
Фигура 9: Пренареждане на върховете в квадратна област Ω (горе) и структура на съответната матрица K . Тъмносивите линии показват подреждането на възлите от първия (маркиран с „X“) до последния (маркиран с „O“).

3.4 Анализ на числени експерименти върху компютърни системи с обща памет

Експерименталните резултати анализирани в този раздел са получени върху един сървър от суперкомпютъра AVITONOL. Оригиналната номерация на възлите (неизвест-

ните) се получава в резултат от генерирането на мрежата реализирано в програмата публикувана в статия [2]. С цел подобряване ефективността на йерархичния солвър от пакета STRUMPACK са анализирани няколко подхода за пренареждане на неизвестните.

Числените експерименти са за задачи с *дробен* лапласиан със степен $\alpha = 0.5$. Визуализация на численото решение на задачата за квадратна и кръгла област е показана на Фигура 10. В автореферата са представени резултати за квадратната област.



Фигура 10: Решение на задачата за *дробна* дифузия за квадратна и кръгла област.

В този раздел анализираме производителността на двата разглеждани метода на базата на тяхната реализация в следните софтуерни пакети за изчислителни системи с обща памет: LU факторизация от MKL и HSS компресия от STRUMPACK.

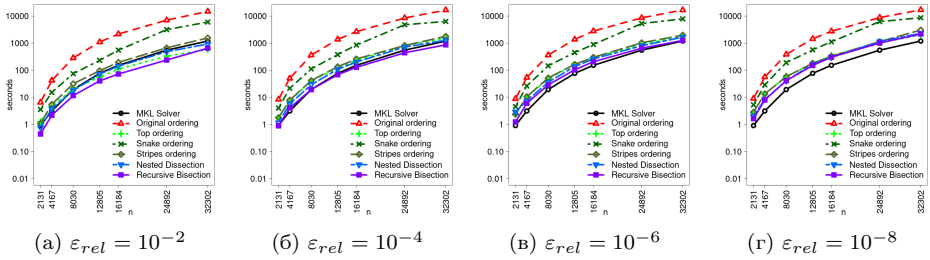
3.5 Квадратна област

На Фигура 11 са представени последователните времена (при използване на една нишка) за решаване на системата линейни алгебрични уравнения. В повечето случаи йерархичният солвър показва най-добри времена при рекурсивната бисекция, следвана от метод на вложените сечения, „top“ и „stripes“. Експериментите показват, че MKL има по-добра производителност от STRUMPACK за всички стойности на относителният праг с изключение на $\varepsilon_{\text{rel}} = 10^{-2}$, където резултатите за рекурсивната бисекция, „stripes“ и „top“ пренарежданията са по-добри.

Можем да направим също така извода, че като изключим варианта без пренареждане и пренареждането от тип „snake“, STRUMPACK показва относително близка до MKL производителност.

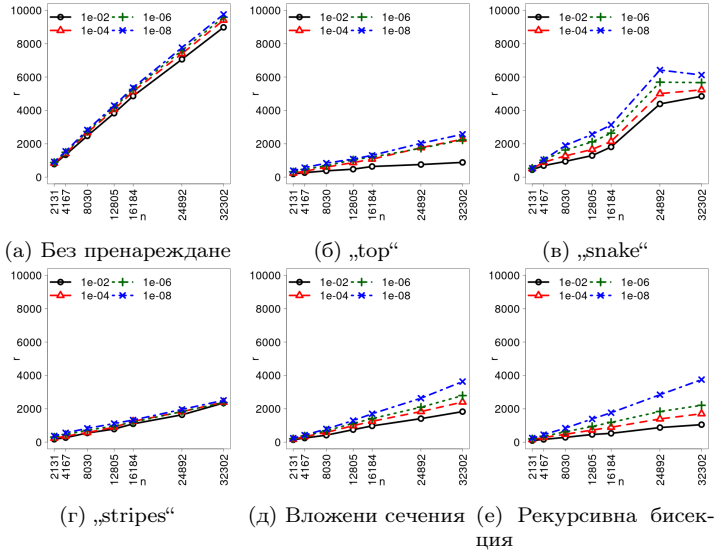
3.5.1 Извъндиагонален ранг

На Фигура 12 е представен извъндиагоналният ранг r изчислен в процеса на HSS компресията, като на Фигура 13 са показани отношенията n/r , където n е броят на неизвестните. Стойността на r е мярка за ефективността на компресията за дадената матрица. Колкото по-малък е рангът, толкова по-ефективна е компресията. При оригиналното подреждане стойностите на r са най-големи, като варират между $1/3$ и $1/4$ от броя на неизвестните. Следващ по големина ранг се получава при пренареждането „snake“. Рангът за останалите пренареждания има относително близки стойности,

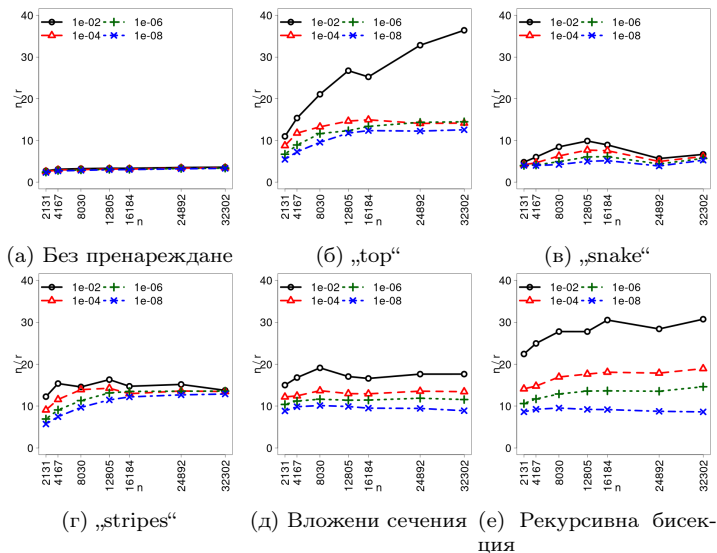


Фигура 11: Сравнителен анализ на времената за решаване на системата линейни алгебрични уравнения с MKL и STRUMPACK с пренареждания „top“, „snake“, „stripes“, вложени сечения (Nested Dissection) и рекурсивна бисекция (Recursive Bisection) за случая на квадратна област.

като за повечето от разгледаните примери за рекурсивната бисекция се получава най-висока ефективност на йерархичната полусепарабелна компресия. Ролята на ранга се потвърждава и от анализираните в предходния раздел резултати показани на Фигура 11.



Фигура 12: Максимален извъндиагонал ранг r .



