Дисертация

Велизар Павлов

1998 г

РУСЕНСКИ УНИВЕРСИТЕТ "АНГЕЛ КЪНЧЕВ"

КАТЕДРА "ЧИСЛЕНИ МЕТОДИ И СТАТИСТИКА"

Велизар Тодоров Павлов

ИЗСЛЕДВАНЕ НА ЧИСЛЕНАТА УСТОЙЧИВОСТ НА НЯКОИ ПАРАЛЕЛНИ АЛГОРИТМИ ЗА РЕШАВАНЕ НА ЛЕНТОВИ СИСТЕМИ ЛИНЕЙНИ АЛГЕБРИЧНИ УРАВНЕНИЯ

ДИСЕРТАЦИЯ

за присъждане на научна и образователна степен "доктор"

Научен ръководител

доц. д-р Пламен Ялъмов

1998

Съдържание

Фигури

Таблици

Увод

Обосновка и състояние на проблема

Едно от най-важните свойства на изчислителните алгоритми е тяхната числена устойчивост. Това свойство е от особено значение, тъй като при компютърна реализация на даден алгоритъм, поради крайното представяне на реалните числа с плаваща точка, изпълнението на аритметичните операции е свързано с грешки от закръгляване. На практика при компютърно пресмятане на резултата от операцията $x * y$, където $* \in \{+, -, \times, /\}$, съгласно основната хипотеза за реализиране на аритметичните операции с плаваща точка (виж [4]), за реалните числа x и y е валидно равенството

$$\mathrm{fl}(x * y) = (x * y)(1 + \sigma), \quad |\sigma| \leq \rho_0,$$

където ρ_0 е машинната точност. Нека да припомним (виж [27]), че $\rho_0 > 0$ е най-малкото машинно число удовлетворяващо неравенството

$$\mathrm{fl}((1 + \rho_0) - 1) > 0.$$

Изследванията в настоящата дисертация са направени в компютърна среда с двойна точност, където $\rho_0 \approx 2.22\text{E-}16$. Известно е, че грешките от закръгляване могат да доведат до решение на даден проблем, което е твърде далеч от истинското решение. Такива примери има много, дори и когато алгоритъмът съдържа относително малък брой елементарни операции. Този проблем възниква още с появата на първите компютри и е засегнат за първи път от Голдщайн и Фон Нойман в [29]. С течение на времето върху компютри се решават все по-големи и по-големи задачи, а появата на паралелни компютри и тяхното серийно производство през последните години доведе до практическото решаване на задачи с милиони операции. Този факт още повече изостря проблема с точността, тъй като при всяка елементарна операция компютърът прави грешка от закръгляване, а натрупването на тези грешки при такъв огромен брой операции може да бъде твърде опасно.

Казваме (виж [4]), че един алгоритъм е *числено устойчив* (или има *добро числено поведение*) в някакво множество от входни данни, ако грешките от закръгляване при пресмятанията са сравними с грешките от закръгляване на входните данни. Обратно, ако грешките от пресмятанията са много по-големи от грешките в данните, алгоритъмът е *числено неустойчив* (или има *лошо числено поведение*). Естествено, резултатът от изпълнението на всеки алгоритъм зависи, както от параметрите на изчислителната среда, така и от това, коя от възможните четири комбинации (добре/лошо обусловена задача и устойчив/неустойчив алгоритъм) е била осъществена.

Съществуват два основни подхода за оценка на численото поведение на даден алгоритъм. Те са известни като *прав и обратен анализ* на грешките от закръгляване (forward and backward roundoff error analysis).

Исторически първо възниква идеята за прав анализ. Такъв подход се използва отново за пръв път от Голдщайн и Фон Нойман в [29], при което основната идея е да се изследва влиянието на грешките на всеки етап от изчисленията върху резултатите от следващия етап. Така последователно се проследява разпространението на грешките от началото до края на изчислителния процес. От гледна точка на правия анализ един алгоритъм има добро числено поведение, ако грешката в крайния резултат е относително малка. Това изисква задачата да бъде добре обусловена. По този начин се смесват обусловеността на задачата и устойчивостта на алгоритъма. Практически недостатък на правия анализ на грешките е и това, че неговата реализация е възможна само за сравнително прости изчислителни алгоритми.

Ето защо в края на 50-те и началото на 60-те години Уилкинсон [65, 66, 67] разработва друг подход, който той нарича обратен анализ. При него ефектът на грешките от закръгляване при пресмятанията се разглежда като породен от някакво фиктивно смущение (наричано *еквивалентно смущение*) във входните данни. Еквивалентните смущения отразяват влиянието на грешките от закръгляване на алгоритъма и дават база за сравнение на отделните алгоритми. По такъв начин пресметнатото решение се разглежда като точно решение на еквивалентно смутена задача. От гледна точка на обратния анализ даден алгоритъм има добро числено поведение, ако еквивалентните смущения са относително малки, т.е. ако алгоритъмът при всяко негово изпълнение води до точно решение на задача, която е относително близка до първоначалната задача. При това може да се окаже, че еквивалентните смущения са от порядъка на грешките от закръгляване на входните данни при техния запис в паметта на компютъра. В такъв случай става въпрос за алгоритъм с най-доброто възможно числено поведение.

Правият и обратният анализ на грешките от закръгляване при изследване на изчислителните алгоритми не се изключват взаимно. В редица случаи се прибягва до комбинация на тези два подхода, като при това се казва, че се използва подход на *смесен (комбиниран) анализ.*

Подходите на прав, обратен и смесен анализ намират голямо приложение за изследване на численото поведение на редица алгоритми в линейната алгебра. Първите резултати в тази област принадлежат на Уилкинсон [65, 66, 67]. Използвайки най-вече обратен анализ, той извежда оценки за еквивалентните смущения във входните данни при реализацията на някои основни алгоритми. Съществено при това е, че за всеки отделен алгоритъм тези оценки са изведени като се използват особеностите на алгоритъма, което изисква известна досетливост при представянето на грешките като смущения във входните данни. По-късно идеите на Уилкинсон са продължени от Воеводин [2] и Хайам [36], в чиито работи са изследвани някои нови алгоритми на линейната алгебра, както и са подобрени някои съществуващи оценки.

Нов етап в развитието на изследванията на числената устойчивост на алгоритмите се явява общият конструктивен подход към правия, обратния и смесения анализ, предложен от Воеводин и Ялъмов в [60, 70]. Този конструктивен подход е особено полезен в случаите, когато оценките на правия и обратния анализ са по-трудни за извеждане. Същността на предложения нов подход се изразява в използване на графа на алгоритъма и неговата паралелна структура. Една от основните разлики между подхода на Уилкинсон [67] и този на Воеводин-Ялъмов [60] е, че в [60] понятието еквивалентно смущение е въведено не само за входните данни, но и за всички междинни и изходни данни. Това позволява да се изследва натрупването на грешките от закръгляване по графа на алгоритъма. В [60] натрупването на грешки от закръгляване се моделира със система алгебрични уравнения, която в общия случай е нелинейна. На практика еднократната грешка от закръгляване е толкова малка, че може да се разгледа линеаризирана система, която е достатъчно добро приближение на оригиналната нелинейна система. Различни оценки за еквивалентните смущения могат да бъдат изведени от решението на горепосочената система. Благодарение на този нов метод за оценка на грешките от закръгляване в [6] е показано, че всички по-известни оценки могат да бъдат получени конструктивно. В [70] са предложени някои разширения на метода представен в [60]. По-конкретно, за пръв път са доказани твърдения, които описват класове от алгоритми (на базата на техните графи), за които обратният анализ се извършва чрез проверка на някои прости свойства на графа на конкретния алгоритъм. За разлика от [60], където се работи с абсолютни еквивалентни смущения, в [70] се използват относителни

еквивалентни смущения, като при това са изолирани класове от алгоритми, за които тези смущения не зависят от ръста на междинните резултати (той може да бъде много опасен понякога).

Както вече стана ясно важна роля при изследване на численото поведение на даден алгоритъм играят грешките от закръгляване. По-конкретно при изследване на численото поведение на различните алгоритми за решаване на системи линейни уравнения, обратният анализ е за предпочитане, тъй като той дава възможност да бъде отделена устойчивостта на алгоритъма от обусловеността на решаваната система. Що се отнася до извежданите оценки, то те най-често се разглеждат по норма или покомпонентно (normwise and componentwise backward error). Подробно описание на тези грешки е дадено в [34]. Разбира се покомпонентните оценки са за предпочитане, тъй като те са по-точни, но за съжаление не винаги могат да се изведат такива оценки.

Различни модели на грешките от закръгляване се използват в [17, 43, 44, 45, 47, 48, 49, 50, 59]. Всички тези публикации разглеждат прав анализ. С изключение на [17], останалите автори се основават върху идеята за сравняване на влиянието на грешките от закръгляване в отделните етапи на даден алгоритъм, с влиянието на грешките от закръгляване на входните данни, което дава оценка за обратния анализ (в абсолютен или относителен смисъл). В [49] се предлага способ за представяне на локалните грешки от закръгляване, като смущения в някои от данните, използвайки графа на алгоритъма.

Множеството от различни подходи при анализа на грешките от закръгляване дава резултати до известна степен. При теоретичното изследване на отделните алгоритми има все още доста трудности, особено при появата на голям брой нови паралелни алгоритми. Съществено важна задача е на базата на получените изследвания, след като се разбере качествено натрупването на грешките от закръгляване, да се предлагат нови по-устойчиви алгоритми. Засега общата хипотеза е, че повечето паралелни алгоритми са по-неустойчиви от класическите последователни такива. Въпросът, до каква степен може да се разчита на новите паралелни алгоритми, стои отворен за много от тях. Целта на настоящата дисертация е да бъде даден отговор на този въпрос за някои известни паралелни алгоритми за решаване на лентови системи линейни алгебрични уравнения.

От своя страна задачата за решаване на лентови системи е една от най-актуалните и често срещани в различни приложения на маремагиката. В случая когато тези системи са и от високи размерности се налага необходимостта от използване на паралелни алгоритми за тяхното решаване [13, 24, 25, 31, 33, 40, 46, 55, 68]. По-конкретно, в [24, 40] са предложени

два паралелни алгоритъма за решаване на лентови (с малка ширина на лентата) системи линейни уравнения. В първата работа са разгледани системи, възникващи при дискретизация на диференциални уравнения от типа на Хелмхолц. Във втората работа са дискутирани някои особености на паралелната реализация върху различен тип архитектури на описания преди това алгоритъм. От подобно естество са и работите [13, 55], в които се решават паралелно лентови системи с произволна ширина на лентата.

В последните години, с цел да се подобрят паралелните свойства на алгоритмите, решаваните лентови системи, най-напред се структурират в блочно тридиагонален вид, след което се прилага и конкретен паралелен алгоритъм за тяхното решаване. При този подход алгоритмите, които се използват обикновено са обобщения на известни паралелни алгоритми за решаване на тридиагонални системи линенйни уравнения. Такъв подход е използван в [25, 31, 33, 46, 68].

По-конкретно, за решаване на тридиагонални системи, в литературата са известни следните паралелни методи: методи от типа на разделяне (partitioning) на системата [8, 9, 10, 19, 20, 39, 42, 64]; методи от типа на цикличната редукция [7, 11, 21, 22, 33, 37, 38, 41]; методи използващи Гаусово изключване (в оригинален вид) [57, 58, 62]. В някои от цитираните работи се разглеждат особености при паралелната реализация на алгоритмите [7, 8, 19, 20, 21, 39, 41, 42, 57, 58, 62], други са посветени върху тяхната числена устойчивост, при което са направени пълни изследвания в [11] (на метода на цикличната редукция [37]), в [69] (на метода на цикличната редукция - модификация без обратен ход [38]), в останалите работи [9, 10, 11, 14, 22, 63] са разгледани само отделни елементи от числената устойчивост на алгоритмите, без да е получен пълен анализ на численото поведение на тези алгоритми. Нека да добавим и това, че пълно изследване на класическия метод на Гаус е представено в [35] (в тридиагоналния случай) и в [36] (в лентовия случай).

Методите на разделяне на системата са характерни с това, че дават ефективни паралелните алгоритми, които са доста популярни поради простота на разделяне на изчислителната работа между отделните процесори. Основната идея е дадената система да бъде блочно разделена по подходящ начин на определен брой подсистеми, които могат да бъдат решавани паралелно, след което се решава т.нар. редуцирана система (с размерност от порядъка на броя на използваните процесори) и накрая посредством обикновено последователно заместване се намират всички компоненти на решението на изходната система (това също може да бъде направено паралелно). Типичен представител на методите на разделяне на системата е този на Уанг [64]. Важно негово свойство е, че

получената редуцирана система е отново тридиагонална (блочно тридиагонална в блочно тридиагоналния случай). Нека да отбележим какво е направено до този момент по отношение изследване на неговата числена устойчивост в тридиагоналния случай. В [10] е доказано, че ако матрицата на изходната тридиагонална система е с диагонално преобладаване по редове (стълбове), то и матрицата на редуцирана система е от същия тип. Освен това, когато матрицата на изходната система е симетрична и положително определена или M-матрица, то и редуцираната матрица е от същия тип, като тези свойства са в сила дори и за плътни матрици (виж [14, стр. 94, 209]). В [63] авторът разглежда много тесен клас от тридиагонални системи със строго диагонално преобладаване в определен смисъл и доказва, че редуцираната система е от същия вид. Така или иначе обаче, по отношение изследване на числената устойчивост на метода на Уанг за решаване на тридиагонални системи линейни уравнения има по-скоро отделни резултати и липсва пълен анализ на разпространяването на грешките от закръгляване от началото до края на изчислителния процес.

Методите на разделяне на системата се използват с успех и за решаване на лентови системи, структурирани по-подходящ начин в блочно тридиагонален вид. Такъв подход е използван в [24, 46, 68]. По-конкретно, в [24] са разгледани особеностите при паралелната реализация на два метода за решаване на лентови системи. За съжаление обаче, липсва изследване на численото поведение на реализираните алгоритми. В [46] отново се разглежда обобщен вариант на метода на Уанг в случая на решаване на лентови системи. При това е представен и критерий за устойчивост на алгоритъма, но липсва анализ на разпространението на грешките от закръгляване, както и оценка на грешката в решението на системата. В тази работа още е направено и числено сравнение с методите на Гаус и на цикличната редукция. В [68], след като е направено специално разделяне на решаваната лентова система, е използван метод на Гаус с частичен избор на главен елемент за всеки блок. При това получената редуцирана система е с променлива размерност, поради избора на главен елемент. По отношение на устойчивостта на предложения алгоритъм са изведени груби оценки, а относно реализацията е показано, че той е подходящ за реализиране върху компютри с мултипроцесорна архитектура състояща се от малък брой процесорни елементи.

В последните десетина години определен интерес представлява методът на цикличната редукция (класически вид [37] и модификация без обратен ход [38]) и неговото прилагане за решаване на тридиагонални системи линейни уравнения [7, 11, 21, 22, 33, 41, 69]. В някои от тези работи са засегнати и аспекти от числената му устойчивост. По-конкретно в

[11] посредством подхода на обратния анализ е представено пълно изследване в случай, че разглежданата система е с диагонално преобладаване, при което изведените оценки са по норма. В [21] е представена още една версия на метода на цикличната редукция в случай на симетрична матрица с постоянни коефициенти и е показано, че в този случай алгоритъмът е по-устойчив. Някои изследвания върху устойчивостта на предложената в [21] версия (в общия случай) са направени по-късно в [22]. В последната работа ръстът на елементите при реализацията на алгоритъма е ограничен и така би могло да се очаква, че правата грешка е също ограничена, но за съжаление липсва точна оценка за нея. Модифицираният вариант на цикличната редукция [38], при предположение, че разглежданата система е добре обусловена, е изследван в [69], при което изведените оценки са покомпонентни.

Засягайки метода на цикличната редукция, то следва да отбележим, че той може да бъде обобщен в блочно тридиагонален вид. В тази посока са изследванията представени в [31, 33]. В първата работа е направено сравнение по отношение на паралелната реализация между два метода, единият от които се основава на блочно циклична редукция (в класически вид), а другият използва двустранно Гаусово изключване. В [33] метода на цикличната реукция за решаване на блочно тридиагонални системи се разглежда като итерационен метод, при което е изследвана неговата сходимост.

В [62] са дискутирани някои елементи от устойчивостта на алгоритъма на Бабушка представен в [15]. Основната идея на този алгоритъм се изразява в прилагане на двустранно Гаусово изключване. Засягайки метода на Гаус, то следва да отбележим, че в [35] е направен подробен анализ на неговото числено поведение при решаване на тридиагонални системи. При това е даден отговор на въпроса - каква е "най-добрата" оценка за грешката в решението на една тридиагонална система получено по метода на Гаус. Показано е, че тази оценка зависи от обусловеността на системата и от ръста на междинните резултати. Отделени са класове от системи, за които този ръст е ограничен и следователно алгоритъмът на Гаус е числено устойчив. По-късно получените в [35] резултати са обобщени в [36] в случай на произволна лентова система.

Предмет на настоящата дисертация е пълното изследване на числената устойчивост на метода на Уанг [64] за решаване на лентови системи линейни уравнения и метода на цикличната редукция (модификация без обратен ход [38]) за решаване на блочно тридиагонални системи линейни уравнения, а също и пълно изследване на метода на Уанг във важния частен случай на тридиагонални системи. Що се отнася до изследване на споменатия метод на цикличната редукция в този важен частен случай, то

такова както отбелязахме, е направено в [69].