Фигура 13: Отношение на броя на неизвестни n и максималния ранг r .

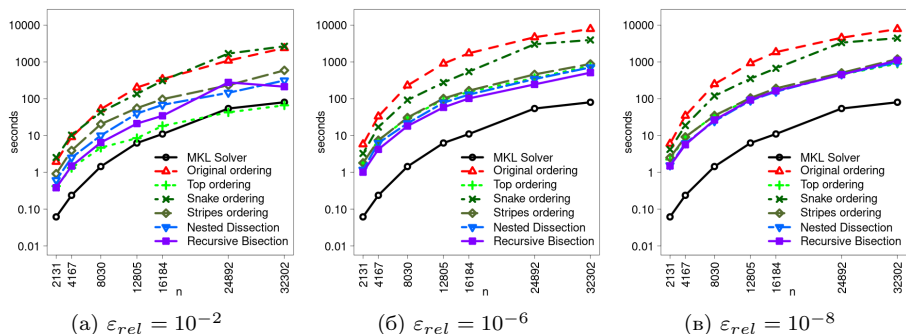
3.5.2 Паралелни времена и ускорения

В този раздел анализираме паралелната ефективност. Проведени са числени експерименти за матрици с нарастваща размерност $n \in [2\ 131, 32\ 302]$. На Фигура 14 са представени резултати получени при използване на 16 нишки. Виждаме, че графиките имат подобно поведение, като за по-големите задачи (по-големите стойности на n) ускоренията стават близки до линейни. Това е в съответствие с теоретичните оценки. Отбелязваме, че за $\varepsilon_{\text{rel}} = 10^{-2}$ и пренареждане „top“ йерархична HSS компресия е с по-добри времена за двете най-големи системи, т.е. при $n = 24\ 892$ и $n = 32\ 302$. В останалите случаи директният гаусов солвър MKL е по-бърз. Това се дължи и на по-сложната рекурсивна *структура* на HSS компресията. Можем да очакваме, че при по-големи стойности на n , STRUMPACK може да покаже по-добри времена от MKL и за по-малки стойности на прага на относителна грешка ε_{rel} .

3.5.3 Анализ на грешката за HSS-базирания солвър

При прилагане на HSS солвъра от пакета STRUMPACK получаваме *приближение* на решението на системата. Това се дължи на факта, че компресираната матрица H е апроксимация на изходната. Както и в предходната глава, ще анализираме относителната грешка на метода (7). В този анализ приемаме решението получено с директният гаусов солвър на MKL за референтно.

В Таблица 4 са представени относителните грешки за оригиналното подреждане и разгледаните пренареждания на неизвестните за квадратна област. За повечето экс-



Фигура 14: Сравнение на паралелните времена за решаване на системата линейни алгебрични уравнения с MKL и STRUMPACK с пренареждания „top“, „snake“, „stripes“, вложени сечения (Nested Dissection) и рекурсивна бисекция (Recursive Bisection) за квадратна област.

перименти относителната грешка е близка до зададения праг на относителна грешка ε_{rel} .

Относителната грешка зависи от ефективността на компресията. Това означава, че когато изчисленият ранг r на извъндиагоналните блокове е по-малък изходящата матрица се компресира по-ефективно, което води до по-малко изчислително време, но и до по-висока относителна грешка R_{relative} поради по-голямата компресия.

3.6 Заклучителни бележки

Експерименталният сравнителен анализ се базира на реализацията на HSS компресия и ULV-подобна факторизация в софтуерния пакет STRUMPACK. Анализът показва добро бързодействие на последователния алгоритъм в сравнение с прекия гаусов солвър, използващ блокна LU факторизация. В същото време, получените паралелни производителности и ускорения при използване на пакета STRUMPACK са по-малки, което се обяснява с по-сложната йерархична и рекурсивна структура на компресията.

Точността и изчислителната ефективност на HSS компресията съществено зависят от праговете на относителна грешка ε_{rel} и абсолютна грешка ε_{abs} , както и от наличието на подходяща *структура* на матрицата. При експериментите се наблюдава относителна грешка R_{relative} на численото решение близка до ε_{rel} . За подобряване на *структурата* на матрицата са предложени пет начина за пренареждане на неизвестните, които увеличават съществено ефективността на компресията. Представеният анализ показва, че при повечето експерименти ефективността на HSS компресията е най-добра при пренареждане с рекурсивната бисекция.

Структурата на матрицата получена при дискретизация на дробно дифузионната задача е по-малко подходяща за HSS компресия, в сравнение със задачата разгледана в Глава 2. Това може да се обясни с факта, че *дробният* лапласиан е силно нелокален. Предложените пренареждания на неизвестните значително подобряват ефективност-

Таблица 4: Относителна грешка за квадратна област.

(а) Без пренареждане пренареждане „top“.

n	Без пренареждане				„top“			
	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}
2131	0.0125	0.000124	1.633e-06	1.088e-08	0.122	0.00014	1.469e-06	1.003e-08
4167	0.0244	0.000211	2.195e-06	1.414e-08	0.194	0.000345	1.868e-06	1.95e-08
8030	0.0435	0.000422	7.515e-06	5.049e-08	0.237	0.00681	5.732e-06	3.446e-08
12805	6.74	0.0212	6.258e-06	5.04e-08	0.329	0.00976	4.3e-06	6.554e-08
16184	0.136	0.000818	1.307e-05	1.035e-07	0.335	0.00117	5.96e-06	6.16e-08
24892	0.115	0.000955	1.791e-05	1.659e-07	0.45	0.00192	4.920e-06	9.5e-08
32302	0.145	0.0011	3.409e-05	1.533e-07	0.479	0.00246	9.788e-06	9.09e-08

(б) Пренареждания „snake“ и „stripes“.

n	„snake“				„stripes“			
	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}
2131	0.0786	0.000229	1.353e-06	1.144e-08	0.111	0.000324	2.01e-06	1.283e-08
4167	0.172	0.000355	2.212e-06	2.357e-08	0.193	0.000623	2.178e-06	4.491e-08
8030	0.206	0.00107	2.737e-06	5.342e-08	0.287	0.00108	7.943e-06	4.421e-08
12805	0.324	0.00286	6.105e-06	1.995e-07	0.382	0.00117	6.793e-06	7.977e-08
16184	0.397	0.00363	8.919e-06	9.394e-08	0.393	0.00163	5.446e-06	7.988e-08
24892	0.513	0.00753	1.32e-05	1.174e-07	0.478	0.00176	7.613e-06	1.211e-07
32302	0.587	0.00647	3.129e-05	1.774e-07	0.497	0.00231	9.161e-06	1.088e-07