Често срещан проблем при решаване на системи линейни уравнения е как да се подобри числената устойчивост на конкретен алгоритъм, така че в случай на добре обусловена система да се получи и решение достатъчно близко до точното. Конкретно, при използване на метода на Уанг, с цел да се подобри числената устойчивост при паралелното решаване на подсистемите, получени при блочното разделяне на изходната система, в [9] авторите правят QR разлагане на всеки блок и използват при необходимост ново блочно разделяне, така че новите блокове да са добре обусловени. Този подход, обаче води до съществено увеличаване броя на аритметичните операции и усложнява алгоритъма. Един друг по-общ подход за подобряване на числената устойчивост на алгоритмите е подходът на изкуствено смущаване на някои данни. Такъв подход е приложен за първи път в [16] при алгоритъма за обръщане на матрици по метода на Шрасен. Същият подход е използван и в [23, 32] при решаване на Тьоплицови системи. Навсякъде при прилагане на този подход стои въпросът: "В качеството на оптимално смущение какви стойности биха могли да се използват"? Оказва се, че при различните алгоритми се предлагат различни оптимални смущения, откъдето се налага и изводът, че те зависят от конкретния алгоритъм.

Подходът със смущения е използван и в настоящата дисертация за подобряване на числената устойчивост на метода на Уанг и метода на цикличната редукция (модифициран вариант без обратен ход) в случай на решаване на тридиагонални системи линейни уравнения. Естествено при прилагане на този подход се получава и смутено решение, което и при двата гореспоменати метода е доуточнено, чрез използване на достатъчно бърза процедура за итерационно уточняване [30]. Сходимостта на използваната процедура е изследвана в [72]. Използваният в дисертацията подход със смущения напълно аналогично би могъл да бъде пренесен и за стабилизиране на алгоритмите в лентовия случай, но тъй като по отношение на резултатите няма новости, на този случай в дисертацията не е отделено специално място.

В теорията на анализа на грешките от закръгляване при компютърната реализация на алгоритмите се използват някои стандартни означения, които са използвани и в настоящата дисертация. С "шапка" са означавани скалари, вектори или матрици, изчислени с грешки от закръгляване. Например ако $\hat{T}$ е изчислена с грешки матрица, а δT е матрицата от грешки (накратко казано *правата грешка*), то е изпълнено $\hat{T} = T + \delta T$. Матрицата от еквивалентни смущения във входните данни (има се предвид матрицата T) пък е означавана с ΔT (*обратната грешка*). Освен това навсякъде в дисертацията векторните и матрични неравенства се разби-

рат в покомпонентен смисъл, при което означение от вида $|T|$ се разбира матрица от модулите на елементите на T.

Методология на дисертационната работа

В качеството на основни методи за изследване на числената устойчивост на метода на Уанг за решаване на тридиагонални и лентови системи (структурирани в блочно тридиагонален вид), а също и на метода на цикличната редукция (модификация без обратен ход) за решаване на блочно тридиагонални системи (очевидно лентовите системи могат да се разглеждат като частен случай на блочно тридиагоналните), в тази дисертация са използвани подходите на прав и обратен анализ [29, 65, 66, 67]. А за подобряване на числената устойчивост на алгоритмите на Уанг и на цикличната редукция (модификация без обратен ход) при решаване на тридиагонални системи е използван подход на изкуствено смущаване на някои данни [16].

Цел на дисертационната работа

Основните цели на настоящата дисертация са:

- Да се направи пълно изследване на числената устойчивост на метода на Уанг [64] за решаване на:
  - тридиагонални системи линейни алгебрични уравнения;
  - лентови системи линейни алгебрични уравнения.
- Да се разгледат някои специални класове от матрици, за които е известно че методът на Гаус е числено устойчив [36] и да се изследва поведението на метода на Уанг за същите класове, както за тридиагонални така и за лентови системи.
- Да се изследва числената устойчивост на метода на цикличната редукция (модифициран вариант без обратен ход [38]) за решаване на блочно тридиагонални системи, притежаващи свойството на равномерно блочно диагонално преобладаване по стълбове.
- Да се подобри числената устойчивост на изследваните методи на Уанг и цикличната редукция, в случай на произволни добре обусловени тридиагонални системи линейни уравнения.

Анотация на получените резултати

- В Глава 1 е направено пълно изследване на числената устойчивост на метода на Уанг за решаване на тридиагонални системи линейни уравнения. Както отбелязахме този метод принадлежи към така наречените методи на разделяне (partitioning) на системата. Той притежава много добри паралелни свойства и следователно е удобен за решаване на тридиагонални системи линейни уравнения, използвайки паралелни компютри.

  Главата се състои от шест раздела. В Раздел 1.1 е направено описание на изследвания алгоритъм. Разглежда се следната тридиагонална система линейни уравнения

  $$Ax = d,$$

  където

  $$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}.$$

  За простота се допуска, че $n = ks - 1$, където k е цяло положително число, а s е броя на паралелните процесори, които могат да се използват. Най-напред се прави специално разделяне на матрицата A, търсеното решение x и на дясната част d на разглежданата система:

  $$\begin{pmatrix} B_1 & \bar{c}_1 & & & & & & \\ a_k & b_k & c_k & & & & & \\ & \bar{a}_2 & B_2 & \bar{c}_2 & & & & \\ & & a_{2k} & b_{2k} & c_{2k} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \bar{a}_{s-1} & B_{s-1} & \bar{c}_{s-1} & \\ & & & & & a_{(s-1)k} & b_{(s-1)k} & c_{(s-1)k} \\ & & & & & & \bar{a}_s & B_s \end{pmatrix} \begin{pmatrix} X_1 \\ x_k \\ X_2 \\ x_{2k} \\ \vdots \\ X_{s-1} \\ x_{(s-1)k} \\ X_s \end{pmatrix} = \begin{pmatrix} D_1 \\ d_k \\ D_2 \\ d_{2k} \\ \vdots \\ D_{s-1} \\ d_{(s-1)k} \\ D_s \end{pmatrix}$$

  където $B_i \in \mathcal{R}^{(k-1)\times(k-1)}, i = 1, 2, \ldots, s$, са също тридиагонални матрици, векторите $\bar{a}_i, \bar{c}_i$ имат следния вид:

  $$\begin{aligned} \bar{a}_i &= (a_{(i-1)k+1}, 0, \ldots, 0)^T \in \mathcal{R}^{(k-1)\times 1}, \quad i = 2, \ldots, s, \\ \bar{c}_i &= (0, \ldots, 0, c_{ik-1})^T, \in \mathcal{R}^{(k-1)\times 1}, \quad i = 1, \ldots, s-1, \end{aligned}$$

  $X_i, D_i, i = 1, 2, \ldots, s$, са също вектори от размерност $k - 1$.

За удобство в изложението изследвания алгоритъм се представя в блочна форма. За тази цел най-напред се дефинира следната пермутация на числата $[1,\ldots,sk-1]$:

$$[1:k-1;\ldots;(i-1)k+1:ik-1;\ldots;(s-1)k+1:sk-1;k,\ldots,ik,\ldots,(s-1)k].$$

Имайки предвид, че $A\in\mathcal{R}^{n\times n}$, където $n=sk-1$, същата пермутация се прилага към редовете и стълбовете на матрицата A, при това съответстващата пермутационна матрица се означава с $\mathcal{P}$ и тогава:

$$\mathcal{A}\mathcal{P}x=\mathcal{P}d,\quad \text{където}\quad \mathcal{A}=\mathcal{P}A\mathcal{P}^T=\begin{pmatrix} A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix},$$

където

$$A_{12}=\begin{pmatrix} \bar{c}_1 & & & \\ \bar{a}_2 & \bar{c}_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \bar{c}_{s-1}\\ & & & \bar{a}_s\end{pmatrix}\in\mathcal{R}^{s(k-1)\times(s-1)},$$

$$A_{21}=\begin{pmatrix} 0 & \cdots & a_k & c_k & \cdots & 0 & & & \\ & & 0 & \cdots & a_{2k} & c_{2k} & \cdots & 0 & \\ & & & \ddots & & \ddots & \ddots & & \ddots \\ & & & & & 0 & \cdots & a_{(s-1)k} & c_{(s-1)k} & \cdots & 0\end{pmatrix},$$

$A_{21}\in\mathcal{R}^{(s-1)\times s(k-1)}$, а

$$\begin{aligned} A_{11} &= \operatorname{diag}\{B_1,B_2,\ldots,B_s\}\in\mathcal{R}^{s(k-1)\times s(k-1)},\\ A_{22} &= \operatorname{diag}\{b_k,b_{2k},\ldots,b_{(s-1)k}\}\in\mathcal{R}^{(s-1)\times(s-1)}.\end{aligned}$$

Така въведената пермутация позволява алгоритъмът да бъде представен в в следния блочен вид:

Етап 1. Получаване на блочна LU факторизация на $\mathcal{A}$

$$\mathcal{A}=\begin{pmatrix} A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix}=LU=\begin{pmatrix} A_{11} & 0\\ A_{21} & I_{s-1}\end{pmatrix}\begin{pmatrix} I_{s(k-1)} & R\\ 0 & S\end{pmatrix}$$

посредством следните стъпки:

1. Получаване на LU факторизация на A_{11} (ако е необходимо използвайки частичен избор на главен елемент)

$$A_{11}=\mathcal{P}_1L_1U_1,$$

където $\mathcal{P}_1$ е пермутационна матрица, L_1 е долнотриъгълна матрица с единици по главния диагонал, а U_1 е горнотриъгълна матрица.

2. Решава се

$$A_{11}R = A_{12},$$

използвайки вече получената в предишната стъпка LU факторизация, след което се конструира т. нар. редуцирана матрица

$$S = A_{22} - A_{21}R.$$

Всъщност S е точно допълнението на Шур на A_{11} в $\mathcal{A}$.

Етап 2. Решава се $Ly = d$, използвайки вече получената в Етап 1 матрица L.

Етап 3. Решава се $Ux = y$, като при това най-напред се решава редуцираната система (с матрицата S), използвайки Гаусово изключване (ако е необходимо с избор на главен елемент), в резултат на което се намират компонентите $x_k, x_{2k}, \ldots, x_{(s-1)k}$ на решението. След това посредством обратна субституция се намират всички останали компоненти.

Имайки предвид направеното описание на алгоритъма лесно се вижда, че той притежава много добри паралелни свойства. Всички изчисления, с изключение на решаването на редуцираната система могат да се направят паралелно, като при това, за да се получи естествено разпаралелване, числото s следва да бъде равно на броя на използваните процесори.

В следващия раздел е представен анализ на разпространяването на грешките от закръгляване от началото до края на изчислителния процес при компютърна реализация. Съществени негови особености са, че той е покомпонентен, както и че е използван обратен анализ в отделните етапи на алгоритъма. Навсякъде извежданите оценки са точни, т.е. не са пренебрегвани високите (след първия) порядъци относно машинната точност, което е едно добро изключение от общото правило на пренебрегване в подобни случаи. Отделните етапи на алгоритъма са анализирани съответно в три леми, които позволяват да се изведат оценки за:

- обратната грешка $|\Delta\mathcal{A}|$ (пермутираната матрица от еквивалентни смущения, в покомпоненетен смисъл).
- правата грешка в решението на системата (в относителен смисъл, по норма безкрайност):

$$\frac{\|\delta x\|_\infty}{\|\hat{x}\|_\infty} = \frac{\|\hat{x} - x\|_\infty}{\|\hat{x}\|_\infty}.$$

Получените оценки съставляват основния резултат в този раздел, формулиран във вид на следната теорема:

Теорема 1.1 *При реализация на алгоритъма на Уанг, за решаване на разглежданата система, е в сила* $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x}=\mathcal{P}d$, *при което*

$$|\Delta\mathcal{A}| \le |\mathcal{A}|\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right]+|\mathcal{A}||N|\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right],$$

а

$$\begin{aligned} h_1(\rho_0) &= (K_1+K_2)f(\rho_0)g(\rho_0)+K_1K_2f^2(\rho_0)+K_1K_2f^2(\rho_0)g(\rho_0),\\ h_2(\rho_0) &= (K_1+K_2)f(\rho_0)g(\rho_0)+2K_1K_2f^2(\rho_0)+K_1K_2f^2(\rho_0)g(\rho_0), \end{aligned}$$

са членове, съдържащи ρ_0 *само във втори и по-висок порядък. Освен това за правата грешка в решението на системата е изпълнено*

$$\begin{aligned} \frac{\|\delta x\|_\infty}{\|\hat{x}\|_\infty} &= \frac{\|\hat{x}-x\|_\infty}{\|\hat{x}\|_\infty}\\ &\le cond(A,\hat{x})\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right]\\ &\quad +cond^*(A,x^*)r\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right]. \end{aligned}$$

В горната теорема $r=\max\{\|\hat{R}\|_\infty,1\}$, $K_1=\max\{k_1,1\}$, $K_2=\max\{k_2,1\}$, където k_1 ограничава ръста на елементите при получаване на LU факторизацията на матрицата A_{11} (Етап 1 на алгоритъма), k_2 ограничава ръста на елементите при прилагане на Гаусово изключване за намиране на решение на редуцираната система (Етап 3 на алгоритъма),

$$f(\rho_0)=4\rho_0+3\rho_0^2+\rho_0^3, \qquad g(\rho_0)=3\rho_0+3\rho_0^2+\rho_0^3,$$

а N е матрицата $N=\begin{pmatrix} 0 & \hat{R}\\ 0 & I_{s-1}\end{pmatrix}$.

Що се отнася, до участващите в оценката за правата грешка, числа на обусловеност, то те са определени по следния начин:

$$cond^*(A,x^*)=\frac{\|\,|A^{-1}|\,|A|\,x^*\,\|_\infty}{\|\hat{x}\|_\infty},$$

където векторът x^* има вида

$$x^*=\left(|\hat{x}_k|e,|\hat{x}_k|,\max\left\{|\hat{x}_k|,|\hat{x}_{2k}|\right\}e,\ldots,|\hat{x}_{(s-1)k}|,\max\left\{|\hat{x}_{(s-2)k}|,|\hat{x}_{(s-1)k}|\right\}e\right)^T,$$

$e=(1,1,\ldots,1)\in\mathcal{R}^{1\times(k-1)}$. Така въведеното число на обусловеност прави оценката за правата грешка в решението на системата по-гъвкава. Потвърждение за това е представено в Раздел 1.5. Също се използва и известното число на обусловеност на Скийл (виж [56])

$$cond(A,\hat{x})=\frac{\|\,|A^{-1}|\,|A|\,|\hat{x}|\,\|_\infty}{\|\hat{x}\|_\infty},$$

като в знаменателя участва полученото изчислено решение $\hat{x}$, вместо теоретичното x. Това прави получените оценки по-реалистични. В изведените в Теорема 1.1 оценки линейните членове (относно ρ_0) са отделени, откъдето се вижда, че членовете от по-висок порядък са достатъчно малки. От тук, ако е необходимо, лесно може да бъде получено приближение от първи порядък. Освен това, както можем да видим, изведената оценка за δx зависи от обусловеността на A и от устойчивостта на алгоритъма (имат се предвид константите K_1, K_2 и r). Първите две константи зависят от ръста на елементите съответно в първия и третия етап на алгоритъма, а множителят r, от ръста на елементите при пресмятане на R. В следващия Раздел 1.3 са разгледани някои често срещани в практиката специални класове от тридиагонални матрици: матрици с диагонално преобладаване (по редове или стълбове), симетрични и положително определени матрици, M-матрици и тотално неотрицателни матрици, за които е показано, че константите K_1, K_2 и rпри пресмятане на R, в Етап 1 и Етап 3 на алгоритъма е ограничен. По-конкретно доказани са следните теореми:

Теорема 1.5 Нека A е неособена тотално неотрицателна или е M-матрица, тогава

$$\|\hat{R}\|_\infty \leq \frac{cond(A)}{1 - k_1 cond(A) f(\rho_0)}.$$

Теорема 1.6 Нека A е неособена матрица с диагонално преобладаване по редове, тогава

$$\|\hat{R}\|_\infty \leq \frac{1}{1 - 2k_1 cond(A) f(\rho_0)}.$$

Теорема 1.7 Нека A е симетрична положително определена, тогава

$$\|\hat{R}\|_\infty \leq \frac{cond_2(A)}{1 - k_1(k-1) cond_2(A) f(\rho_0)},$$

където $cond_2(A) = \|A^{-1}\|_2 \|A\|_2$.

Що се отнася до константите K_1 и K_2, ограничаващи ръста на елементите в Етап 1 и Етап 3 на алгоритъма, то ограничения за тези константи за разгледаните специални класове от матрици са известни от [35]. За да се пренесат обаче известните оценки за K_2 е необходимо да се покаже, че щом A принадлежи на някои от разгледаните специални класове, то и редуцираната матрица S принадлежи на множеството от същите класове (Теорема 1.2, Теорема 1.3, Теорема 1.4).

В края на раздела е показано и че (Теорема 1.8):

$$\frac{\|\Omega S\|_\infty}{\|S\|_\infty} \leq K_1 cond(A) rf(\rho_0),$$

където $\Omega S = \hat{S} - S$ е общата грешка в компютърно пресметнатата редуцирана матрица.