(в) Пренареждания по методите на вложените сечение и рекурсивната бисекция.

n	Вложени сечения				Рекурсивна бисекция			
	$R_{relative}$ 10^{-2}	for STRUMPACK 10^{-4}	with r_{tol} 10^{-6}	10^{-8}	$R_{relative}$ 10^{-2}	for STRUMPACK 10^{-4}	with r_{tol} 10^{-6}	10^{-8}
2131	0.145	0.000403	2.837e-06	3.022e-08	0.0892	0.000338	2.694e-06	2.373e-08
4167	0.247	0.00138	2.866e-06	6.963e-08	0.223	0.000987	3.28e-06	5.245e-08
8030	0.378	0.00221	6.636e-06	7.277e-08	0.373	0.00172	6.402e-06	7.322e-08
12805	0.469	0.00342	8.211e-06	1.159e-07	0.424	0.000929	7.06e-05	1.043e-07
16184	0.499	0.00268	8.668e-06	1.542e-07	0.487	0.00217	8.736e-06	1.745e-07
24892	0.583	0.00318	1.04e-05	1.536e-07	0.536	0.0033	1.738e-05	2.e-07
32302	0.615	0.00539	1.432e-05	2.159e-07	0.612	0.0031	1.419e-05	3.386e-07

та на STRUMPACK. Въпреки това паралелното бързодействие на HSS компресията отстъпва на прекия солвър използваващ блочна LU факторизация.

Едно от предимствата на HSS компресия е, че при решаване на серия от системи линейни алгебрични уравнения, при които матрицата не се изменя по-ниската изчислителна сложност на решаване с факторизирана матрица след HSS компресия и ULV-подобна факторизация $O(nr)$ ще има предимство пред решаването с факторизирана матрица след LU факторизация – $O(n^2)$. Такъв например е случаят на параболична задача, при дискретизацията на която се използва матрица на масата с диагонална концентрация (lumped mass matrix) разгледана в следващата глава.

Глава 4 Метод на крайните елементи за решаване на двумерна параболична задача за дробна дифузия

Водещ интерес за изследванията в тази глава представлява решаването на системи с факторизираната матрица при прилагане на йерархичния метод. Тази стъпка, след

HSS компресия и ULV-подобна факторизация, има изчислителната сложност $O(nr)$ [5]. За сравнение при използването на LU факторизация (гаусова елиминация) решаването на системи с факторизираната матрица изисква $O(n^2)$ аритметични операции. При стационарната задача тази стъпка се изпълнява само веднъж и почти не влияе на бързодействието на солвърите. Това се променя, когато численият метод включва решаване на последователност от системи линейни алгебрични уравнения с една и съща матрица. В този случай относителната тежест на времето за решаване на системи с факторизираната матрица се увеличава съществено. Такъв е разглежданият в настоящата глава метод за решаване на параболична задача с *дробна* дифузия.

За генериране на матрицата на масата е разработен алгоритъм и софтуерен модул, който използва информацията за геометрията на триангулацията $\mathcal{T} \in \Omega$. За дискретизация по времето ще използваме неявна диференчна схема на Ойлер с постоянна стъпка и диагонална концентрация на матрицата на масата (lumped mass matrix).

За числените експерименти в настоящата глава използваме аналог на тестовия пример от статията на Вабишчевич [26]. Това дава възможност да направим сравнение на числените резултати, съответстващи на двете различни дефиниции на дробния лапласиан.

4.1 Постановка на задачата

Използваме интегралното представяне (9) на дробният лапласиан. Разглеждаме следната параболична задача за неизвестната функция $u(x, t)$, $(x, t) \in \Omega \times [0, T]$

$$\left\{ \begin{array}{ll} \frac{\partial u(x, t)}{\partial t} + (-\Delta)^\alpha u(x, t) = f(x, t), & x \in \Omega, \quad t \in [0, T], \\ u(x, t) = 0, & x \in \Omega^c, \quad t \in [0, T], \\ u(x, 0) = u^0(x), & x \in \Omega. \end{array} \right.$$

Тук $[0, T]$ е времевият интервал. Хомогенните гранични условия на Дирихле се налагат както при стационарната задача разгледана в предходната глава.

Използваме същата допустима триангулация $\mathcal{T} \in \Omega$ като в предходната глава. Така получаваме задачата на Коши

$$M_L \frac{d\mathbf{u}}{dt} + K\mathbf{u} = M_L \mathbf{f}, \quad 0 < t \leq T, \quad \mathbf{u}(0) = \mathbf{u}^0,$$

за неизвестните функции $\mathbf{u} = (u_j(t)) \in \mathbb{R}^N$, $t \in [0, T]$ и дясна част $\mathbf{f} = (f_j(t)) \in \mathbb{R}^N$. Тук $K = K_{ij} \in \mathbb{R}^{N \times N}$ е матрицата на коравина, съответстваща на *дробния* лапласиан. Тя има вида дефиниран в уравнения (12) и (14). С $M_L = \text{diag}(m_L^i) \in \mathbb{R}^N$ е означена матрицата на масата с диагонална концентрация (lumped mass matrix), където m_L^i е концентрираната маса във възела x_i . Алгоритъмът и програмният модул за изчисляване на матрицата M_L са представени в Приложение Б.

За дискретизация по времето използваме неявния метод на Ойлер, който в общия случай има вида

$$M_L \frac{\mathbf{u}^{j+1} - \mathbf{u}^j}{\tau_j} + K\mathbf{u}^{j+1} = M_L \frac{\mathbf{f}^{j+1} + \mathbf{f}^j}{2}, \quad j = 0, \dots, m-1, \quad (15)$$

където m е броят на стъпките по времето,

$$\sum_{j=0}^{m-1} \tau_j = T,$$

$t_0 = 0$, $t_{j+1} = t_j + \tau_j$ и $\mathbf{u}^j = \mathbf{u}(t_j)$, $\mathbf{f}^j = \mathbf{f}(t_j)$.

В настоящата дисертация ще се ограничим до случая на постоянна стъпка по времето $\tau_j = \tau$. При това условие реализацията на всяка стъпка по времето на (15) се свежда до решаване на системата линейни алгебрични уравнения

$$\tilde{K} \mathbf{u}^{j+1} = \tilde{\mathbf{f}}^j, \quad (16)$$

където

$$\tilde{K} = \frac{M_L}{\tau} + K, \quad \tilde{\mathbf{f}}^j = M_L \left(\frac{\mathbf{f}^{j+1} + \mathbf{f}^j}{2} + \frac{\mathbf{u}^j}{\tau} \right).$$

Матрицата на коравина и матрицата на масата са симетрични и положително определени, и следователно (16) има единствено решение. Матрицата $\tilde{K}$ не зависи от j , т.е. тя е една и съща за всички стъпки по времето. Това означава, че при реализацията на метода на Ойлер факторизацията се изпълнява един път, след което решаваме m системи с факторизираната матрица.