Така в резултат на представените в този раздел изследвания е направен изводът, че щом матрицата на решаваната тридиагонална система принадлежи към някои от разглежданите специални класове от матрици, то алгоритъмът е числено устойчив. Същият резултат е получен и за метода на Гаус в [35]. Това показва, че паралелният алгоритъм на Уанг може да бъде използван безопасно в същите случаи, както и методът на Гаус, което е едно добро изключение от общото хипотетично правило, че паралелните алгоритми са по-неустойчиви от последователните за една и съща задача.

Вече стана ясно, че в ситуацията, когато A е добре обусловена, но не принадлежи към разглежданите специални класове, алгоритъмът може да бъде числено неустойчив, което разбира се е доста неприятно. Такава ситуация може да се случи, когато някои от блоковете (един или повече) по главния диагонал на пермутираната матрица $\mathcal{A}$ са лошо обусловени. Изход от тази неприятна ситуация е посочен в Раздел 1.4, където с цел да бъде подобрена числената устойчивост на изследвания алгоритъм, е използван подход на изкуствено смущаване на някои данни, при който се избягва евентуалното прекъсване на алгоритъма (поради препълване при деление на нула) или възможна поява на взрив на грешките от закръгляване (при деление на числа близки до нулата). В резултат на този подход естествено се получава смутено решение, което се доуточнява, използвайки стандартна процедура за итерационно уточняване [30].

Направени са числени експерименти (Раздел 1.5), които потвърждават, че теоретично получените оценки са почти достижими, както и ефективността на предложения подход на стабилизация. В повечето случаи на практика е необходима само една стъпка на итерационно уточняване на решението, получено в резултат на стабилизирания вариант на изследвания алгоритъм, а в качеството на оптимално смущение се препоръчва стойността 10^{-8}.

Накрая в Раздел 1.6 е представена и анализирана паралелна реализация на изследвания алгоритъм, използвайки PVM (Parallel Vitrual Machine).

Основните резултати, представени в тази глава, са публикувани в [52, 74].

- В Глава 2 е разгледано обобщение на изследвания в Глава 1 метод на Уанг за решаване на лентови системи линейни уравнения (структурирани в блочно тридиагонален вид) и е представен анализ на числената устойчивост на алгоритъма в този случай.

  Нека да отбележим, че хронологията на излагане на изследванията в тази глава е същата, както в предходната. При това навсякъде, където доказателствата на твърденията са аналогични на тези в тридиагоналния случай са пропускани, а там където нещата са специфични или не съвсем очевидни са представяни подробно.

  Главата се състои от четири раздела. Най-напред в Раздел 2.1 е направено описание на обобщения вариант на алгоритъма за решаване на лентови системи линейни уравнения. Разглежда се следната лентова система

  $$Ax = d,$$

  където матрицата $A \in \mathcal{R}^{n\times n}$ и има ширина на лентата $2j+1$ или с други думи броят на нейните ненулеви диагонали над и под главния диагонал е един и същ и е равен на j. При това естествено се предполага, че $2j+1 << n$, т.е. A не е близка до плътна матрица. За простота се предполага още, че $n = ks - j$, където k е произволно цяло положително число, а s е броят на паралелните процесори, които могат да се използват. Всички тези предположения са направени за удобство и не са съществени за представените изследвания.

  Нека да отбележим, че имайки предвид описания в Глава 1 алгоритъм, основните моменти в описанието му се запазват. Променя се само смисълът на отделните елементи. По-конкретно, най-напред направеното в Глава 1 разделяне на решаваната система се запазва, при което $B_i \in \mathcal{R}^{(k-j)\times(k-j)}, i = 1, 2, \ldots, s$, са лентови матрици със същата ширина на лентата като на A, но от няколко (от порядъка на s) пъти по-малка размерност, $\bar{a}_i \in \mathcal{R}^{(k-j)\times j}, i = 2, \ldots, s$, $\bar{c}_i \in \mathcal{R}^{(k-j)\times j}, i = 1, \ldots, s-1$ са матрици от следния вид:

  $$\bar{a}_i = (a_{(i-1)k+1}, 0, \ldots, 0)^T, \quad \bar{c}_i = (0, \ldots, 0, c_{ik-1})^T,$$

  чиито елементи $a_{(i-1)k+1}, c_{ik-1} \in \mathcal{R}^{j\times j}$. Същото важи и за елементите a_{ik}, b_{ik}, c_{ik}, т.е. $a_{ik}, b_{ik}, c_{ik} \in \mathcal{R}^{j\times j}, i = 1, 2, \ldots, s-1$. Накрая, за компонентите на вектора стълб от неизвестни и на дясната част е изпълнено

  $$X_i, D_i \in \mathcal{R}^{(k-j)\times 1}, \ i = 1, 2, \ldots, s, \quad x_{ik}, d_{ik} \in \mathcal{R}^{j\times 1}, \ i = 1, 2, \ldots, s-1.$$

  Понататък алгоритъмът се представя в блочна форма и се структурира по същия начин, както в тридиагоналния случай (предходната

глава), при което отделните негови етапи се запазват. Само ще отбележим, че специфично свойство в лентовия случай е, че конструираната редуцирана матрица S сега е блочно тридиагонална (може да бъде разглеждана и като лентова с ширина на лентата $4j-1$).

Що се отнася до възможностите за разпаралелване на описания алгоритъм, то както и в тридиагоналния случай всички изчисления, с изключение на решаването на редуцираната система могат да се направят паралелно, като при това за естествено разпаралелване са необходими s на брой процесори.

В следващия Раздел 2.2 е представен анализ на разпространяването на грешките от закръгляване от началото до края на изчислителния процес. Съществени негови особености са отново, че той е покомпонентен, както и че е използван на обратен анализ в отделните етапи на алгоритъма. Навсякъде извежданите оценки са отново точни, т.е. не са пренебрегвани високите (след първия) порядъци относно машинната точност. Трите етапа на алгоритъма отново са анализирани съответно в три леми, естествено отчитайки спецификите на алгоритъма в лентовия случай. Основният резултат в раздела съдържа оценки за обратната грешка (в покомпоненетен смисъл) и за правата грешка в решението на системата (в относителен смисъл, по норма безкрайност) и е формулиран в следната теорема:

Теорема 2.1 *При реализация на алгоритъма на Уанг за решаване на разглежданата лентова системата е в сила* $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x}=\mathcal{P}d$, *като*

$$|\Delta\mathcal{A}| \le |\mathcal{A}|h_1(\rho_0) + |\mathcal{A}||N|h_2(\rho_0),$$

а

$$\begin{aligned} h_1(\rho_0) &= K_1 f(\rho_0) + K_2 h(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + K_1 f(\rho_0)g(\rho_0) + K_2 h(\rho_0)g(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0), \\ h_2(\rho_0) &= 3K_1 f(\rho_0) + 2K_2 h(\rho_0) + 2K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + 3K_1 f(\rho_0)g(\rho_0) + 3K_2 h(\rho_0)g(\rho_0) + 3K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0) \\ &\quad + K_1 f(\rho_0)g^2(\rho_0) + K_2 h(\rho_0)g^2(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g^2(\rho_0). \end{aligned}$$

Освен това за правата грешка в решението на системата е изпълнено

$$\frac{\|\delta x\|}{\|\hat{x}\|} = \frac{\|\hat{x}-x\|_\infty}{\|\hat{x}\|_\infty} \le cond(A,\hat{x})h_1(\rho_0) + cond^*(A,x^*)rh_2(\rho_0).$$

В горната теорема смисълът на константите K_1, K_2 и r е същия както при Терема 1.1, функциите $f(\rho_0), g(\rho_0)$ са дефинирани така

$$f(\rho_0) = \gamma_{j+1} + \gamma_{2j+1}, \qquad g(\rho_0) = \gamma_{j+1} + \rho_0,$$

където $\gamma_n = n\rho_0/(1-n\rho_0)$, а N е матрицата $N = \begin{pmatrix} 0 & \hat{R} \\ 0 & I_{j(s-1)} \end{pmatrix}$.

Участващите в оценката за правата грешка, числа на обусловеност, са дефинирани по същия начин, както в Глава 1, при което дефиницията на използваното специално число $cond^*(A, x^*)$ е адаптирана в новите условия, където

$$x^* = (\|\hat{x}_k\|_\infty e, |\hat{x}_k|, \max\{\|\hat{x}_k\|_\infty, \|\hat{x}_{2k}\|_\infty\} e, \ldots, |\hat{x}_{(s-1)k}|, \max\{\|\hat{x}_{(s-2)k}\|_\infty, \|\hat{x}_{(s-1)k}\|_\infty\} e)^T,$$

но смисълът му се запазва. В изведените в Теорема 2.1 оценки линейните членове не са отделени, но при необходимост да бъде получено приближение от първи порядък, това може да бъде направено. Изведената оценка за δx отново зависи от обусловеността на A и от устойчивостта на алгоритъма (имат се предвид константите K_1, K_2 и r). Нека да припомним, че първите две константи зависят от ръста на елементите съответно в първия и третия етап на алгоритъма, а множителят r, от ръста на елементите при пресмятане на R. В следващия Раздел 2.3 са разгледани някои често срещани в практиката специални класове от лентови матрици: матрици с диагонално преобладаване (по редове или стълбове), симетрични и положително определени матрици и M-матрици, за които класове е показано, че ръстът на елементите при пресмятане на R, в Етап 1 и Етап 3 на алгоритъма е ограничен. Доказани са подобни твърдения на тези от Раздел 1.3 (Теореми 2.2 - 2.6). Следователно за разглежданите специални класове от лентови матрици алгоритъмът е числено устойчив. Същият резултат е получен и за метода на Гаус в [36]. Следователно и тук е валиден изводът, че изследваният паралелен алгоритъм може да бъде използван безопасно в същите случаи, както и методът на Гаус, т.е. и в лентовия случай е налице изключение от общото хипотетично правило, че паралелните алгоритми са по-неустойчиви от последователните за една и съща задача.

За подобряване на числената устойчивост на алгоритъма в случай на решаване на лентови системи отново може да бъде използван подхода на стабилизация изложен в Раздел 1.4. Но тъй като по отношение на резултатите няма новости при прилагане на този подход, на него не е отделено специално място.

Направените числени експерименти (Раздел 2.4), потвърждават, че теоретично получените оценки са почти достижими.

Основните резултати, представени в тази глава, са публикувани в [75].

- Както вече отбелязахме, освен методите от типа на разделяне, за паралелно решаване на тридиагонални системи линейни уравения се използват и методите от типа на цикличната редукция (ЦР) [37, 38]. В Глава 3 е предложен един по-устойчив вариант на метода на цикличната редукция [38] (модификация без обратен ход) и направено изследване на числената устойчивост на алгоритъма в този случай.

  Същността на модифицирания метод на цикличната редукция [38] се изразява в това че на всяка стъпка от алгоритъма се намира матрица от специален вид, такава че умножавайки с нея изходната матрица (естествено и дясната част) се получава нова матрица с изместени ненулеви диагонали под и над главния диагонал по направление съответно към долния ляв и горния десен ъгъл на изходната матрица. И ако n е размерността на решаваната система, то след извършването на $\lceil \log_2 n \rceil$ такива стъпки (умножения) тези диагонали изчезват и се получава диагонална матрица. Следователно в този си вид алгоритъмът няма обратен ход, т.е. директно се получава решението на системата.

  Предимства при паралелна реализация на модифицирания вариант на метода цикличната редукция, в сравнение с оригиналния [37] са по-малкия брой паралелни стъпки, поради липса на обратна субституция, както и по-равномерното натоварване на отделните процесори, и следователно в крайна сметка по-добра ефективност.

  Числената устойчивост на тази модификация, при предположение, че матрицата на разглежданата система е неособена е изследвана в [69]. В тази работа ръстът на елементите при реализацията на алгоритъма е ограничен, в случай че изходната матрица принадлежи на някои от следните специални класове: матрици с диагонално преобладаване, симетрични и положително определени, M-матрици или тотално неотрицателни матрици. Откъдето е направен изводът, че за разглежданите специални класове алгоритъмът е числено устойчив. Освен това от направените в [69] изследвания, може да се направи и изводът, че числената устойчивост на модифицирания метод на цикличната редукция е доста близка до тази на метода на Гаус (виж [35], където е представен аналогичен анализ). Следователно, разглежданият паралелен метод на цикличната редукция на практика може да бъде използван в същите случаи, както и метода на Гаус (известен като последователен такъв).

  Проблеми с реализацията на модифицирания метод на цикличната редукция възникват, когато решаваната система е добре обусловена, но нейната матрица не принадлежи към отделените в [69] специални класове от матрици. Тогава е възможно да се получи прекъсване

на алгоритъма (поради препълване при деление на нула) или да се получи взрив на грешките от закръгляване (при деление на числа близки до нулата). В такива случаи ако се приложи вариант за решаване на системата с избор на главен елемент, това би довело до нарушаване на нейната структура и до много комуникации между отделните процесори. Ето защо е желателно да се избегне подхода на избор на главен елемент и да се потърсят други, по-прости за реализация подходи за подобряване на числената устойчивост на алгоритъма. За решаването на възникналия проблем в настоящата глава отново се прилага, вече използваният в Глава 1 подход на изкуствено смущаване на някои данни. Този подход, освен че е прост за реализация и не нарушава структурата на системата. В тази глава е изследвана и числената устойчивост на получения стабилизиран вариант на алгоритъма, а също и неговото приложение за решаване на тридиагонални системи с много десни части.

Главата съдържа шест раздела. В Раздел 3.1 е направено описание на изследвания алгоритъм. Разглежда се тридиагоналната система линейни уравнения

$$Ax = d,$$

където

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix},$$

при което A е добре обусловена тридиагонална матрица. Алгоритъмът се състои от $m+1$ стъпки, където $m = \lceil \log_2 n \rceil$, при което за m стъпки чрез подходящи умножения от ляво матрицата A се преобразува в диагонален вид и на $(m+1)$-вата стъпка се намира решението на системата. В аналитичен вид тези стъпки се описват по следния начин. Последователно за $k = 1, 2, \ldots, m$ се извършват следните действия:

1. Изчисляват се елементите

$$\begin{aligned} \alpha_i^{(k)} &= -\frac{a_i^{(k-1)}}{b_{i-2^{k-1}}^{(k-1)}}, \\ \beta_i^{(k)} &= -\frac{c_i^{(k-1)}}{b_{i+2^{k-1}}^{(k-1)}}, \\ i &= 1, 2, \ldots, n, \end{aligned}$$

на матрицата

$$L^{(k)} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \beta_1^{(k)} & & & & \\ 0 & \ddots & & & & \ddots & & & \\ \vdots & & \ddots & & & & \ddots & & \\ 0 & & & \ddots & & & & \ddots & \\ \alpha_{2^{k-1}+1}^{(k)} & & & & \ddots & & & & \beta_{n-2^{k-1}}^{(k)} \\ & \ddots & & & & \ddots & & & 0 \\ & & \ddots & & & & \ddots & & \vdots \\ & & & \ddots & & & & \ddots & 0 \\ & & & & \alpha_n^{(k)} & 0 & \cdots & 0 & 1 \end{pmatrix},$$

с която се умножава от ляво получените на предишната стъпка матрица $A^{(k-1)}$ и дясна част $d^{(k-1)}$ (за $k = 1$, $A^{(0)} = A$, $d^{(0)} = d$).

2. С помощта на така конструираната матрица $L^{(k)}$ се получава

$$A^{(k)} = L^{(k)} A^{(k-1)}, \; d^{(k)} = L^{(k)} d^{(k-1)},$$

където елементите на $A^{(k)}$ и $d^{(k)}$ се намират по следните формули:

$$\begin{aligned} a_i^{(k)} &= \alpha_i^{(k)} a_{i-2^{k-1}}^{(k-1)}, \\ c_i^{(k)} &= \beta_i^{(k)} c_{i+2^{k-1}}^{(k-1)}, \\ b_i^{(k)} &= b_i^{(k-1)} + \alpha_i^{(k)} c_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} a_{i+2^{k-1}}^{(k-1)}, \\ d_i^{(k)} &= d_i^{(k-1)} + \alpha_i^{(k)} d_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} d_{i+2^{k-1}}^{(k-1)}, \\ i &= 1, 2, \ldots, n. \end{aligned}$$

В резултат, ненулевите диагонали под и над главния диагонал на новата матрица $A^{(k)}$ се преместват в посока съответно към към долния ляв и горния десен ъгъл. При това броят на нулевите диагонали между тях и главния диагонал е $2^k - 1$. По този начин точно след m стъпки се стига до диагоналната матрица

$$A^{(m)} = \begin{pmatrix} b_1^{(m)} & & \\ & \ddots & \\ & & b_n^{(m)} \end{pmatrix}.$$

На $(m+1)$-вата се намира решението на системата

$$x = L^{(m+1)} d^{(m)},$$

където

$$L^{(m+1)} = \begin{pmatrix} \frac{1}{b_1^{(m)}} & & \\ & \ddots & \\ & & \frac{1}{b_n^{(m)}} \end{pmatrix},$$

или по-точно

$$x_i = d_i^{(m)}/b_i^{(m)}, \quad i = 1, 2, \ldots, n.$$

Нека да отбележим, че макар и A да е добре обусловена, при изчисляване на елементите на матрицата $L^{(k)}$ е достатъчно някой от елементите $b_{i-2^{k-1}}^{(k-1)}$ или $b_{i+2^{k-1}}^{(k-1)}$ да стане равен на нула и алгоритъмът да прекъсне, или пък близък до нула и да се получи взрив на грешките от закръгляване. В Раздел 3.2 е даден отговор на въпроса как да се избегне тази неприятна ситуация и представен стабилизиран вариант на разглеждания алгоритъм. Същността се състои в добавяне към алгоритъма на стабилизираща стъпка, при което ако някой от елементите $b_{i-2^{k-1}}^{(k-1)}$ и $b_{i+2^{k-1}}^{(k-1)}$ стане по модул по-малък от някакво предварително избрано достатъчно малко число δ_0, се смущава изкуствено с δ_0. Този подход гарантира, че елементите $b_{i-2^{k-1}}^{(k-1)}$ и $b_{i+2^{k-1}}^{(k-1)}$ ще бъдат достатъчно отдалечени от нулата, така че няма да се налага да се дели на числа близки до нулата. В резултат обаче, се получава и допълнително смутено решение, т.е. влияние върху решението оказват, както грешките от закръгляване, така и направените изкуствени смущения. Полученото решение се доуточнява чрез стандартна процедура за итерационно уточняване [30] (Раздел 3.4).