Изчислителната сложност на LU факторизацията е $O(n^3)$, като след това реализацията m на брой стъпки по времето изисква още $O(n^2 m)$ аритметични операции.

При йерархичния метод – HSS компресията и ULV-подобната факторизация имат обща изчислителна сложност $O(n^2 r)$, като след това за решаване на системите с факторизираната матрица са необходими още $O(nrm)$ аритметични операции. Така, определящ за ефективността на йерархичния метод е извъндиагоналният ранг r .

В Глава 3 бяха предложени и анализирани няколко метода за пренареждане на неизвестните за стационарната задача с дробна дифузия. Така тук ще се ограничим до анализ на числени резултати при най-ефективното пренареждане – рекурсивната бисекция.

За числените експерименти ще използваме аналог на параболичната задача от статията на Вабишчевич [26], в която се използва спектралната дефиниция на *дробния* лапласиан. Задачата се решава за $(x, t) \in \Omega \times [0, T] = (-1, 1)^2 \times (0, 0.1)$. Решението се определя от независеща от времето дясна част

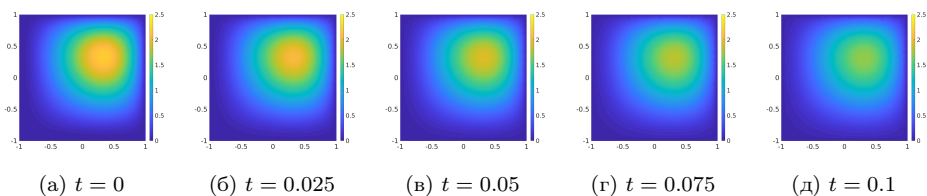
$$f(x) = \frac{(x_1 + 1)(x_2 + 1)}{4},$$

и начално условие

$$u^0(x) = 100 \left(\frac{x_1 + 1}{2} \right)^2 \left(1 - \frac{x_1 + 1}{2} \right) \left(\frac{x_2 + 1}{2} \right)^2 \left(1 - \frac{x_2 + 1}{2} \right)$$

За дискретизация по пространството са използвани триангулациите от Глава 3, при стъпка по времето $\tau = T/m$, $m = 256$.

На Фигура 15 са показани началното условие $u^0(x)$ и числените решения при $\alpha = 0.5$ за $t \in \{0.025, 0.05, 0.075, 0.1\}$. Получените резултати са качествено подобни на резултатите, представени в [26].



Фигура 15: Числени решения на моделната параболична задача с дробна дифузия със степен $\alpha = 0.5$: МКЕ по пространството и неявен метод на Ойлер по времето.

4.2 Анализ на числени експерименти върху компютърни системи с обща памет

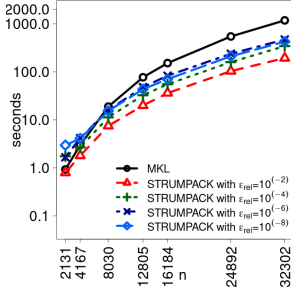
Анализираните в тази глава числени резултати са получени върху компютърни системи с обща памет. Както и в предходните глави, този тип експерименти са проведени на суперкомпютър AVITONOL.

За генериране на матрицата на коравина K и дясната част $\mathbf{f}$ е използвана програма на MatLab, която е публикувана в [2]. За пренареждане на неизвестните по метода на рекурсивната бисекция е използван алгоритъмът описан в Раздел 3.3.5 и разработеният код, представен в Приложение А.4. За изчисляване на матрицата на масата с диагонална концентрация M_L е реализиран алгоритъмът представен в Приложение Б.

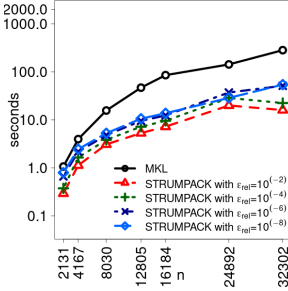
4.3 Последователни и паралелни експерименти.

На Фигура 16 са представени: времената за изпълнение на йерархичната полусепарабелна компресия и ULV-подобна факторизация при използване на софтуерния пакет STRUMPACK и на LU факторизацията при използване на MKL – (Фигури 16а и 16г); общото време за решаване на системите с факторизираните матрици на всяка от стъпките по времето в метода на Ойлер – (Фигури 16б и 16д); и цялото време за решаване на задачата – (Фигури 16в и 16е). В почти всички случаи резултатите от числените експерименти показват по-добрата ефективност на йерархичния метод, като тенденцията е тя да се повишава с увеличаване на броя на неизвестните n . За най-голямата система ($n = 32\,302$) времената за решаване на цялата задача с прилагане на HSS компресия от пакета STRUMPACK са между ~ 2.5 и ~ 5 пъти по-добри отколкото при използване на LU факторизация от пакета MKL. Тези резултати потвърждават теоретичните очаквания за по-силно подобряване на ефективността на йерархичния солвър при параболичната задача, в сравнение със стационарната задача, разгледана в предходната глава.

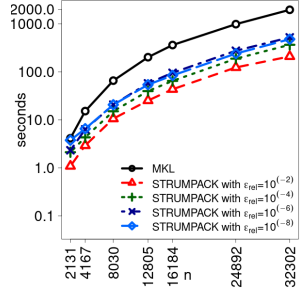
На Фигура 17 са показани паралелните ускорения при решаване на параболичната задача с прилагане на солвъра от пакета STRUMPACK, използващ HSS компресия. При паралелните експерименти с 16 нишки получаваме ускорения от ~ 3 (за $n = 2\,131$) до ~ 8 (за $n = 32\,302$). По-ниското паралелно ускорение на йерархичния метод може да се обясни с по-сложната йерархична и рекурсивна структура на алгоритъма за компресия. Паралелната реализация на решаването на системи с HSS факторизираната



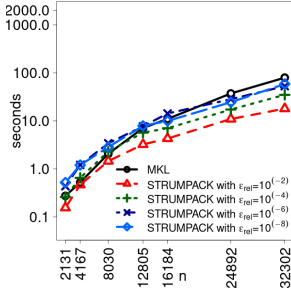
(a) Последователни времена: компресия и факторизация