В следващия Раздел 3.3 е изследвана числената устойчивост на стабилизирания вариант на алгоритъма. При това се използва представения в [69] анализ на числената устойчивост на оригиналния алгоритъм (без изкуствени смущения). В крайна сметка е намерена следната оценка за грешката в решението на системата, получено в резултат на прилагане на стабилизирания вариант на алгоритъма

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq \frac{C\, cond(A,x)(\delta_0 + \frac{\rho_0}{\delta_0^s})}{1 - C\, cond(A,e)(\delta_0 + \frac{\rho_0}{\delta_0^s})},$$

където $C = \mathcal{O}(1)$ и

$$\mathrm{cond}(A,x) = \frac{\| \, |A^{-1}| \, |A| \, |x| \, \|_\infty}{\|x\|_\infty}, \qquad e = (1, 1, \ldots, 1)^T,$$

На практика $s \in [0.6, 1]$, откъдето се получава $\delta_0 \in [10^{-11}, 10^{-8}]$. По-късно в Раздел 3.6, в резултат на много експерименти в качеството на оптимална стойност се препоръчва $\delta_0 = 10^{-9}$.

Както вече бе споменато в Раздел 3.4 е отделено внимание на използваната процедура на итерационно уточняване на решението (виж [30]), с малка модификация

$$
\begin{aligned}
&x^{(0)} = \hat{x};\\
&\textbf{for } k = 1, 2, \ldots\\
&\quad r^{(k-1)} = b - Ax^{(k-1)};\\
&\quad (A + \Delta)y^{(k)} = r^{(k-1)};\\
&\quad x^{(k)} = x^{(k-1)} + y^{(k)};\\
&\textbf{end}
\end{aligned}
$$

Модификацията се изразява в това, че вместо с A смутената система се решава с матрицата $A+\Delta$, където Δ матрицата от всички изкуствени смущения, които са всъщност смущения и в изходната матрица A. По-точно, от факта, че на практика се смущават само някои от диагоналните елементи на A е ясно, че Δ е диагонална матрица, с някои ненулеви диагонални елементи. При прилагане на горе-описаната процедура на итерационно уточняване се налага да се решава няколко пъти смутената система

$$(A + \Delta)y^{(k)} = r^{(k-1)},$$

за чието решаване са предложени различни подходи. Що се отнася до сходимостта на използваната процедура на итерационно уточняване на решението, то използвайки направения в [72] анализ, е получена следната оценка за сходимостта:

$$\|x^{(k)} - x\|_\infty \leq M\|x^{(k-1)} - x\|_\infty,$$

където за константата M е валидна оценката

$$M \leq \frac{C^* \, cond(A, x)\delta_0}{1 - C^* \, cond(A, e)\delta_0}, \qquad C^* = \mathcal{O}(1).$$

Този резултат показва, че когато матрицата A не е лошо обусловена, тогава грешката в решението x се намалява съществено след една итерация на итерационно уточняване. Така след една или две итерации се намира решение, което е достатъчно близко до точното. При това итерациите спират, когато поне едно от следните две условия са изпълнят:

1. $\|Ax^{(k)} - d\|_\infty / \|d\|_\infty \leq 1000\rho_0$;
2. Броят на итерациите е > 10;

Специално внимание върху предимствата при прилагане на предложения стабилизиран алгоритъм за решаване на системи линейни уравнения с много десни части, е отделено в Раздел 3.5. В края на главата (Раздел 3.6) са представени множество числени експерименти, включително и със случайни матрици, които показват,че стабилизираният вариант на алгоритъма работи значително по-добре от оригиналния, а също и че за итерационното уточняване на решението са необходими само една-две итерации.

Основните резултати, представени в тази глава, са публикувани в [51] и [53].

- В Глава 4 е разгледано обобщение на метода на цикличната редукция [38] (модификация без обратен ход) в случая на решаване на блочно тридиагонални системи линейни уравнения и е представен анализ на числената устойчивост на алгоритъма в този случай, при предположение че матрицата на решаваната система е с равномерно блочно диагонално преобладаване по стълбове (смисълът на това понятие се изяснява понататък). Полученият алгоритъм притежава определени предимства при паралелна реализация, изразяващи се най-вече в равномерно натоварване на отделните процесори и намалено време за комуникациии (имайки предвид блочния вид на алгоритъма), както и малък брой паралелни стъпки (поради липсата на обратна субституция).

  Нека да отбележим, че описание на същия алгоритъм в тридиагоналния случай е направено в предходната глава, а също и че изследване на неговата числена устойчивост в този случай е представено в [69], където е използван подход на обратен анализ. За съжаление обаче такъв подход в блочния случай не дава резултат. Това налага изследването да се направи по друг начин, по-точно използван е подход на прав анализ.

  Главата се състои от четири раздела. В Раздел 4.1 е направено описание на изследвания модифициран алгоритъм на цикличната редукция обобщен в блочен вид.

  Разглежда се блочно тридиагоналната система

$$AX = D,$$

където

$$A = \begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & C_{n-1} \\ & & & A_n & B_n \end{pmatrix}, \quad D = \begin{pmatrix} D_1 \\ \vdots \\ D_n \end{pmatrix}.$$

Блоковете $A_i, B_i, C_i \in \mathcal{R}^{N\times N}$, и в общия случай са плътни, $D_i \in \mathcal{R}^N$ са вектори, а n блочната размерност на решаваната система.

Нека да отбележим, че представените в главата изследвания запазват своята валидност и в случая, когато A е лентова матрица (произволна лентова матрица би могла да бъде структурирана в блочно тридиагонален вид).

Алгоритъмът, както и в тридиагоналния случай се състои от $m+1$ стъпки, където $m = \lceil \log_2 n \rceil$. При което на всяка k-та стъпка $k = 1, 2, \ldots, m$ се конструира матрица $L^{(k)}$, с която се умножава от ляво получените на предишната стъпка матрица $A^{(k-1)}$ и дясна част $D^{(k-1)}$ (за $k = 1$, $A^{(0)} = A$, $D^{(0)} = D$).

$$A^{(k)} = L^{(k)} A^{(k-1)}, \; D^{(k)} = L^{(k)} D^{(k-1)},$$

В резултат, ненулевите диагонали под и над главния диагонал на новата матрица $A^{(k)}$ са се преместили в посока към долния ляв и горния десен ъгъл. При това броят на нулевите диагонали между тях и главния диагонал е $2^k - 1$. По този начин точно след m стъпки матрицата A се преобразува в блочно диагонален вид, след което на $(m+1)$-вата стъпка се намира решението на системата.

В Раздел 4.2, при предположение за обратимост на блоковете по главния диагонал на матрицата на изходната система и равномерно блочно диагонално преобладаване по стълбове, т.е.

$$\|A_i[B_{i-1}]^{-1}\| \le p, \; \|C_i[B_{i+1}]^{-1}\| \le q, \; s = p + q \le 1, \; i = 1, 2, \ldots, n,$$

относно произволна норма, за която е изпълнено $\|I\| = 1$ и където p и q не зависят от i, са разгледани някои основни свойства на алгоритъма. Тези свойства са формулирани и доказани в три леми, благодарение на които в следващия раздел е получена оценка за грешката в решението на системата. При това е въведена следната норма:

$$\|G\|_{B\infty} = \max_{1\le i\le n} \sum_{j=1}^{n} \|G_{ij}\|,$$

$$\|g\|_{B\infty} = \max_{1\le j\le n} \|g_j\|,$$

където G е една произволна блочна матрица с блочна размерност $n \times n$, и g е произволен блочен вектор (всеки елемент, на който е също вектор) с блочна размерност n.

Изследванията в следващия Раздел 4.3 са направени при предположение, че матрицата на решаваната система притежава свойството на равномерно блочно диагонално преобладаване по стълбове. В този раздел, използвайки подхода на правия анализ е проследено разпространението на грешките от закръгляване от началото до края на изчислителния процес при компютърната реализация на изследвания алгоритъм. Използването на такъв подход се налага и поради факта, че обратния анализ не дава резултат. При това за яснота, навсякъде са пренебрегвани членовете съдържащи машинната точност във втори и по-висок порядък. В противен случай се получават сложни и дълги изрази, работата с които значително би усложнила извеждането на оценките и би влошила тяхната обозримост. В резултат е получено линейно приближение за правата грешка в решението на системата, при което са разгледани два случая. Първият, когато $s = 1$, т.е. изходната матрица е с нестрого блочно диагонално преобладаване по стълбове. Тогава оценката има вида:

$$\frac{\|\delta X\|_{B\infty}}{\|X\|_{B\infty}} \leq \frac{C_N b n^2}{\|X\|_{B\infty}} \left(\bar{B}\|X\|_{B\infty} + (\log_2 n + 1)\|D\|_{B\infty}\right) \rho_0 \leq f(n,N)\kappa\rho_0,$$

където C_N е константа зависеща линейно от N,

$$\begin{aligned} b &= \max_i \|B_i^{-1}\|, \qquad \bar{B} = \max_i \|B_i\| \\ \kappa &= \frac{b(\bar{B}\|X\|_{B\infty} + \|D\|_{B\infty})}{\|X\|_{B\infty}}, \; f(n,N) = C_N n^2 \log_2 n, \end{aligned}$$

тук κ е число на обусловеност зависещо от входните данни.

Вторият случай е, когато $0 < s < 1$, т.е. изходната матрица е със строго блочно диагонално преобладаване по стълбове. Тогава оценката има вида:

$$\frac{\|\delta X\|_{B\infty}}{\|X\|_{B\infty}} \leq \frac{C_N b g(s)}{\|X\|_{B\infty}} \left(\bar{B}\|X\|_{B\infty} + (\log_2 n + 1)\|D\|_{B\infty}\right) \rho_0 \leq h(n,N)\kappa\rho_0,$$

където $h(n,N) = C_N g(s) \log_2 n$, κ е вече дефинираното по-горе число на обусловеност, а

$$g(s) = \prod_{j=0}^{j_0} (1 + s^{2^j})^2, \quad j_0 = \lceil \log_2 \log_s (\frac{1}{2}\rho_0) \rceil,$$

е лесно изчислима функция, чиито стойности за $s \in (0,1)$ са ограничени.

Въз основа на изведените оценки са направени изводите, че когато $s < 1$ (и разбира се s не е близко до 1) относителната грешка в решението на системата има порядък $\mathcal{O}(C_N g(s)(\log_2 n)\kappa\rho_0)$, който намалява, когато s намалява. В другия случай, когато $s = 1$ или е близко до 1, относителната грешка в решението на системата има порядък $\mathcal{O}(C_N n^2(\log_2 n)\kappa\rho_0)$. Накрая ако сравним порядъка на грешките в двата случая, то очевидно в случая на строго блочно диагонално преобладаване по стълбове грешката е от порядък $\mathcal{O}(n^2)$ по-малка от тази в другия случай.

В края на главата (Раздел 4.4) са представени числени експерименти, които потвърждават, че изведените теоретични оценки за правата грешка в решението на системата са почти достижими.

Основните резултатите, представени в тази глава, са публикувани в [73].

Научни приноси

1. Представено е пълно изследване на числената устойчивост на метода на Уанг [64] за решаване на тридиагонални и лентови системи линейни уравнения. Разгледани са някои специални класове от матрици, за които е показано, че алгоритъмът е числено устойчив, а грешката в решението на системата зависи главно от нейната обусловеност. Това означава, че паралелният метод на Уанг може да бъде използван успешно в същите случаи, както и методът на Гаус. Приложените числени експерименти доказват, че изведените теоретични оценки са почти достижими.

2. Представено е пълно изследване на числената устойчивост на метода на цикличната редукция (модификация без обратен ход) [38] за решаване на блочно тридиагонални системи линейни уравнения, притежаващи свойството на равномерно блочно диагонално преобладаване по стълбове. Получена е оценка за грешката в решението на системата, при което са разгледани случаите на строго и нестрого равномерно блочно диагонално преобладаване по стълбове. Доказано е, че в случая на строго преобладаване грешката е от порядък $\mathcal{O}(n^2)$ по-малка от тази в другия случай (n е блочната размерност на системата). Изведените теоретични оценки са почти достижими, което се доказва от приложените числени експерименти.

3. Подобрена е числената устойчивост на изследваните методи на Уанг и цикличната редукция в случай на решаване на произволни добре обусловени тридиагонални системи линейни уравнения. Представени са числени експерименти, които потвърждават ефективността на предложения подход за подобряване на числената устойчивост на алгоритмите.

Апробация на дисертационната работа

Резултатите от тази дисертация са докладвани на следните международни конференции и симпозиуми:

- International Conference on Numerical Methods and Applications, Bankya, Bulgaria, August 21–26, 1994.
- International Conference on Differential equations and Applications, Rousse, Bulgaria, August 21–24, 1995.
- First International Workshop on Numerical Analysis and Applications, Rousse, Bulgaria, June 24–27, 1996.
- Sixth Conference of ILAS, Chemnitz, Germany, August 14–17, 1996.
- Fourth International Conference on Numerical Methods and Applications, Sofia, August 19–23, 1998.

Публикации

Резултатите от тази дисертация са публикувани както следва:

- В международни научни списания:
  - Pavlov, V., P. Yalamov. Stabilization by Perturbation of Ill-Conditioned Cyclic Reduction. *International Journal of Computer Mathematics*, 68 (1998), 273–283 (импакт фактор 0.126).
  - Yalamov, P., V. Pavlov. Stability of the Block Cyclic Reduction. *Linear Algebra and Its Applications*, 249 (1996), 341–358 (импакт фактор 0.430).
  - Yalamov, P., V. Pavlov. On the Stabilty of a Partitioning Algorithm for Tridiagonal Systems. *SIAM J. Matrix Anal. Appl.*, v. 20, N 1 (1999), 159–181 (импакт фактор 1.000).

- В сборници с доклади от международни научни конференции:

  - Pavlov, V. Iterative Refinement for Ill-Conditioned Cyclic Reduction. *Proc. Fifth International Conference on Differential Equations and Applications*, (Eds. S. Bilchev and S. Tersian), Rousse, August 24–29, 1995, 84–95.

  - Pavlov, V., D. Todorova. Stabilization and Experience with the Partitioning Method for Tridiagonal Systems. *Lecture Notes in Computer Science*, (Eds. L. Vulkov, J. Wasniewski and P. Yalamov), Springer, 1196 (1997), 380–388 (импакт фактор 0.296).

  - Yalamov, P., V. Pavlov. Backward Stabilty of a Parallel Partitioning Algorithm for Banded Linear Systems. *Proc. of 4th International Conference on Numerical Methods and Applications (Eds. O. Iliev et. al.), Sofia, August 19–23, 1998*, World Scientific Publ., 655–663, 1999.

В заключение изказвам благодарност на колегите от Секцията по числени методи към Института по математика и информатика на БАН и от Секцията по високо производителни системи и алгоритми към Централната лаборатория по паралелна обработка на информацията на БАН за творческата атмосфера при съвместните ни контакти и на колегите ми от Центъра по приложна математика и информатика към Педагогическия факултет на РУ "А. Кънчев" за предоставената ми възможност за пълноценна творческа работа.

Най-голяма признателност дължа на научния ми ръководител доц. д-р Пламен Ялъмов, който ме насочи към интересната проблематика на численото поведение на алгоритмите, оказваше ми постоянно помощ с много полезни съвети и проявяваше изключителна загриженост и търпение през периода на съвместна работа.

Глава 1

Изследване на числената устойчивост на метода на Уанг за решаване на тридиагонални системи линейни уравнения

Основните резултати, представени в тази глава, са публикувани в статиите:

- Yalamov, P., V. Pavlov. On the Stabilty of a Partitioning Algorithm for Tridiagonal Systems. *SIAM J. Matrix Anal. Appl.*, v. 20, N 1 (1999), 159–181.

- Pavlov, V., D. Todorova. Stabilization and Experience with the Partitioning Method for Tridiagonal Systems. *Lecture Notes in Computer Science*, (Eds. L. Vulkov, J. Wasniewski and P. Yalamov), Springer, 1196 (1997), 380–388.

Сред директните методи за решаване на тридиагонални системи линейни уравнения са така наречените методи на разделяне (partitioning) на системата. Този тип методи са характерни с това, че дават ефективни паралелните алгоритми, които са доста популярни поради простота на разделяне на изчислителната работа между отделните процесори. Основната идея при тези методи е дадената система да бъде блочно разделена по подходящ начин на определен брой подсистеми, които могат да бъдат решени паралелно, след което се конструира и решава т.нар. редуцирана система (с размерност от порядъка на броя на използваните процесори) и накрая посредством обикновено последователно заместване се намират всички останали компоненти на решението на изходната система (това

също може да бъде направено паралелно).

В настоящата глава е направено пълно изследване на числената устойчивост на метода на Уанг [64] за решаване на тридиагонални системи линейни уравнения. Този метод е типичен представител на методите на разделяне на системата. Важно негово свойство е, че конструираната редуцирана система е отново тридиагонална. Що се отнася до числената устойчивост на метода на Уанг, то отделни нейни елементи са засегнати в [9, 10, 14, 63]. В [14] е доказано (в случай на плътни матрици), че ако матрицата на изходната система е симетрична и положително определена или M-матрица, то и редуцираната матрица е от същия тип. В [10, 63] е доказано подобно нещо относно свойството на (съответно) диагонално преобладаване по редове и строго диагонално преобладаване (в по-общ смисъл). В [9] е показано е, че при малки смущения във входните данни (има се предвид матрицата на решаваната система), грешките в редуцираната матрица са малки в определен смисъл. Ясно е обаче, че това са по-скоро отделни резултати, касаещи числената устойчивост на метода на Уанг.

Главата се състои от шест раздела. Най-напред в Раздел 1.1 е направено описание на разглеждания алгоритъм. При това за удобство и по-голяма яснота в изложението, алгоритъмът е структуриран в три етапа. В следващия раздел е представен пълен анализ на разпространението на грешките от закръгляване при компютърна реализация на алгоритъма. Съществени негови особености са, че той е покомпонентен, както и че е използван комбиниран подход на прав и обратен анализ в отделните етапи на алгоритъма. Навсякъде извежданите оценки са точни, т.е. не са пренебрегвани високите (след първия) порядъци относно машинната точност, което е едно добро изключение от общото правило на пренебрегване в подобни случаи. Трите етапа на алгоритъма са анализирани съответно в три леми. Основният резултат в раздела съдържа оценки за обратната грешка (матрицата от еквивалентни смущения, в покомпоненетен смисъл) и за правата грешка в решението на системата (в относителен смисъл, по норма безкрайност). Съществено за тези оценки е, че линейните членове са отделени, откъдето се вижда, че членовете от по-висок порядък са достатъчно малки и при необходимост, лесно може да се намери приближение от първи порядък. Освен това, получените оценки показват, че грешката в решението на системата зависи от нейната обусловеност и от устойчивостта на алгоритъма. Естествено, влияние върху устойчивостта оказва ръстът на елементите в отделните етапи на алгоритъма. В общия случай, дори и решаваната система да е добре обусловена, този ръст може да е неограничен и в крайна сметка полученото решение да се различава съществено от точното. Поради тази причина в следващия Раздел 1.3 са

разгледани някои специални класове от тридиагонални матрици: матрици с диагонално преобладаване (по редове или стълбове), симетрични и положително определени матрици, M-матрици и тотално неотрицателни матрици. За тези класове ръстът на елементите в отделните етапи на алгоритъма е ограничен от малки константи. Нека да отбележим, че това са същите класове, за които е устойчив и методът на Гаус [35], което показва, че методът на Уанг може да бъде използван на практика в същите случаи, както и методът на Гаус. Всичко това дава основание да се направи важният извод, че за разглежданите специални класове алгоритъмът е числено устойчив, а грешката в решението на системата зависи главно от обусловеността на решаваната система. В случай обаче, когато матрицата на разглежданата система не принадлежи към споменатите специални класове, макар и добре обусловена, алгоритъмът може да прекъсне или да се получи взрив на грешките от закръгляване. С цел да се подобри числената устойчивост на алгоритъма в [9] авторите предлагат да се използва QR разлагане при паралелното решаване на получените при разделянето подсистеми. След което, при необходимост се прави ново блочно разделяне на изходната система, така че новите подсистеми да са добре обусловени. Този подход, обаче води до съществено увеличаване броя на аритметичните операции и изисква повече машинно време. Ето защо в Раздел 1.4 е разгледан един друг вариант за стабилизиране на алгоритъма. Този вариант се свежда до смущаване на някои данни, при който се избягва деление на числа равни или близки до нулата (подобен подход е използван в [23, 32] за други алгоритми). В резултат на това обаче, се получава и смутено решение, което се доуточнява чрез използване на достатъчно бърза процедура на итерационно уточняване [30]. Представени са числени експерименти (Раздел 1.5), които потвърждават, че теоретично получените оценки са почти достижими, както и ефективността на предложения подход на стабилизация. Накрая в Раздел 1.6 е представена и анализирана паралелна реализация на изследвания алгоритъм, използвайки PVM (Parallel Vitrual Machine).

1.1 Описание на алгоритъма на Уанг

Нека да разгледаме следната тридиагонална система линейни уравнения

$$Ax = d, \tag{1.1}$$

където

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}.$$

За простота да допуснем, че $n = ks-1$, където k е цяло положително число, а s е броя на паралелните процесори, които бихме желали да използваме. Нека да направим следното разделяне на матрицата A, търсеното решение x и на дясната част d на разглежданата система (1.1):

$$\begin{pmatrix} B_1 & \bar{c}_1 & & & & & & \\ a_k & b_k & c_k & & & & & \\ & \bar{a}_2 & B_2 & \bar{c}_2 & & & & \\ & & a_{2k} & b_{2k} & c_{2k} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \bar{a}_{s-1} & B_{s-1} & \bar{c}_{s-1} & \\ & & & & & a_{(s-1)k} & b_{(s-1)k} & c_{(s-1)k} \\ & & & & & & \bar{a}_s & B_s \end{pmatrix} \begin{pmatrix} X_1 \\ x_k \\ X_2 \\ x_{2k} \\ \vdots \\ X_{s-1} \\ x_{(s-1)k} \\ X_s \end{pmatrix} = \begin{pmatrix} D_1 \\ d_k \\ D_2 \\ d_{2k} \\ \vdots \\ D_{s-1} \\ d_{(s-1)k} \\ D_s \end{pmatrix},$$

където $B_i \in \mathcal{R}^{(k-1)\times(k-1)}, i = 1, 2, \ldots, s$, е тридиагонална матрица

$$B_i = \begin{pmatrix} b_{(i-1)k+1} & c_{(i-1)k+1} & & & \\ a_{(i-1)k+2} & b_{(i-1)k+2} & c_{(i-1)k+2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{ik-2} \\ & & & a_{ik-1} & b_{ik-1} \end{pmatrix},$$

$\bar{a}_i, \bar{c}_i$ са вектори от следния вид:

$$\begin{aligned} \bar{a}_i &= (a_{(i-1)k+1}, 0, \ldots, 0)^T \in \mathcal{R}^{(k-1)\times 1}, \quad i = 2, \ldots, s, \\ \bar{c}_i &= (0, \ldots, 0, c_{ik-1})^T \in \mathcal{R}^{(k-1)\times 1}, \quad i = 1, \ldots, s-1, \end{aligned}$$

а $X_i, D_i, i = 1, 2, \ldots, s$, са вектори от размер $k-1$ от следната форма

$$X_i = (x_{(i-1)k+1}, x_{(i-1)k+2}, \ldots, x_{ik-1})^T, D_i = (d_{(i-1)k+1}, d_{(i-1)k+2}, \ldots, d_{ik-1})^T.$$

Нека, за удобство при изложението на направените изследвания, да представим разглеждания алгоритъм в блочна форма. За тази цел най-напред да дефинираме следната пермутация на числата $[1, \ldots, sk-1]$:

$$[1 : k-1; \ldots; (i-1)k+1 : ik-1; \ldots; (s-1)k+1 : sk-1; k, \ldots, ik, \ldots, (s-1)k].$$

Имайки предвид, че $A \in \mathcal{R}^{n\times n}$, където $n = sk - 1$, нека да приложим същата пермутация към редовете и стълбовете на матрицата A. Тогава ако означим съответната пермутационна матрица с $\mathcal{P}$, то това бихме могли да изразим по следния начин:

$$\mathcal{A}\mathcal{P}x = \mathcal{P}d, \quad \text{където} \quad \mathcal{A} = \mathcal{P}A\mathcal{P}^T = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{1.2}$$

където $A_{11} = \mathrm{diag}\{B_1, B_2, \ldots, B_s\} \in \mathcal{R}^{s(k-1)\times s(k-1)}$,

$$A_{12} = \begin{pmatrix} \bar{c}_1 & & & \\ \bar{a}_2 & \bar{c}_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \bar{c}_{s-1} \\ & & & \bar{a}_s \end{pmatrix} \in \mathcal{R}^{s(k-1)\times(s-1)},$$

$$A_{21} = \begin{pmatrix} 0 & \cdots & a_k & c_k & \cdots & 0 & & & \\ & & & 0 & \cdots & a_{2k} & c_{2k} & \cdots & 0 \\ & & & & \ddots & & \ddots & \ddots & & \ddots \\ & & & & & & 0 & \cdots & a_{(s-1)k} & c_{(s-1)k} & \cdots & 0 \end{pmatrix} \in \mathcal{R}^{(s-1)\times s(k-1)},$$

и $A_{22} = \mathrm{diag}(b_k, b_{2k}, \ldots, b_{(s-1)k}) \in \mathcal{R}^{(s-1)\times(s-1)}$.

Понататък ще правим разлика между двете матрици A (оригиналната) и $\mathcal{A}$ (пермутираната). Очевидно направената пермутация няма да оказва влияние върху числената устойчивост на изследвания алгоритъм, но както ще видим в Раздел 1.3, тя оказва влияние при извеждане на някои оценки. Що се отнася до вектора от неизвестни x и дясната част d, то пермутираните вектори ще означаваме посредством $\mathcal{P}x$ и $\mathcal{P}d$. В крайна сметка ще се стремим да изведем оценки по норма безкрайност за грешката в решението x, които естествено не се влияят от това дали неговите компоненти са разместени или не.

Алгоритъмът би могъл да бъде структуриран по следния начин:

Етап 1. Получаване на блочна LU-факторизация на $\mathcal{A}$

$$\mathcal{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = LU = \begin{pmatrix} A_{11} & 0 \\ A_{21} & I_{s-1} \end{pmatrix} \begin{pmatrix} I_{s(k-1)} & R \\ 0 & S \end{pmatrix} \tag{1.3}$$

посредством следните стъпки:

1. Получаване на LU-факторизация на A_{11} (ако е необходимо използвайки частичен избор на главен елемент по редове)

$$A_{11} = \mathcal{P}_1 L_1 U_1,$$

където $\mathcal{P}_1$ е пермутационна матрица, L_1 е долнотриъгълна матрица с единици по главния диагонал, а U_1 е горнотриъгълна матрица с диагонални елементи $u_1^{(1)}, u_2^{(1)}, \ldots, u_{s(k-1)}^{(1)}$.

2. Решава се

$$A_{11}R = A_{12}, \tag{1.4}$$

използвайки вече получената в предишната стъпка LU-факторизация, след което се конструира т. нар. редуцирана матрица

$$S = A_{22} - A_{21}R. \tag{1.5}$$

Всъщност S е точно допълнението на Шур на A_{11} в $\mathcal{A}$.

При това естествено считаме, че разлагането (1.3) съществува, т.е. предполагаме, че A и A_{11} са неособени матрици. Възможно е обаче A_{11} да бъде особена или някои от блоковете B_i да са особени и това би довело до прекъсване на алгоритъма или до взрив на грешките от закръгляване. Коментар за изход от тази опасна ситуация е направен в Раздел 1.4.

Етап 2. Решава се $Ly = d$, използвайки вече получената в Етап 1 матрица L.

Етап 3. Решава се $Ux = y$, като най-напред се решава редуцираната система (с матрицата S), използвайки Гаусово изключване (ако е необходимо с избор на главен елемент), в резултат на което се намират компонентите $x_k, x_{2k}, \ldots, x_{(s-1)k}$ на решението. След това посредством обратна субституция се намират всички останали компоненти.

Ако вземем предвид структурата на A_{11} и A_{12}, и (1.4), то лесно се вижда, че матрицата R е също структурирана и има следния вид:

$$R = \begin{pmatrix} p^{(1)} & & & \\ q^{(2)} & p^{(2)} & & \\ & \ddots & \ddots & \\ & & \ddots & p^{(s-1)} \\ & & & q^{(s)} \end{pmatrix} \in \mathcal{R}^{s(k-1)\times(s-1)}, \tag{1.6}$$

където

$$\begin{aligned} p^{(i)} &= (p_{(i-1)k+1}, p_{(i-1)k+2}, \ldots, p_{ik-1})^T \in \mathcal{R}^{(k-1)\times 1}, \\ q^{(i)} &= (q_{(i-1)k+1}, q_{(i-1)k+2}, \ldots, q_{ik-1})^T \in \mathcal{R}^{(k-1)\times 1}. \end{aligned}$$

Що се отнася до редуцираната матрица, то специфично свойство на алгоритъма е, че тя е отново тридиагонална

$$S = \begin{pmatrix} v_1 & w_1 & & & \\ u_2 & v_2 & w_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & w_{s-2} \\ & & & u_{s-1} & v_{s-1} \end{pmatrix} \in \mathcal{R}^{j(s-1)\times j(s-1)},$$

където елементите на S се получават по следните формули:

$$u_i = -a_{ik}q_{ik-1}, \quad v_i = b_{ik} - a_{ik}p_{ik-1} - c_{ik}q_{ik+1}, \quad w_i = -c_{ik}p_{ik+1}. \tag{1.7}$$

И накрая, имайки предвид направеното структуриране на алгоритъма и блочната структура на A_{11} и R, то лесно се вижда, че алгоритъмът притежава много добри паралелни свойства. По-точно всички изчисления, с изключение на решаването на редуцираната система могат да се направят паралелно, като при това, за да се получи естествено разпаралелване следва числото s да бъде равно на броя на използваните процесори.

1.2 Анализ на грешките от закръгляване

В този раздел ще представим пълен анализ на разпространението на грешките от закръгляване при компютърна реализация на описания алгоритъм.

Нека $\hat{x}$ е компютърно изчисленото по метода на Гаус решение на тридиагоналната система $Ax = d$. Тогава естествено можем да се постави въпросът за това каква е "най-добрата" оценка за грешката в решението на суистемата. На този въпрос, използвайки подхода на обратния анализ, теорията на смущенията и свойствата на LU разлагането, е даден отговор в [35]. В тази работа е показано, че ако се използва LU разлагане за решаване на системата $Ax = d$, то полученото числено решение $\hat{x}$ удовлетворява

$$(A + \Delta A)\hat{x} = d,$$

където за обратната грешка е изпълнено

$$|\Delta A| \leq f(\rho_0)|\hat{L}||\hat{U}|, \quad f(\rho_0) = 4\rho_0 + 3\rho_0^2 + \rho_0^3.$$

Да допуснем, че $|\hat{L}||\hat{U}| \leq K|A|$, където K е константа, която е горна граница за ръста на елементите при намиране на LU разлагането на матрицата

A. Така получаваме

$$|\Delta A| \leq K|A|f(\rho_0). \tag{1.8}$$

Нека да отбележим че дори и A да е добре обусловена е възможно ръстът на елементите при намиране на нейното LU разлагане да бъде неограничен или да е ограничен, но константата K е твърде голяма. Изход от тази неприятна ситуация е посочен в Раздел 1.4. Конкретни стойности за K за някои специални класове от матрици (същите класове са разгледани и в Раздел 1.3) са намерени в [35].

Нека сега да представим нашите изследвания относно числената устойчивост на алгоритъма. Отделните негови етапи са анализирани в следващите три леми, след което в Теорема 1.1 е представен и основният резултат в този раздел.