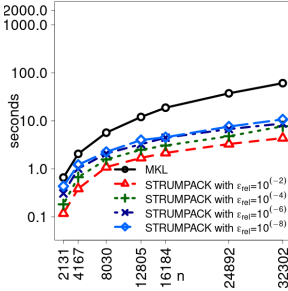
(б) Последователни времена: стъпки по времето



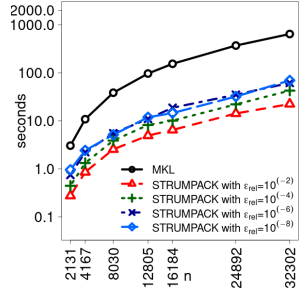
(в) Последователни времена за цялата задача



(г) Паралелни времена при 16 нишки: компресия и факторизация



(д) Паралелни времена при 16 нишки: стъпки по времето



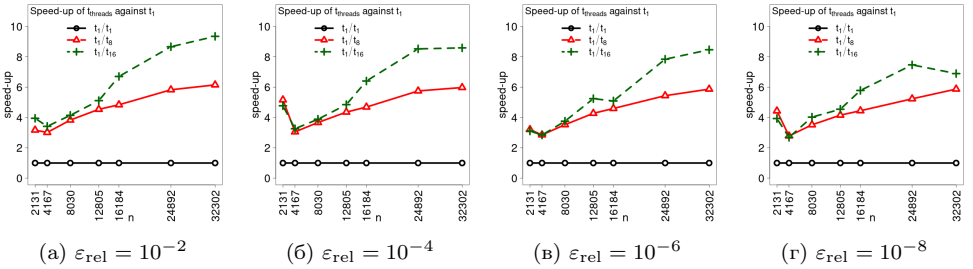
(е) Паралелни времена при 16 нишки за цялата задача

Фигура 16: Сравнение между времената за решаване на параболичната задача (16) при използване на MKL и STRUMPACK при праг на относителната грешка $\varepsilon_{\text{rel}} \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$.

матрица също има по-сложна (и по-малко балансирана) структура от блочната LU факторизация.

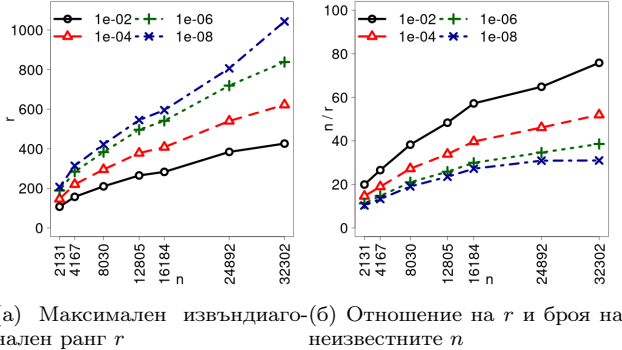
4.4 Извъндиагонален ранг

Изчисленият в процеса на HSS компресията максимален извъндиагонален ранг r е представен на Фигура 18a, като на Фигура 18б е показано отношението n/r . Рангът е мярка за ефективността на компресията, като също така е определящ в оценките за изчислителната сложност. За разглежданата задача r има значително по-малки стойности от n . Компресията е най-силна, т.е. отношението r/n е най-малко, при най-голяма стойност на прага на относителна грешка. Така, при $\varepsilon_{\text{rel}} = 10^{-2}$ рангът r е между ~ 20 и ~ 80 пъти по-малък от n , докато при най-финия праг $\varepsilon_{\text{rel}} = 10^{-8}$ това отношение е между ~ 10 и ~ 30 . Този анализ показва, че пренаредената с рекурсивна бисекция матрица $\tilde{K}$ има подходяща структура за прилагане на HSS компресия. Това



Фигура 17: Паралелни ускорения при решаване на параболичната задача с прилагане на HSS базирания солвър за $m = 256$ стъпки по времето.

се потвърждава от числените експерименти, показващи предимство на йерархичния метод в сравнение с гаусовата елиминация (блочната LU факторизация). Тук е важно да припомним, че компресията е *приблизителна*. Така, по-високата ефективност на компресията, която получаваме при по-големи стойности на прага ε_{rel} е за сметка на по-малка точност на решението.



Фигура 18: Визуализация на максималния извъндиагонал ранг r и отношение n/r .

4.5 Анализ на грешката за HSS-базирания солвър

Матрицата H получена след HSS компресия е приближение на $\tilde{K}$. Както при стационарната задача разгледана в Глава 3, ще анализираме относителната грешка R_{relative} (7).

Получените относителни грешки R_{relative} за избраните 4 стойности на времето t са представени в Таблица 5. Както и при стационарната задача, относителната грешка е близка по стойност до зададения относителен праг ε_{rel} . Представените числени резултати показват също така, че относителната грешка не нараства съществено с уве-

личаване на времевия интервал. Това потвърждава устойчивостта на неявния метод на Ойлер.

Таблица 5: Относителна грешка на HSS базираният солвър.

(а) $t = 0.025$ и $t = 0.05$

n	Относителна грешка в $t = 0.025$				Относителна грешка в $t = 0.05$			
	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}
2131	0.00385	7.52e-05	2.78e-07	5.53e-08	0.00659	0.000124	4.62e-07	9.72e-08
4167	0.006	0.00013	6.4e-07	8.62e-08	0.01	0.000225	1.03e-06	1.36e-07
8030	0.0083	0.00023	1.28e-06	2.17e-07	0.0146	0.000375	2.e-06	3.49e-07
12805	0.0106	0.00028	1.68e-06	4.77e-07	0.0178	0.000484	2.49e-06	7.56e-07
16184	0.0126	0.0003	1.75e-06	5.16e-07	0.0227	0.00052	2.77e-06	7.88e-07
24892	0.0192	0.000393	2.5e-06	9.69e-07	0.0349	0.000619	3.97e-06	1.49e-06
32302	0.0234	0.000345	2.48e-06	1.18e-06	0.0437	0.000537	3.97e-06	1.76e-06

(б) $t = 0.075$ и $t = 0.1$

n	Относителна грешка в $t = 0.075$				Относителна грешка в $t = 0.1$			
	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}	10^{-2}	$R_{relative}$ 10^{-4}	за r_{tol} 10^{-6}	10^{-8}
2131	0.0088	0.000159	5.98e-07	1.32e-07	0.0108	0.000188	7.07e-07	1.62e-07
4167	0.0135	0.000298	1.31e-06	1.707-07	0.0166	0.000361	1.55e-06	1.98e-07
8030	0.0206	0.000498	2.52e-06	4.55e-07	0.0264	0.000606	2.93e-06	5.49e-07
12805	0.0242	0.000653	3.09e-06	9.74e-07	0.0301	0.0008	3.6e-06	1.16e-06
16184	0.0324	0.000706	3.6e-06	1.01e-06	0.0419	0.000875	4.34e-06	1.2e-06
24892	0.0499	0.000807	5.19e-06	1.92e-06	0.0646	0.000973	6.26e-06	2.3e-06
32302	0.0636	0.000693	5.25e-06	2.21e-06	0.0832	0.000826	6.41e-06	2.58e-06

4.6 Заключениени бележки

Водеща тема в представените резултати е анализът на изчислителната ефективност на метода базиран на йерархична полусепарабелна компресия и ULV-подобна факторизация и неговата паралелна реализация в софтуерния пакет STRUMPACK. Специфика на приложения неявния метод на Ойлер с постоянна стъпка τ е, че численото решаване на параболната задача се свежда до $m = T/\tau$ системи линейни алгебрични уравнения с една и съща матрица $\tilde{K}$ и променящи се на всяка стъпка по времето десни части. Това означава, че матрицата $\tilde{K}$ се факторизира еднократно и акцентът пада върху решаването на m системи с известна факторизирана матрица. Ще припомним, че тази стъпка има изчислителна сложност $O(n^2)$ за гаусовия солвър и $O(nr)$ за йерархичния солвър. Представеният анализ показва, че за разглежданата задача рангът r е съществено по-малък от броя на неизвестните n , което обуславя преимуществото на йерархичния метод. В резултат на това, както последователните, така и паралелните времена за решаване на параболната задача с използване на солвъра от софтуерния пакет STRUMPACK са съществено по-добри в сравнение с времената при използване на блочна LU факторизация от пакета MKL.

Относителната грешка $R_{relative}$ е близка до зададения относителен праг, като нараства устойчиво с развитие на процеса по времето. Това потвърждава, че йерархичният метод осигурява добра точност на решението при подходящо избран праг ε_{rel} . Представените резултати показват съществено преимущество на алгоритъма използващ

йерархична полусепарабелна компресия и неговата реализация в паралелния пакет STRUMPACK.

Заклучение

В дисертацията е анализирана изчислителната ефективност на блочни числени методи и алгоритми за решаване на системи линейни алгебрични уравнения с плътни матрици. Мотивация на това изследване са приложения свързани с числено решаване на елиптични и параболични частни диференциални уравнения. Две такива задачи са използвани в представения сравнителен анализ: а) гранична задача описваща обтичане на крилни профили на Жуковски, дискретизирана с метод на граничните елементи; б) аномална дифузия в ограничена област моделирана с *дробен* лапласиан, където за дискретизация е приложен метод на крайните елементи. И в двата случая непрекъснатите задачи се свеждат до системи линейни алгебрични уравнения с плътни матрици. Показано е, че *структурата* на тези матрици е подходяща за прилагане на йерархичен метод използващ HSS компресия.

Важна част от Глава 2 е сравнителният анализ на изчислителната ефективност на софтуерни пакета имплементиращи блочна LU факторизация, като вариант на гаусова елиминация. Общият извод е, че софтуерният пакет MKL има по-добро бързодействие от анализирания алтернативни паралелни реализации на блочна LU факторизация.

Основен акцент в дисертацията е анализът на възможността за подобряване на изчислителната ефективност на решаването на системи линейни алгебрични уравнения с плътни матрици с помощта на блочен йерархичен метод използващ HSS компресия. Този метод е реализиран в софтуерния пакет със свободен достъп STRUMPACK. В Глави 2 и 3 е анализирана производителността на йерархичния алгоритъм за системи линейни алгебрични уравнения, получени при прилагане съответно на метод на граничните елементи и метод на крайните елементи за разглежданите елиптични гранични задачи. Анализът показва, че тези плътни матрици имат подходяща *структура* за прилагане на йерархичния метод. Това означава, че в процеса на HSS се получават извъндиагонални блокове с нисък ранг.

Последователните експерименти потвърждават оценките на изчислителната сложност на анализирания блочни методи. И за двете задачи йерархичният солвър има по-добро бързодействие от гаусовия солвър от пакета MKL – най-ефективният от анализирания в настоящата работа софтуерни средства използващи LU факторизация.

При прилагане на йерархичния метод се намира приближено решение на системата, като неговата точност зависи от това каква е точността на HSS компресията. На базата на проведените числени експерименти е анализирана относителната грешка R_{relative} , като решението получено с прекия гаусов солвър се приема за референтно. За задачата с дробна дифузия се наблюдават относителни грешки близки до зададения праг ε_{rel} . Това може да бъде прието като добра характеристика на алгоритъма за HSS компресия. За задачата за обтичане на крилни профили на Жуковски относителните грешки за съответните прагове ε_{rel} са по-големи, което обаче се компенсира от по-доброто бързодействие.

Известно е, че качеството на йерархичната полусепарабелна компресия силно зависи от *структурата* на плътната матрица. При двумерната дробно дифузионна задача

структурата на получената плътна матрица не е подходяща за HSS компресия. За нейното подобряване са предложени и пет метода за пренареждане. Представеният анализ показва преимущества на методите на вложените сечения и на рекурсивната бисекция.

В Глава 4 е изследвана изчислителната ефективност и точност на йерархичния солвър базиран на HSS компресия за параболична задача с *дробна* дифузия по пространството. За дискретизация по времето е използвана неявна диференчна схема на Ойлер с постоянна стъпка. При тази постановка на задачата намирането на численото решение се свежда до решаване на последователност от системи линейни алгебрични уравнения с една и съща матрица. Така на всяка стъпка по времето се решава система с матрицата на прехода, която е факторизирана еднократно. Решаването на такива системи с помощта на HSS компресия има по-ниска изчислителна сложност – $O(nr)$, в сравнение с използваната в метода на Гаус LU факторизация – $O(n^2)$. За разглежданата параболична задача времената на йерархичния солвър са по-добри, както при последователните, така и при паралелните експерименти. В същото време, благодарение на безусловната устойчивост на неявния метод на Ойлер, относителната грешка на решението се запазва близка до зададения относителен праг ε_{rel} .