Лема 1.1 *Ако е изпълнено* $L\hat{U} = \mathcal{A} + E$, *тогава за обратната грешка* E, *получена от изчисленията за намиране на блочната* LU *факторизация на* $\mathcal{A}$, *е валидна оценката*

$$|E| \leq K_1|\mathcal{A}||N|f(\rho_0), \quad K_1 = \max\{k_1, 1\},$$

където k_1 *ограничава ръста на елементите при получаване на* LU *факторизацията на матрицата* A_{11} *(Етап 1 на алгоритъма), а* N *е следната матрица*

$$N = \begin{pmatrix} 0 & \hat{R} \\ 0 & I_{s-1} \end{pmatrix}.$$

Доказателство. Нека да разгледаме матричното умножение

$$L\hat{U} = \begin{pmatrix} A_{11} & 0 \\ A_{21} & I_{s-1} \end{pmatrix} \begin{pmatrix} I_{s(k-1)} & R + \delta R \\ 0 & \tilde{S} + \delta S \end{pmatrix},$$

където $\tilde{S} = A_{22} - A_{21}\hat{R}$. Тук посредством "вълна" сме означили матрицата S намерена по тази формула в точна аритметика, а грешката, която се поражда при пресмятането е означена с δS. Матрицата L е без "шапка" тъй като тя се състои само от входни елементи, т.е. не са необходими изчисления. Като резултат от разглежданото матрично умножение получаваме

$$L\hat{U} = \begin{pmatrix} A_{11} & A_{12} + A_{11}\delta R \\ A_{21} & A_{21}\hat{R} + \tilde{S} + \delta S \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} + A_{11}\delta R \\ A_{21} & A_{22} + \delta S \end{pmatrix} = \mathcal{A} + E,$$

където с E е означена обратната грешка от изчисленията за получаване на блочната LU факторизация

$$E = \begin{pmatrix} 0 & A_{11}\delta R \\ 0 & \delta S \end{pmatrix}. \tag{1.9}$$

За да оценим E е необходимо да получим оценка за нейните ненулеви елементи. Да започнем от елемента $A_{11}\delta R$ (δR е матрица). Нека, за да опростим разсъжденията, най-напред да оценим елемент от вида $A_{11}\delta y$, където δy е правата грешка, получена при изчислението на вектора y, който е решение на системата $A_{11}y = z$ (z също е вектор). Използвайки класическия обратен анализ (виж [67]), получаваме

$$(A_{11} + \Delta A_{11})\hat{y} = (A_{11} + \Delta A_{11})(y + \delta y) = z,$$

откъдето

$$A_{11}\delta y = -\Delta A_{11}\hat{y}, \tag{1.10}$$

и следователно

$$|A_{11}\delta y| \leq |\Delta A_{11}||\hat{y}|. \tag{1.11}$$

Ако използваме сега (1.8) за обратния анализ при прилагане на Гаусово изключване за тридиагонални системи ще имаме, че $|\Delta A_{11}| \leq k_1|A_{11}|f(\rho_0)$, където k_1 е константа, ограничаваща ръста на елементите при прилагане на Гаусово изключване за матрицата A_{11} в Етап 1 на алгоритъма. Замествайки тази оценка в (1.11), стигаме до

$$|A_{11}\delta y| \leq k_1|A_{11}||\hat{y}|f(\rho_0). \tag{1.12}$$

В нашия случай, за да получим оценка за $A_{11}\delta R$, е достатъчно да заместим y с R и z с A_{12} и да приложим (1.12) за всеки стълб на матрицата R:

$$|A_{11}\delta R| \leq k_1|A_{11}||\hat{R}|f(\rho_0). \tag{1.13}$$

Нека сега да оценим другия ненулев елемент δS на матрицата E. Както отбелязахме, елементите на матрицата S се пресмятат по формули (1.7). Използвайки подхода на правия анализ за диагоналните елементи, получаваме

$$\begin{aligned}
\hat{v}_i &= \{[b_{ik} - a_{ik}p_{ik-1}(1+\sigma_1)](1+\sigma_2) - q_{ik+1}c_{ik}(1+\sigma_3)\}(1+\sigma_4) \\
&= b_{ik} - a_{ik}p_{ik-1} - c_{ik}q_{ik+1} + b_{ik}(\sigma_2 + \sigma_4 + \sigma_1\sigma_4) \\
&\quad -a_{ik}p_{ik-1}(\sigma_1 + \sigma_2 + \sigma_4 + \sigma_1\sigma_2 + \sigma_1\sigma_4 + \sigma_2\sigma_4 + \sigma_1\sigma_2\sigma_4) \\
&\quad -q_{ik+1}c_{ik}(\sigma_3 + \sigma_4 + \sigma_3\sigma_4),
\end{aligned}$$

където $|\sigma_i| \leq \rho_0, i = 1,2,3,4$. Следователно за правата грешка δv_i е изпълнено

$$\begin{aligned}
|\delta v_i| &\leq |b_{ik}|(2\rho_0 + \rho_0^2) + |a_{ik}p_{ik-1}|(3\rho_0 + 3\rho_0^2 + \rho_0^3) + |q_{ik+1}c_{ik}|(2\rho_0 + \rho_0^2) \\
&\leq (|b_{ik}| + |a_{ik}p_{ik-1}| + |q_{ik+1}c_{ik}|)(3\rho_0 + 3\rho_0^2 + \rho_0^3).
\end{aligned}$$

Аналогични оценки са валидни и за правите грешки в другите елементи $|\delta u_i| \leq |a_{ik}q_{ik-1}|\rho_0, |\delta w_i| \leq |c_{ik}p_{ik+1}|\rho_0$ (това са точно грешките от едно умножение). Имайки предвид израза (1.5), по който се пресмята S и горните оценки за елементите на матрицата δS, получаваме

$$|\delta S| \leq (|A_{22}| + |A_{21}||\hat{R}|)g(\rho_0), \tag{1.14}$$

където $g(\rho_0) = 3\rho_0 + 3\rho_0^2 + \rho_0^3$, при което е очевидно че $g(\rho_0) \leq f(\rho_0)$. Тогава от (1.9), (1.13) и (1.14) стигаме до търсената оценка

$$\begin{aligned}
|E| &\leq \begin{pmatrix} 0 & k_1|A_{11}||\hat{R}|f(\rho_0) \\ 0 & (|A_{22}| + |A_{21}||\hat{R}|)g(\rho_0) \end{pmatrix} \\
&\leq K_1 \begin{pmatrix} |A_{11}| & 0 \\ |A_{21}| & |A_{22}| \end{pmatrix} \begin{pmatrix} 0 & |\hat{R}| \\ 0 & I_{s-1} \end{pmatrix} f(\rho_0) \\
&\leq K_1|\mathcal{A}| \begin{pmatrix} 0 & |\hat{R}| \\ 0 & I_{s-1} \end{pmatrix} f(\rho_0) \\
&= K_1|\mathcal{A}||N|f(\rho_0),
\end{aligned}$$

където $K_1 = \max\{k_1, 1\}$ и $N = \begin{pmatrix} 0 & \hat{R} \\ 0 & I_{s-1} \end{pmatrix}$. ◊

След като получихме оценка за грешката, допусната при блочната LU факторизация на $\mathcal{A}$, нека понататък да анализираме и грешката, която се допуска при намиране на решението на системата с двете блочно триъгълни матрици L и $\hat{U}$.

Лема 1.2 *При решаване на система с долнотриъгълната матрица L е в сила $(L + \Delta L)\hat{y} = d$, където за матрицата от еквивалентни смущения ΔL е валидна оценката*

$$|\Delta L| \leq K_1|L|f(\rho_0).$$

Доказателство. Имайки предвид вида на L (виж (1.3)), то за матрицата от еквивалентни смущения ΔL получаваме

$$|\Delta L| = \begin{pmatrix} |\Delta A_{11}| & 0 \\ |\Delta A_{21}| & |\Delta I_{s-1}| \end{pmatrix}, \tag{1.15}$$

където ΔA_{11} е смущението от Етап 1 (решаване на тридиагонална система), $\Delta A_{21}, \Delta I_{s-1}$ са смущения, идващи от изключването на A_{21} (това очевидно е еквивалентно на решаване на триъгълна система). За блочно диагоналната матрица A_{11} с тридиагонални блокове, прилагайки (1.8), получаваме

$$|\Delta A_{11}| \leq k_1|A_{11}|f(\rho_0). \tag{1.16}$$

След това, за да решим получената триъгълна система, е необходимо да направим аритметични операции от следния вид: $l = (\alpha - \beta\mu - \gamma\nu)/\lambda$, където α е даден елемент от дясната част, β, γ, λ са дадени матрични елементи, а μ, ν са вече дефинирани елементи на решението на системата $Ly = d$. Извършвайки прав анализ на грешката, получаваме, че

$$\begin{aligned} \hat{l} &= \{\alpha - [\beta\mu(1+\sigma_1) + \gamma\nu(1+\sigma_2)](1+\sigma_3)\}(1+\sigma_4)/\lambda \\ &= \frac{\alpha - [\beta\mu(1+\sigma_1) + \gamma\nu(1+\sigma_2)](1+\sigma_3)}{\lambda/(1+\sigma_4)}. \end{aligned}$$

Елементът λ е диагонален елемент от L, или по-точно от $I_{s(k-1)}$ т.е. $\lambda = 1$, следователно когато делим на него, няма да се появи грешка от закръгляване. Въпреки това обаче въвеждаме λ с цел да избегнем смущения в дясната част (тези смущения отиват в λ). Така стигаме до

$$\hat{l} = \frac{\alpha - \beta(1+\sigma_1)(1+\sigma_3)\mu - \gamma(1+\sigma_2)(1+\sigma_3)\nu}{\lambda(1+\xi)}, \tag{1.17}$$

където

$$\xi = \frac{\sigma_4}{1+\sigma_4}. \tag{1.18}$$

От (1.17) и (1.18) следва, че

$$|\Delta A_{21}| \leq (2\rho_0 + \rho_0^2)|A_{21}|, \tag{1.19}$$

$$|\Delta I_{s-1}| \leq |\xi| I_{s-1} \leq 2\rho_0 I_{s-1}. \tag{1.20}$$

След като получихме оценки за всички ненулеви елементи на матрицата ΔL, то от (1.15), (1.16), (1.19) и (1.20) стигаме до твърдението на лемата

$$|\Delta L| \leq K_1 |L| f(\rho_0). \qquad \Diamond$$

Лема 1.3 *При решаване на система с горнотриъгълната матрица $\hat{U}$ е в сила $(\hat{U} + \Delta\hat{U})\hat{x} = \hat{y}$, при което за матрицата от еквивалентни смущения $\Delta\hat{U}$ е валидна оценката*

$$|\Delta\hat{U}| \leq K_2 |\hat{U}| f(\rho_0), \quad K_2 = \max\{k_2, 1\},$$

където k_2 ограничава ръста на елементите при прилагане на Гаусово изключване за намиране на решение на редуцираната система (Етап 3 на алгоритъма).

Доказателство. Доказателството е почти същото, като това направено в Лема 1.2. $\Diamond$

Преди да представим основния резултат, относно устойчивостта на целия алгоритъм, ще направим някои уточнения. Нека да разгледаме $j-$та компонента на произведението $|N||\mathcal{P}\hat{x}|, j = 1, 2, \ldots, s(k-1)$. Имайки предвид вида на матрицата R (виж (1.6)) и по-точно факта, че във всеки ред на тази матрица има само два ненулеви елемента, то за тази $j-$та компонента е изпълнено

$$\begin{aligned}(|N||\mathcal{P}\hat{x}|)_j &\leq |\hat{R}_{j,i}||\hat{x}_{ik}| + |\hat{R}_{j,i+1}||\hat{x}_{(i+1)k}| \\ &\leq \|\hat{R}\|_\infty \max\left\{|\hat{x}_{ik}|, |\hat{x}_{(i+1)k}|\right\}, \end{aligned} \tag{1.21}$$

за някое i. Нека да дефинираме вектор $\mathcal{P}x^*$ по следния начин:

$$\mathcal{P}x^* = [(x_1^*)^T \ (x_2^*)^T]^T,$$

където

$$x_1^* = (|\hat{x}_k|e, \max\{|\hat{x}_k|, |\hat{x}_{2k}|\}\, e, \ldots, \max\left\{|\hat{x}_{(s-2)k}|, |\hat{x}_{(s-1)k}|\right\}\, e)^T,$$

$$x_2^* = (|\hat{x}_k|, \ldots, |\hat{x}_{(s-1)k}|)^T,$$

$e = (1, 1, \ldots, 1) \in \mathcal{R}^{1\times(k-1)}$. Тогава можем да определим и вектора x^*

$$x^* = (|\hat{x}_k|e, |\hat{x}_k|, \max\{|\hat{x}_k|, |\hat{x}_{2k}|\}\, e, \ldots, |\hat{x}_{(s-1)k}|, \max\left\{|\hat{x}_{(s-2)k}|, |\hat{x}_{(s-1)k}|\right\}\, e)^T.$$

Изхождайки от (1.21) и от вида на x^* получаваме

$$|N||\mathcal{P}\hat{x}| \leq \begin{pmatrix} \|R\|_\infty x_1^* \\ x_2^* \end{pmatrix} \leq r\mathcal{P}x^*, \tag{1.22}$$

където $r = \max\left\{\|\hat{R}\|_\infty, 1\right\}$.

Сега използвайки така дефинирания вектор x^*, можем да въведем следното число на обусловеност:

$$cond^*(\mathcal{A}, \mathcal{P}x^*) = \frac{\| \ |\mathcal{A}^{-1}| \ |\mathcal{A}| \ \mathcal{P}x^* \ \|_\infty}{\|\mathcal{P}\hat{x}\|_\infty}.$$

Лесно може да се провери че

$$cond^*(\mathcal{A}, x^*) \leq cond(\mathcal{A}),$$

където $cond(\mathcal{A}) = \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty$ (виж [36] за такива числа на обусловеност). Нека още да видим и каква е връзката между обусловеността на оригиналната матрица A и пермутираната $\mathcal{A}$:

$$\begin{aligned} cond(\mathcal{A}) &= \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty = \|\mathcal{P}|A^{-1}|\mathcal{P}^T\mathcal{P}|A|\mathcal{P}^T\|_\infty \\ &= \|\mathcal{P}|A^{-1}||A|\mathcal{P}^T\|_\infty = \||A^{-1}||A|\|_\infty = cond(A), \end{aligned} \tag{1.23}$$

и по същия начин

$$cond^*(\mathcal{A}, \mathcal{P}x^*) = cond^*(A, x^*). \tag{1.24}$$

Понататък ще използваме също и така нареченото число на обусловеност на Скийл (виж [56])

$$cond(\mathcal{A}, \mathcal{P}\hat{x}) = \frac{\| \, |\mathcal{A}^{-1}| \, |\mathcal{A}| \, |\mathcal{P}\hat{x}| \, \|_\infty}{\|\hat{x}\|_\infty},$$

като в знаменателя участва полученото изчислено решение $\hat{x}$, вместо точното x. Това дава възможност да бъдат получени по-реалистични оценки, както и да се намерят изчислими горни граници за правата грешка в решението на системата. По същия начин както в (1.23) и (1.24) може да се покаже, че

$$cond(\mathcal{A}, \mathcal{P}\hat{x}) = cond(A, \hat{x}). \tag{1.25}$$

Естествен е въпросът за това, какъв е смисълът на така дефинираното число на обусловеност $cond^*(A, x^*)$? Както ще видим понататък в изложението, това число дава възможност да се получат по-гъвкави оценки. Освен това в изведената оценка за правата грешка (1.33) числото на обусловеност $cond^*(A, x^*)$ се умножава с въведения множител r (той понякога може да бъде твърде голям), докато числото на обусловеност $cond(A, \hat{x})$ не се умножава. При това се оказва, че когато $cond^*(A, x^*)$ е малко, влиянието на r може да се пренебрегне, т.е. не е съществено. Пример, който потвърждава това, е представен в Раздел 1.5.

В следващата теорема, използвайки Леми 1.1 - 1.3, ще представим основния резултат в този раздел.

Теорема 1.1 *При реализация на алгоритъма на Уанг за решаване на системата (1.1) е в сила* $(\mathcal{A} + \Delta\mathcal{A})\mathcal{P}\hat{x} = \mathcal{P}d$, *при което*

$$|\Delta\mathcal{A}| \leq |\mathcal{A}| \left[(K_1 + K_2) f(\rho_0) + h_1(\rho_0)\right] + |\mathcal{A}||N| \left[(3K_1 + 2K_2) f(\rho_0) + h_2(\rho_0)\right],$$

а

$$\begin{aligned} h_1(\rho_0) &= (K_1 + K_2) f(\rho_0) g(\rho_0) + K_1 K_2 f^2(\rho_0) + K_1 K_2 f^2(\rho_0) g(\rho_0), \\ h_2(\rho_0) &= (K_1 + K_2) f(\rho_0) g(\rho_0) + 2K_1 K_2 f^2(\rho_0) + K_1 K_2 f^2(\rho_0) g(\rho_0), \end{aligned}$$

са членове, съдържащи ρ_0 *само във втори и по-висок порядък. Освен това за правата грешка в решението на системата е изпълнено*

$$\begin{aligned} \frac{\|\delta x\|_\infty}{\|\hat{x}\|_\infty} &= \frac{\|\hat{x} - x\|_\infty}{\|\hat{x}\|_\infty} \\ &\leq cond(A, \hat{x}) \left[(K_1 + K_2) f(\rho_0) + h_1(\rho_0)\right] \\ &\quad + cond^*(A, x^*) r \left[(3K_1 + 2K_2) f(\rho_0) + h_2(\rho_0)\right]. \end{aligned}$$

Доказателство. За изчисленото решение имаме, че

$$(L+\Delta L)(\hat{U}+\Delta\hat{U})\mathcal{P}\hat{x}=\mathcal{P}d,$$

тогава

$$(L\hat{U}+\Delta L\hat{U}+L\Delta\hat{U}+\Delta L\Delta\hat{U})\mathcal{P}\hat{x}=\mathcal{P}d,$$

и от факта, че $L\hat{U}=\mathcal{A}+E$ стигаме до

$$(\mathcal{A}+E+\Delta L\hat{U}+L\Delta\hat{U}+\Delta L\Delta\hat{U})\mathcal{P}\hat{x}=\mathcal{P}d.$$

Следователно в представянето $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x}=\mathcal{P}d$ ще имаме

$$|\Delta\mathcal{A}|\leq|E|+|\Delta L||\hat{U}|+|L||\Delta\hat{U}|+|\Delta L||\Delta\hat{U}|. \tag{1.26}$$