Списък на публикациите по дисертацията

Основна част от представените в дисертацията резултати са публикувани в 5 излезли от печат и 2 приети за печат статии, както следва:

- D. Slavchev and S. Margenov. *Performance Analysis of Intel Xeon Phi MICs and Intel Xeon CPUs for Solving Dense Systems of Linear Algebraic Equations: Case Study of Boundary Element Method for Flow Around Airfoils*, pages 369–381. Springer International Publishing, Cham, 2019. ISBN 978-3-319-97277-0. doi: 10.1007/978-3-319-97277-0_30
- D. Slavchev and S. Margenov. *Analysis of Hierarchical Compression Parallel Solver for BEM Problems on Intel Xeon CPUs*. In G. Nikolov, N. Kolkovska, and K. Georgiev, editors, *Numerical Methods and Applications*, pages 466–473, Cham, 2019. Springer International Publishing. ISBN 978-3-030-10692-8
- D. Slavchev and S. Margenov. *Performance analysis of a parallel hierarchical semi-separable compression solver in shared and distributed memory environment for BEM discretization of flow around airfoils*. Springer International Publishing, Cham, in press
- D. Slavchev. *On the impact of reordering in a hierarchical semi-separable compression solver for fractional diffusion problems*. In I. Lirkov and S. Margenov, editors, *Large-Scale Scientific Computing*, pages 373–381, Cham, 2020. Springer International Publishing. ISBN 978-3-030-41032-2
- D. Slavchev. *Performance Analysis of Hierarchical Semi-separable Compression Solver for Fractional Diffusion Problems*. In I. Georgiev, H. Kostadinov, and E. Lilkova, editors, *Advanced Computing in Industrial Mathematics*, pages 333–344, Cham, 2021. Springer International Publishing. ISBN 978-3-030-71616-5

- D. Slavchev, S. Margenov, and I. G. Georgiev. *On the application of recursive bisection and nested dissection reorderings for solving fractional diffusion problems using HSS compression*. AIP Conference Proceedings, 2302 (1):120008, 2020. doi: 10.1063/5.0034506

Апробация на резултатите

Резултатите представени в дисертацията са докладвани в редица международни конференции и уъркшопи.

Международни конференции:

- Large-Scale Scientific Computations (LSSC), Созопол, 2017, 2019, 2021
- Annual Meeting of the Bulgarian Section of SIAM (BGSIAM), София, 2017, 2018, 2019
- Numerical Methods for Scientific Computations and Advanced Applications (NMSCAA), Хисаря, 2018
- Numerical Methods and Applications (NM&A), Боровец, 2018
- Twelfth On-Line Conference of the Euro-American Consortium for Promoting the Application of Mathematics in Technical and Natural Sciences (AMiTaNS), Албена, 2020

Уъркшопи:

- Две години Авитохол: Иновативни Суперкомпютърни Приложения, Панагюрище, 2017
- Numerical Solution of Fractional Differential Equations and Applications (NSFDE&A), Созопол, 2020

Основни научни и научно-приложни приноси

1. Изследвана е производителността на следните софтуерни пакети за решаване на линейни системи с плътни матрици, с помощта на блочна LU факторизация: за процесори с общо предназначение (CPU) – пакетът на Intel Math Kernel Library (MKL) и библиотеката със свободен достъп Parallel Linear Algebra Software for Multicore Architectures (PLASMA); за ускорители с архитектурата Many Integrated Core (MIC) на Intel – MKL и пакетът със свободен достъп Matrix Algebra on GPU and Multicore Architectures (MAGMA). Резултатите от числените експерименти за системи получени при дискретизация с метод на граничните елементи за гранична задача за ламинарен поток около крилни профили на Жуковски са в съответствие с асимптотичните оценки на изчислителната сложност. Сравнителният анализ показва по-добро бързодействие и много добра паралелна скалируемост на пакета MKL.

2. Изследвана е изчислителната сложност, паралелната ефективност и относителната грешка на метод на йерархична полусепарабелна компресия (HSS). Числените експерименти са проведени с пакета със свободен достъп STRUctured Matrices PACKage (STRUMPACK), в който е реализиран паралелен солвър на базата HSS компресия и ULV-подобна факторизация. Сравнителният анализ включва два типа плътни матрици, които са получени при дискретизация с: а) метод на граничните елементи на гранична задача за ламинарен поток около крилни профили на Жуковски; б) метод на крайните елементи за двумерна дробно дифузионна гранична задача. Направен е сравнителен анализ на солвърите от пакетите MKL и STRUMPACK. Получена е характеристика в зависимост от прага на относителна грешка при HSS компресията на случаите, в които йерархичният метод има по-добро бързодействие.
3. Показано е, че за задачата за обтичане на профили на Жуковски при дискретизация по метода на граничните елементи, последователната номерация на възлите по границата на профилите води до матрица с подходяща за HSS компресия *структура*. Това не е така за дробно дифузионната гранична задача, дискретизирана с метод на крайните елементи. С цел подобряване на ефективността на йерархичната полусепарабелна компресия са предложени и изследвани пет метода за пренареждане на неизвестните. За три от тях са разработени собствени алгоритми и програмни реализации. Сравнителният анализ показва съществено подобрение на резултатите при прилагане методите на вложените сечения и на рекурсивната бисекция.
4. Разработен е метод, алгоритъм и програмна реализация за числено решаване на параболично уравнение с дробно дифузионен оператор по пространството. За дискретизация по времето е приложен неявен метод на Ойлер с постоянна стъпка по времето и диагонална концентрация на масата. Доказано е, че за тази нестационарна задача, изчислителната сложност на отделните части на алгоритъма създава условия за предимство на йерархичния метод на базата на HSS компресия. Това е потвърдено от проведените числени експерименти. Така за всички размерности на дискретната задача по пространството, както и при всички варианти на праг на относителна грешка, вариантът на програмата използваща солвъра от пакета STRUMPACK има по-добро бързодействие от този използващ MKL.

Библиография

- [1] G. Acosta and J.P. Borthagaray. A Fractional Laplace Equation: Regularity of Solutions and Finite Element Approximations. In *SIAM Journal on Numerical Analysis*, volume 55, pages 472–495, 2017. doi: 10.1137/15M1033952.
- [2] G. Acosta, F.M. Bersetche, and J.P. Borthagaray. A short FE implementation for a 2d homogeneous Dirichlet problem of a fractional Laplacian. In *Computers & Mathematics with Applications*, volume 74, pages 784 – 816, 2017. doi: <https://doi.org/10.1016/j.camwa.2017.05.026>.