От Леми 1.1 - 1.3 и (1.26) получаваме, че

$$\begin{aligned}|\Delta\mathcal{A}| &\leq K_1|\mathcal{A}||N|f(\rho_0)+K_1|L||\hat{U}|f(\rho_0)\\ &\quad +K_2|L||\hat{U}|f(\rho_0)+K_1K_2|L||\hat{U}|f^2(\rho_0)\\ &= K_1|\mathcal{A}||N|f(\rho_0)+\left(K_1f(\rho_0)+K_2f(\rho_0)+K_1K_2f^2(\rho_0)\right)|L||\hat{U}|,\end{aligned} \tag{1.27}$$

но

$$\begin{aligned}|L||\hat{U}| &= \begin{pmatrix}|A_{11}| & 0\\ |A_{21}| & I_{s-1}\end{pmatrix}\begin{pmatrix}I_{s(k-1)} & |\hat{R}|\\ 0 & |\hat{S}|\end{pmatrix}\\ &= \begin{pmatrix}|A_{11}| & |A_{11}||\hat{R}|\\ |A_{21}| & |A_{21}||\hat{R}|+|\hat{S}|\end{pmatrix}\\ &\leq \begin{pmatrix}|A_{11}| & |A_{11}||\hat{R}|+|A_{12}|\\ |A_{21}| & |A_{22}|+2|A_{21}||\hat{R}|+|\delta S|\end{pmatrix},\end{aligned} \tag{1.28}$$

където сме използвали факта че $\hat{S}=A_{22}-A_{21}\hat{R}+\delta S$. Ако заместим (1.14) в (1.28) и направим някои преобразувания стигаме до

$$\begin{aligned}|L||\hat{U}| &\leq \begin{pmatrix}|A_{11}| & |A_{11}||\hat{R}|+|A_{12}|\\ |A_{21}| & |A_{22}|(1+g(\rho_0))+|A_{21}||\hat{R}|(2+g(\rho_0))\end{pmatrix}\\ &\leq |\mathcal{A}|(1+g(\rho_0))+\begin{pmatrix}0 & |A_{11}||\hat{R}|\\ 0 & |A_{21}||\hat{R}|\end{pmatrix}(2+g(\rho_0))\\ &\leq |\mathcal{A}|(1+g(\rho_0))+|\mathcal{A}|\begin{pmatrix}0 & |\hat{R}|\\ 0 & I_{s-1}\end{pmatrix}(2+g(\rho_0))\\ &= |\mathcal{A}|(1+g(\rho_0))+|\mathcal{A}||N|(2+g(\rho_0)).\end{aligned} \tag{1.29}$$

Сега от (1.27) и (1.29) получаваме следната оценка за матрицата от еквивалентни смущения $\Delta\mathcal{A}$ при компютърна реализация на целия алгоритъм:

$$\begin{aligned}|\Delta\mathcal{A}| &\leq K_1|\mathcal{A}||N|f(\rho_0)+\left(K_1f(\rho_0)+K_2f(\rho_0)+K_1K_2f^2(\rho_0)\right)\\ &\quad [|\mathcal{A}|(1+g(\rho_0))+|\mathcal{A}||N|(2+g(\rho_0))]\\ &= |\mathcal{A}|\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right]\\ &\quad +|\mathcal{A}||N|\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right],\end{aligned} \tag{1.30}$$

където

$$\begin{aligned} h_1(\rho_0) &= (K_1+K_2)f(\rho_0)g(\rho_0)+K_1K_2f^2(\rho_0)+K_1K_2f^2(\rho_0)g(\rho_0), \\ h_2(\rho_0) &= (K_1+K_2)f(\rho_0)g(\rho_0)+2K_1K_2f^2(\rho_0)+K_1K_2f^2(\rho_0)g(\rho_0). \end{aligned}$$

Тук сме отделили членовете, съдържащи ρ_0 във втори и по-висок порядък, във функциите h_1 и h_2. Това дава възможност лесно да получим приближение от първи порядък, пренебргвайки нелинейните членове. Освен това очевидно е, че h_1 и h_2 са малки, когато няма значителен ръст на K_1 и K_2.

Остава да оценим правата грешка $\mathcal{P}(\hat{x}-x)$. От представянето $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x}=\mathcal{P}d$ и (1.2) лесно получаваме израз, който я съдържа

$$\mathcal{P}(\hat{x}-x) = -\mathcal{A}^{-1}\Delta\mathcal{A}\mathcal{P}\hat{x}. \tag{1.31}$$

Тогава от (1.30) и (1.31) получаваме

$$\begin{aligned} |\mathcal{P}(\hat{x}-x)| \;\leq\; & |\mathcal{A}^{-1}||\mathcal{A}||\mathcal{P}\hat{x}|\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right] \\ & +|\mathcal{A}^{-1}||\mathcal{A}||N||\mathcal{P}\hat{x}|\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right]. \end{aligned} \tag{1.32}$$

Накрая от (1.22) и (1.32) стигаме до

$$\begin{aligned} \frac{\|\delta x\|_\infty}{\|\hat{x}\|_\infty} = \frac{\|\mathcal{P}(\hat{x}-x)\|_\infty}{\|\mathcal{P}\hat{x}\|_\infty} \;\leq\; & \left\{\||\mathcal{A}^{-1}||\mathcal{A}||\mathcal{P}\hat{x}|\|_\infty\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right]\right. \\ & \left.+\||\mathcal{A}^{-1}||\mathcal{A}|\mathcal{P}x^*\|_\infty r\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right]\right\}/\|\hat{x}\|_\infty \\ = \; & cond(\mathcal{A},\mathcal{P}\hat{x})\left[(K_1+K_2)f(\rho_0)+h_1(\rho_0)\right] \\ & +cond^*(\mathcal{A},\mathcal{P}x^*)r\left[(3K_1+2K_2)f(\rho_0)+h_2(\rho_0)\right]. \end{aligned} \tag{1.33}$$

Остава да вземем под внимание (1.24) и (1.25), откъдето получаваме и окончателната оценка в относителен смисъл за грешката в решението на системата. $\Diamond$

Получените оценки (1.30) и (1.33) показват, че грешката в решението на системата зависи от нейната обусловеност и от устойчивостта на алгоритъма (имат се предвид константите K_1, K_2 и r). Удовлетворителни ограничения за тези константи могат да бъдат намерени ако матрицата A принадлежи към някои от следните специални класове от матрици: матрици с диагонално преобладаване по редове или стълбове, симетрични и положително определени матрици, M-матрици и тотално неотрицателни матрици. Известно е [35], че за посочените класове от тридиагонални матрици методът на Гаус е числено устойчив. По-точно теоретично изведените в [35] стойности за константата, ограничаваща ръста на елементите при решаване на една тридиагонална система по метода на Гаус са: за матрици с диагонално преобладаване по редове (стълбове) е равна на

3 (т.е. $K = 3$, виж (1.8)) и равна на 1 ($K = 1$) за останалите специални класове.

Следователно, щом A принадлежи към някои от посочените специални класове, ограничение за K_1 е известно. Ако успеем да покажем, че редуцираната матрица S запазва свойствата на A, то това атоматично ще означава, че за K_2 ще е валидно същото ограничение. Върху този проблем, както и върху оценяване на $\|\hat{R}\|_\infty$, за посочените специални класове е посветен следващият раздел.

Нека също да отбележим и че изведените в следващия раздел оценки са горни граници за члена $\|\hat{R}\|_\infty$ и че не е трудно да изчислим $\|\hat{R}\|_\infty$ извършвайки относително малък брой стъпки. По-точно, необходими са $k-1$ паралелни събирания и $k+s$ паралелни логически проверки. Ясно е, че това не би увеличило значително общото време за изчисления, тъй като са необходими общо $17k + 8s - 41$ паралелни стъпки (без да вземаме предвид времето за комуникации) за целия алгоритъм.

1.3 Специални класове от матрици

Вече стана ясно, че в този раздел ще отделим специално внимание на случая, когато матрицата A принадлежи към някои от следните специални класове: матрици с диагонално преобладаване по редове, симетрични и положително определени матрици, M-матрици или тотално неотрицателни матрици. Нека да отбележим, че една матрица $A \in \mathcal{R}^{n\times n}$ се нарича *неособена M-матрица* (виж [18, стр.133]) ако за нейните елементи е изпълнено $a_{ij} \leq 0$ за всички $i \neq j$ и $A^{-1} \geq 0$. Неособената матрица A пък се нарича *тотално неотрицателна* (виж [18, стр.57]) ако всички нейни минори от произволен ред са неотрицателни. Едно еквивалентно условие (доказано в [59]) е, че обратната на тотално неотрицателна матрица е знаково регулярна матрица, т.е. изпълнени са едновременно следните две изисквания:

(i) $\bar{a}_{ij} = (-1)^{i+j} a^*_{ij}$, където $\bar{a}_{ij}$ са елементите на обратната матрица A^{-1}.

(ii) Матрицата $A^* = \{a^*_{ij}\}_{i,j=1}^n = |A^{-1}|$ е тотално неотрицателна.

В този раздел класът от матрици с диагонално преобладаване ще разгледаме в смисъл на диагонално преобладаване по редове, т.е. $|a_i| + |c_i| \leq |b_i|, \quad i = 1, 2, \ldots, n$. Дефиницията за симетрична и положително определена матрица е добре известна (виж [5]). За тези класове ще докажем, че

константите K_2 и r са достатъчно малки и следователно числената устойчивост на алгоритъма ще зависи главно от обусловеността на решаваната тридиагонална система.

Нека да направим важното уточнение, че щом оригиналната матрица A е симетрична и положително определена, с диагонално преобладаване по редове или M-матрица, то и пермутираната матрица $\mathcal{A}$ е от същия тип (не е трудно да се покаже). Но това не важи за случая на тотално неотрицателни матрици, тъй като очевидно разместванията на редовете и стълбовете на една такава матрица биха променили знаците на някои нейни минори по такъв начин, че $\mathcal{A}$ да не удовлетворява дефиницията за тотална неотрицателност. На практика обаче ние ще се интересуваме не от тоталната неотрицателност на $\mathcal{A}$, а от знаковата регулярност на част от $\mathcal{A}^{-1}$ и както ще видим понататък в този раздел това е напълно достатъчно за нашите цели.

За да намерим оценки на $\|\hat{R}\|_\infty$ и k_2, е необходимо да покажем че редуцираната матрица S запазва свойствата на изходната матрица A. Това ще направим най-напред в точна аритметика, т.е. ще считаме, че редуцираната матрица S е пресметната точно. След като покажем, че S запазва свойствата на A, това ще ни даде възможност да използваме известните оценки ([35]) за ръста на елементите при решаване по метода на Гаус на тридиагонални системи от посочените специални видове. Така ръстът на константата k_2 ще бъде ограничен. Що се отнася до $\|\hat{R}\|_\infty$, то при точно пресметната S, ще намерим оценки за този член, зависещи от обусловеността на A. Накрая естествено ще изведем и оценка за общата грешка в матрицата S, при което ще покажем, че тази грешка зависи отново главно от обусловеността на матрицата A. Следователно, щом A е добре обусловена, компютърно пресметнатата редуцирана матрица ще бъде достатъчно близка до точната и всички наши разсъждения ще запазят своята валидност.

Нека да отбележим, че следващата теорема е валидна не само за тридиагонални матрици, но и за произволни плътни матрици (виж [14]).

Теорема 1.2 *Нека $A \in \mathcal{R}^{n\times n}$, тогава ако A принадлежи на един от следните класове:*

(a) симетрични и положително определени матрици,

(b) неособени M-матрици,

то редуцираната матрица S е матрица от същия клас.

Доказателство. Свойство (а) е доказано в [14, стр. 94], а второто свойство е доказано в [14, стр. 209]. ◊

Доказателство в случая, когато A е матрица с диагонално пробладаване по редове, е представено в [10]. За съжаление обаче в него има някои неточности. Ето защо тук ние ще представим алтернативно доказателство, по-голямата част от което ще използваме и при доказателството на Теорема 1.6, където се извеждат оценки за $\|\hat{R}\|_\infty$.

Теорема 1.3 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена тридиагонална матрица, и нека е с диагонално преобладаване по редове. Тогава редуцираната матрица S е матрица от същия тип.*

Доказателство. Щом A е матрица с диагонално пробладаване по редове, то за нейните елементи, както вече отбелязахме, е изпълнено

$$|a_i| + |c_i| \leq |b_i|, \tag{1.34}$$

за всяко $i = 1, 2, \ldots, n$. Нека да разгледаме матрицата $\mathcal{B}_i = (B_i, \bar{a}_i, \bar{c}_i)$, $i = 1, 2, \ldots, s$ и да покажем, че тя запазва свойството на диагонално преобладаване по редове и след прилагане на Гаусово изключване за обръщане на матрицата B_i. Нека, в резултат на правия ход на метода на Гаус, сме получили следната матрица:

$$\mathcal{B}_i^{(1)} = (B_i^{(1)}, \bar{a}_i^{(1)}, \bar{c}_i^{(1)}),$$

където $\bar{a}_i^{(1)} = (a_{i,1}^{(1)}, a_{i,2}^{(1)}, \ldots, a_{i,k-1}^{(1)})^T$, $\bar{c}_i^{(1)} = (0, 0, \ldots, 0, c_{i,k-1}^{(1)})^T$ и

$$B_i^{(1)} = \begin{pmatrix} 1 & c_{(i-1)k+1}^{(1)} & & & \\ & 1 & c_{(i-1)k+2}^{(1)} & & \\ & & \ddots & \ddots & \\ & & & 1 & c_{ik-2}^{(1)} \\ & & & & 1 \end{pmatrix}.$$

Новите елементи са получени по следните формули

$$\begin{aligned} a_{i,1}^{(1)} &= a_{i,1}/b_{(i-1)k+1}, \\ c_{(i-1)k+1}^{(1)} &= c_{(i-1)k+1}/b_{(i-1)k+1}, \\ b_{(i-1)k+j+1}^{(1)} &= b_{(i-1)k+j+1} - a_{(i-1)k+j+1}c_{(i-1)k+j}^{(1)}, \qquad (1.35) \\ a_{i,j+1}^{(1)} &= -a_{(i-1)k+j+1}a_{i,j}^{(1)}/b_{(i-1)k+j+1}^{(1)}, \qquad (1.36) \\ c_{(i-1)k+j+1}^{(1)} &= c_{(i-1)k+j+1}/b_{(i-1)k+j+1}^{(1)}, \qquad (1.37) \\ j &= 1, 2, \ldots, k-2. \end{aligned}$$

Всъщност, тук за удобство разглеждаме LU-факторизация, при която U е горнотриъгълна матрица с единици по главния диагонал. Доказателството е напълно аналогично в случая, когато пък L е долнотриъгълна матрица с единици по главния диагонал.

Ще извършим доказателството за запазване на диагоналното преобладаване по редове след правия ход по индукция относно j-ия ред на матрицата $\mathcal{B}_i^{(1)}$. За първия ред на тази матрица свойството очевидно е изпълнено, тъй като това е и ред на изходната матрица A. Нека да допуснем, че

$$|a_{i,j}^{(1)}| + |c_{(i-1)k+j}^{(1)}| \leq 1. \tag{1.38}$$

Ще докажем, че $|a_{i,j+1}^{(1)}| + |c_{(i-1)k+j+1}^{(1)}| \leq 1$. Използвайки формулите (1.34 - 1.38), имаме

$$\begin{aligned}
1 &= \frac{|b_{(i-1)k+j+1} - a_{(i-1)k+j+1}c_{(i-1)k+j}^{(1)}|}{|b_{(i-1)k+j+1}^{(1)}|} \\
&\geq \frac{|b_{(i-1)k+j+1}| - |a_{(i-1)k+j+1}|(1 - |a_{i,j}^{(1)}|)}{|b_{(i-1)k+j+1}^{(1)}|} \\
&= \frac{|b_{(i-1)k+j+1}| - |a_{(i-1)k+j+1}| + |a_{(i-1)k+j+1}||a_{i,j}^{(1)}|}{|b_{(i-1)k+j+1}^{(1)}|} \\
&\geq \frac{|c_{(i-1)k+j+1}|}{|b_{(i-1)k+j+1}^{(1)}|} + \frac{|a_{(i-1)k+j+1}||a_{i,j}^{(1)}|}{|b_{(i-1)k+j+1}^{(1)}|} \\
&= |c_{(i-1)k+j+1}^{(1)}| + |a_{i,j+1}^{(1)}|.
\end{aligned} \tag{1.39}$$

Нека сега да покажем, че разглежданото свойство се запазва и след извършване на обратния ход на Гаусовото изключване. В резултат на този ход получаваме матрицата

$$\mathcal{B}_i^{(2)} = (I_{k-1}, q^{(i)}, p^{(i)}),$$

където $p^{(i)} = (p_{(i-1)k+1}, p_{(i-1)k+2}, \ldots, p_{ik-1})^T$, $q^{(i)} = (q_{(i-1)k+1}, q_{(i-1)k+2}, \ldots, q_{ik-1})^T$. Тук новите елементи са получени по следните формули:

$$q_{(i-1)k+j-1} = a_{i,j-1}^{(1)} - q_{(i-1)k+j}c_{i,j-1}^{(1)}, \tag{1.40}$$

$$p_{(i-1)k+j-1} = -p_{(i-1)k+j}c_{i,j-1}^{(1)}, \tag{1.41}$$

$$j = k-1, k-2, \ldots, 2.$$

Доказателството отново ще извършим по индукция относно j-ия ред този път на матрицата $\mathcal{B}_i^{(2)}$. Свойството е удовлетворено за последния ред на

тази матрица, тъй като това беше доказано още при правия ход. Нека да допуснем, че

$$|q_{(i-1)k+j}| + |p_{(i-1)k+j}| \leq 1. \tag{1.42}$$

Тогава от (1.39), (1.40), (1.41) и (1.42) ще имаме, че

$$\begin{aligned} |q_{(i-1)k+j-1}| &\leq |a^{(1)}_{i,j-1}| + |q_{(i-1)k+j}||c^{(1)}_{i,j-1}| \\ &\leq |a^{(1)}_{i,j-1}| + (1 - |p_{(i-1)k+j}|)|c^{(1)}_{i,j-1}| \\ &= |a^{(1)}_{i,j-1}| + |c^{(1)}_{i,j-1}| - |p_{(i-1)k+j}||c^{(1)}_{i,j-1}| \\ &\leq |a^{(1)}_{i,j-1}| + |c^{(1)}_{i,j-1}| - |p_{(i-1)k+j-1}|. \end{aligned}$$

Следователно,

$$|q_{(i-1)k+j-1}| + |p_{(i-1)k+j-1}| \leq |a^{(1)}_{i,j-1}| + |c^{(1)}_{i,j-1}| \leq 1. \tag{1.43}$$

Това означава, че свойството на диагонално преобладаване по редове се запазва и след обратния ход. От този факт и от (1.7) за елементите на S е изпълнено

$$\begin{aligned} |v_i| &\geq |b_{ik}| - |a_{ik}||p_{ik-1}| - |c_{ik}||q_{ik+1}| \\ &\geq |b_{ik}| - |a_{ik}|(1 - |q_{ik-1}|) - |c_{ik}|(1 - |p_{ik+1}|) \\ &\geq |c_{ik}| + |a_{ik}||q_{ik-1}| - |c_{ik}| + |c_{ik}||p_{ik+1}| \\ &\geq |u_i| + |w_i|. \end{aligned}$$

Следователно редуцираната матрица S е също матрица с диагонално преобладаване по редове. $\Diamond$

Нека сега да разгледаме случая, когато A е тотално неотрицателна. Ще представим доказателство на този случай, тъй като липсва в съществуващата литература, касаеща разглежданата проблематика.