- [3] M. Benzi, D. Bini, D. Kressner, H. Munthe-Kaas, and C. Van Loan. *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications*. Springer, Cham, Cetraro, Italy, 2015. ISBN 978-3-319-49887-4. doi: 10.1007/978-3-319-49887-4.
- [4] K. Binder, Ch. Bennemann, J. Baschnagel, and W. Paul. Anomalous diffusion of polymers in supercooled melts near the glass transition. In A. Pękałski and K. Sznajd-Weron, editors, *Anomalous Diffusion From Basics to Applications*, pages 124–139, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-49259-7.
- [5] S. Chandrasekaran, M. Gu, and T. Pals. A fast *ulv* decomposition solver for hierarchically semiseparable representations. In *SIAM J. Matrix Anal. Appl.*, volume 28, page 603–622, USA, August 2006. Society for Industrial and Applied Mathematics. doi: 10.1137/S0895479803436652.
- [6] S. Chaturapruek, J. Breslau, D. Yazdi, T. Kolokolnikov, and S.G. McCalla. Crime modeling with Lévy flights. In *SIAM Journal on Applied Mathematics*, volume 73, pages 1703–1720. Society for Industrial and Applied Mathematics, 2021/12/06/ 2013. URL <http://www.jstor.org/stable/24510700>.
- [7] J. Dongarra, M. Abalenkovs, A. Abdelfattah, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, I. Yamazaki, and A. YarKhan. Parallel programming models for dense linear algebra on heterogeneous systems. In *Supercomputing Frontiers and Innovations*, volume 2, 2016. URL <http://superfri.org/superfri/article/view/90>.
- [8] P. Ghysels, X.S. Li, F.-H. Rouet, S. Williams, and A. Napov. An Efficient Multicore Implementation of a Novel HSS-Structured Multifrontal Solver Using Randomized Sampling. In *SIAM Journal on Scientific Computing*, volume 38, pages S358–S384, 2016. doi: 10.1137/15M1010117.
- [9] C. Gorman, G. Chavez, P. Ghysels, T. Mary, F.-H. Rouet, and X. Li. Matrix-free construction of hss representation using adaptive randomized sampling. In *ArXiv*, volume abs/1810.04125, 2018.
- [10] W. Hackbusch. A Sparse Matrix Arithmetic Based on $\mathcal{H}$ -Matrices. Part I: Introduction to $\mathcal{H}$ -Matrices. In *Computing*, volume 62, pages 89–108, Apr 1999. doi: 10.1007/s006070050015.
- [11] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$ -matrices. In H.-J. Bungartz, R.H.W. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–29, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-642-59709-1. doi: 10.1007/978-3-642-59709-1_2.
- [12] S. Harizanov, R. Lazarov, and S. Margenov. A survey on numerical methods for spectral space-fractional diffusion problems. In *Fractional Calculus and Applied Analysis*, volume 23, pages 1605–1646, 2020. doi: doi:10.1515/fca-2020-0080.

- [13] S. Harizanov, S. Margenov, and N. Popivanov. Spectral Fractional Laplacian with Inhomogeneous Dirichlet Data: Questions, Problems, Solutions. In I. Georgiev, H. Kostadinov, and E. Lilkova, editors, *Advanced Computing in Industrial Mathematics*, pages 123–138, Cham, 2021. Springer International Publishing. ISBN 978-3-030-71616-5.
- [14] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 1991. ISBN 0-534-13014-3.
- [15] T.A.M. Langlands, B.I. Henry, and S.L. Wearne. Fractional Cable Equation Models for Anomalous Electrodiffusion in Nerve Cells: Finite Domain Solutions. In *SIAM Journal on Applied Mathematics*, volume 71, pages 1168–1203, 2011. doi: 10.1137/090775920.
- [16] P.G. Martinsson. A Fast Randomized Algorithm for Computing a Hierarchically Semiseparable Representation of a Matrix. In *SIAM Journal on Matrix Analysis and Applications*, volume 32, pages 1251–1274, 2011. doi: 10.1137/100786617.
- [17] R. Musina and A.I. Nazarov. On fractional laplacians. In *Communications in Partial Differential Equations*, volume 39, pages 1780–1790. Taylor & Francis, 2014. doi: 10.1080/03605302.2013.864304.
- [18] V. Pasheva and R. Lazarov. Boundary element method for 2D problems of ideal fluid flows with free boundaries. In *Advances in Water Resources*, volume 12, pages 37–45, 1989. doi: 10.1016/0309-1708(89)90014-6.
- [19] E. Rebrova, G. Chávez, Y. Liu, P. Ghysels, and X.S. Li. A Study of Clustering Techniques and Hierarchical Matrix Formats for Kernel Ridge Regression. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 883–892, 2018. doi: 10.1109/IPDPSW.2018.00140.
- [20] L. Rosasco, M. Belkin, and E. De Vito. On learning with integral operators. In *Journal of Machine Learning Research*, volume 11, pages 905–934, 2010. URL <http://jmlr.org/papers/v11/rosasco10a.html>.
- [21] F.-H. Rouet, X.S. Li, P. Ghysels, and A. Napov. A Distributed-Memory Package for Dense Hierarchically Semi-Separable Matrix Computations Using Randomization. In *ACM Trans. Math. Softw.*, volume 42, pages 27:1–27:35, New York, NY, USA, June 2016. ACM. doi: 10.1145/2930660.
- [22] H.D. Simon and S.-H. Teng. How Good is Recursive Bisection? In *SIAM Journal on Scientific Computing*, volume 18, pages 1436–1445, 1997. doi: 10.1137/S1064827593255135.
- [23] D. Slavchev. Parallelization of boundary element method for laplasian equation. Master’s thesis, Technical University - Sofia, 2014.
- [24] D. Slavchev and S. Margenov. Performance analysis of a parallel hierarchical semi-separable compression solver in shared and distributed memory environment for bem discretization of flow around airfoils. In *Advanced Computing in Industrial Mathematics*, Cham, in press. Springer International Publishing.

- [25] H. Taitelbaum. Diagnosis using photon diffusion: From brain oxygenation to the fat of the atlantic salmon. In A. Pękalski and K. Sznajd-Weron, editors, *Anomalous Diffusion From Basics to Applications*, pages 160–174, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-49259-7.
- [26] P.N. Vabishchevich. Splitting schemes for non-stationary problems with a rational approximation for fractional powers of the operator. In *Applied Numerical Mathematics*, volume 165, pages 414–430, 2021. doi: <https://doi.org/10.1016/j.apnum.2021.03.006>.
- [27] R.C. Whaley. Atlas (automatically tuned linear algebra software). In D. Padua, editor, *Encyclopedia of Parallel Computing*, pages 95–101, Boston, MA, 2011. Springer US. ISBN 978-0-387-09766-4. doi: 10.1007/978-0-387-09766-4_85.
- [28] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. In *Numerical Lin. Alg. with Applic.*, volume 17, pages 953–976, 2010. doi: 10.1002/nla.691.
- [29] С. Маргенов. *Числени методи за системи с разредени матрици*. Институт по паралелна обработка на информацията, Българска академия на науките, 2007. ISBN 9780198520115.