Теорема 1.4 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена тридиагонална матрица. Тогава, ако матрицата A е тотално неотрицателна, то редуцираната матрица S е*

- *M-матрица, когато блоковете B_i са от нечетен ред, т.е. k е четно,*
- *тотално неотрицателна, когато блоковете B_i са от четен ред, т.е. k е нечетно.*

Доказателство. Имайки предвид блочното представяне (1.3) и от факта че A е тотално неотрицателна матрица следва че A_{11}, A_{22} са матрици от същия тип, A_{11}^{-1} е знаково регулярна матрица, а $A_{12} \geq 0, A_{21} \geq 0$. Нека

да намерим обратната матрица на $\mathcal{A}$. Изхождайки от (1.3), получаваме

$$\begin{aligned}\mathcal{A}^{-1} = U^{-1}L^{-1} &= \begin{pmatrix} I_{s(k-1)} & -RS^{-1} \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} A_{11}^{-1} & 0 \\ -A_{21}A_{11}^{-1} & I_{s-1} \end{pmatrix} \\ &= \begin{pmatrix} A_{11}^{-1} + H & -RS^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{pmatrix}, \end{aligned} \tag{1.44}$$

където

$$H = RS^{-1}A_{21}A_{11}^{-1} = A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1}. \tag{1.45}$$

От (1.44) се вижда, че знаково регулярната матрица A^{-1} е пермутирана по такъв начин, че в долния десен ъгъл на $\mathcal{A}^{-1}$ се появява точно S^{-1}, състояща се от елементите на A^{-1} с индекси

$$\begin{pmatrix} k,k & k,2k & \dots & k,(s-1)k \\ 2k,k & 2k,2k & \dots & 2k,(s-1)k \\ \vdots & \vdots & \dots & \vdots \\ (s-1)k,k & (s-1)k,2k & \dots & (s-1)k,(s-1)k \end{pmatrix}.$$

Вземайки под внимание алтернативната смяна на знаците $+,-,+,\dots$, във всеки ред и стълб на матрицата A^{-1} и отчитайки направените размествания следва, че щом k е четно, тогава матрицата S^{-1} се състои само от положителни елементи (това са точно елементите с горе споменатите индекси на A^{-1}), т.е. $S^{-1} \geq 0$. Остава да покажем, че извънднагоналните елементи на S са неположителни. Нека да запишем знаците на елементите на резултата от матричното произведение $A_{21}A_{11}^{-1}A_{12}$ за един диагонален блок на A^{-1}, вземайки предвид само знаците на елементите на всяка една от матриците в произведението (някои от елементите може да са нулеви, но това не оказва съществено влияние върху нашите разсъждения):

$$\begin{pmatrix} + & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & + \end{pmatrix} \begin{pmatrix} + & - & + & \dots & + \\ - & + & - & \dots & - \\ + & - & + & \dots & + \\ \vdots & \vdots & \vdots & & \vdots \\ + & - & + & \dots & + \end{pmatrix} \begin{pmatrix} + & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & + \end{pmatrix} = \begin{pmatrix} + & + \\ + & + \end{pmatrix}. \tag{1.46}$$

От (1.46) се вижда, че извънднагоналните елементи на $A_{21}A_{11}^{-1}A_{12}$ са неотрицателни. Но $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$, където A_{22} е диагонална матрица и следователно всички извънднагонални елементи на S са неположителни. От дефиницията за неособена M-матрица следва че S е такава.

Нека сега да разгледаме случая, когато k е нечетно. Вземайки предвид отново смяната на знаците в A^{-1}, не е трудно да се види, че елементите с гореспоменатите индекси (това бяха точно елементите на S^{-1}) имат знаци, които се сменят по същия начин, както тези в A^{-1}.

Остава да отбележим, че $|S^{-1}|$ е долният десен блок на матрицата $|\mathcal{A}^{-1}|$ и че пермутационната матрица $\mathcal{P}$ е такава че всички минори на $|S^{-1}|$ са също минори и на $|A^{-1}|$. Тогава $|S^{-1}|$ е тотално неотрицателна, а S^{-1} е знаково регулярна матрица и следователно S е тотално неотрицателна матрица. $\Diamond$

Както видяхме в Теорема 1.1, оценката за правата грешка в решението зависи не само от константите K_1 и K_2, но също и от множителя r, който измерва ръста на елементите при пресмятане на $\hat{R}$. В случая обаче, когато някои от блоковете B_i са лошо обусловени (макар цялата матрица A да е добре обусловена), това би могло да доведе до значителен ръст на елементите в $\hat{R}$, което ще означава, че $\|\hat{R}\|_\infty$ става голяма, т.е. множителят r нараства. Това пък от своя страна може да доведе до големи грешки за добре обусловени матрици. Следователно, необходимо е да оценим r (респективно $\|\hat{R}\|_\infty$). Понататък ще покажем, че $\|\hat{R}\|_\infty$ е ограничена от не големи константи в случай на гореспоменатите класове от матрици. За тази цел ще ни бъде необходима следната лема:

Лема 1.4 *Ако* $k_1 cond(A_{11}) f(\rho_0) < 1$, *то е валидна следната оценка*

$$\|\hat{R}\|_\infty \leq \frac{\|R\|_\infty}{1 - k_1 cond(A_{11}) f(\rho_0)}.$$

Доказателство. За нормата $\|\hat{R}\|_\infty$ очевидно е изпълнено

$$\|\hat{R}\|_\infty \leq \|R\|_\infty + \|\delta R\|_\infty. \tag{1.47}$$

За да докажем твърдението на лемата е необходимо да получим оценка за $\|\delta R\|_\infty$. Нека да въведем следните означения:

$$|A_{11}^{-1}||A_{11}| = M = \{m_{it}\}_{i,t=1}^{s(k-1)},$$

където $M_i = i$-тия ред на M, $l = s(k-1)$. Изхождайки от (1.10) имаме, че

$$\delta y = -A_{11}^{-1} \Delta A_{11} \hat{y},$$

и ако заместим тук (1.16) получаваме

$$|\delta y^{(j)}| \leq k_1 |A_{11}^{-1}||A_{11}||\hat{y}^{(j)}| f(\rho_0), \tag{1.48}$$

където $\hat{y}^{(j)}$ е j-ият стълб на матрицата $\hat{R}$. Тогава за $\|\delta R\|_\infty$, използвайки (1.48), имаме

$$\|\delta R\|_\infty = \max_{1 \leq i \leq l} \sum_{j=1}^{s-1} |\delta y_i^{(j)}| \leq \max_{1 \leq i \leq l} \sum_{j=1}^{s-1} k_1 M_i |\hat{y}^{(j)}| f(\rho_0)$$

$$
\begin{aligned}
&= k_1 \max_{1\le i\le l} \Big[m_{i1}|\hat{y}_1^{(1)}| + \ldots + m_{il}|\hat{y}_l^{(1)}| + \ldots \\
&\quad + m_{i1}|\hat{y}_1^{(s-1)}| + \ldots + m_{il}|\hat{y}_l^{(s-1)}| \Big] f(\rho_0) \\
&= k_1 \max_{1\le i\le l} \left[m_{i1} \sum_{j=1}^{s-1} |\hat{y}_1^{(j)}| + m_{i2} \sum_{j=1}^{s-1} |\hat{y}_2^{(j)}| + \ldots \right. \\
&\quad \left. + m_{il} \sum_{j=1}^{s-1} |\hat{y}_l^{(j)}| \right] f(\rho_0). \qquad (1.49)
\end{aligned}
$$

От очевидния факт $\sum_{j=1}^{s-1} |\hat{y}_i^{(j)}| \le \|\hat{R}\|_\infty$, за всяко $1 \le i \le l$, заместваики в дясната страна на (1.49) стигаме до

$$
\begin{aligned}
\|\delta R\|_\infty &\le k_1 \|\hat{R}\|_\infty \max_{1\le i\le l} \sum_{j=1}^{l} m_{ij} f(\rho_0) \\
&= k_1 cond(A_{11}) \|\hat{R}\|_\infty f(\rho_0). \qquad (1.50)
\end{aligned}
$$

Накрая от (1.47) и (1.50) получаваме твърдението на лемата

$$\|\hat{R}\|_\infty \le \|R\|_\infty + k_1 cond(A_{11}) \|\hat{R}\|_\infty f(\rho_0),$$

или

$$\|\hat{R}\|_\infty \le \frac{\|R\|_\infty}{1 - k_1 cond(A_{11}) f(\rho_0)}. \qquad \Diamond$$

Имайки предвид току-що доказаната Лема 1.4, за да намерим оценки за $\|\hat{R}\|_\infty$, понататък ще бъде достатъчно да намерим такива само за $\|R\|_\infty$. Освен това от Лема 1.4 се вижда, че оценката за $\|\hat{R}\|_\infty$ зависи от големината на $cond(A_{11})$. Ето защо е небходимо да получим оценки и за това число на обусловеност. По-точно ще покажем също, че A_{11} е по-добре обусловена от A за разглежданите специални класове от матрици.

Теорема 1.5 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена тридиагонална матрица и нека $k_1 cond(A_{11}) f(\rho_0) < 1$. Тогава ако кое да е от следните две условие е изпълнено*

(a) A е тотално неотрицателна,

(b) A е M-матрица,

то

$$\|\hat{R}\|_\infty \le \frac{cond(A)}{1 - k_1 cond(A_{11}) f(\rho_0)} \le \frac{cond(A)}{1 - k_1 cond(A) f(\rho_0)}.$$

Доказателство. (а) Нека A е тотално неотрицателна матрица. Тогава от дефиницията за такава матрица ще имаме, че $A_{12} \geq 0, A_{21} \geq 0$, а A_{11}^{-1} е знаково регулярна матрица. В Теорема 1.4 показахме, че щом блоковете B_i са от нечетен ред, то редуцираната матрица S е M-матрица, т.е. $S^{-1} \geq 0$. И щом блоковете B_i са от четен ред, то редуцираната матрица S е тотално неотрицателна, т.е. S^{-1} е знаково регулярна. За да можем да оценим $\|R\|_\infty$, ще ни бъде необходимо да покажем, че блочно диагоналните елементи $B_i^{-1} + H_{ii}$ на матрицата $A_{11}^{-1} + H$ удовлетворяват първото изискване от дефиницията за знакова регулярност, направена в началото на раздела. Както вече отбелязахме матрицата, A_{11}^{-1} е знаково регулярна и следователно нейните блочно диагонални елементи B_i^{-1} са също знаково регулярни. Остава да покажем, че блочно диагоналните елементи H_{ii} на матрицата H удовлетворяват първото изискване за знакова регулярност. За тази цел е необходимо да разгледаме два случая.

Първият е, когато B_i е от нечетен ред. Нека да запишем в знакова форма i-ия диагонален блок на резултата от матричното произведение $A_{12}S^{-1}A_{21}$, вземайки предвид само знаците на елементите на всяка една от матриците в произведението (отчитаме и нулевите елементи)

$$\begin{pmatrix} + & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & + \end{pmatrix} \begin{pmatrix} + & + \\ + & + \end{pmatrix} \begin{pmatrix} + & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & + \end{pmatrix} = \begin{pmatrix} + & 0 & \dots & 0 & + \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ + & 0 & \dots & 0 & + \end{pmatrix}. \quad (1.51)$$

Сега от (1.45) и (1.51) за H_{ii} получаваме:

$$\begin{aligned} H_{ii} &= \begin{pmatrix} + & - & + & \dots & + \\ - & + & - & \dots & - \\ + & - & + & \dots & + \\ \vdots & \vdots & \vdots & & \vdots \\ + & - & + & \dots & + \end{pmatrix} \begin{pmatrix} + & 0 & \dots & 0 & + \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ + & 0 & \dots & 0 & + \end{pmatrix} \begin{pmatrix} + & - & + & \dots & + \\ - & + & - & \dots & - \\ + & - & + & \dots & + \\ \vdots & \vdots & \vdots & & \vdots \\ + & - & + & \dots & + \end{pmatrix} \\ &= \begin{pmatrix} + & 0 & \dots & 0 & + \\ - & 0 & \dots & 0 & - \\ \vdots & \vdots & & \vdots & \vdots \\ - & 0 & \dots & 0 & - \\ + & 0 & \dots & 0 & + \end{pmatrix} \begin{pmatrix} + & - & + & \dots & + \\ - & + & - & \dots & - \\ + & - & + & \dots & + \\ \vdots & \vdots & \vdots & & \vdots \\ + & - & + & \dots & + \end{pmatrix} = \begin{pmatrix} + & - & + & \dots & + \\ - & + & - & \dots & - \\ + & - & + & \dots & + \\ \vdots & \vdots & \vdots & & \vdots \\ + & - & + & \dots & + \end{pmatrix}. \end{aligned}$$

Така получихме, че действително H_{ii} удовлетворяват първото изискване за знакова регулярност.

Вторият случай е, когато B_i е от четен ред. Нека по аналогичен начин да определим отново знаците на i-ия диагонален блок на $A_{12}S^{-1}A_{21}$

$$\begin{pmatrix} + & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & + \end{pmatrix} \begin{pmatrix} + & - \\ - & + \end{pmatrix} \begin{pmatrix} + & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & + \end{pmatrix} = \begin{pmatrix} + & 0 & \dots & 0 & - \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ - & 0 & \dots & 0 & + \end{pmatrix}. \quad (1.52)$$

Тогава от (1.45) и (1.52) получаваме:

$$H_{ii} = \begin{pmatrix} + & - & + & \dots & - \\ - & + & - & \dots & + \\ + & - & + & \dots & - \\ \vdots & \vdots & \vdots & & \vdots \\ - & + & - & \dots & + \end{pmatrix} \begin{pmatrix} + & 0 & \dots & 0 & - \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ - & 0 & \dots & 0 & + \end{pmatrix} \begin{pmatrix} + & - & + & \dots & - \\ - & + & - & \dots & + \\ + & - & + & \dots & - \\ \vdots & \vdots & \vdots & & \vdots \\ - & + & - & \dots & + \end{pmatrix}$$

$$= \begin{pmatrix} + & 0 & \dots & 0 & - \\ - & 0 & \dots & 0 & + \\ \vdots & \vdots & & \vdots & \vdots \\ + & 0 & \dots & 0 & - \\ - & 0 & \dots & 0 & + \end{pmatrix} \begin{pmatrix} + & - & + & \dots & - \\ - & + & - & \dots & + \\ + & - & + & \dots & - \\ \vdots & \vdots & \vdots & & \vdots \\ - & + & - & \dots & + \end{pmatrix} = \begin{pmatrix} + & - & + & \dots & - \\ - & + & - & \dots & + \\ + & - & + & \dots & - \\ \vdots & \vdots & \vdots & & \vdots \\ - & + & - & \dots & + \end{pmatrix}.$$

Следователно, отново получихме, че H_{ii} удовлетворяват първото изискване за знакова регулярност.

Тогава и в двата случая е изпълнено, че

$$|A_{11}^{-1} + H| = |A_{11}^{-1}| + |H|. \quad (1.53)$$

За блочно диагоналните елементи този факт следва от това, че B_i^{-1} и H_{ii} имат една и съща знакова форма, а за извъндиагоналните блочни елементи това е очевидно, тъй като тези елементи на матрицата A_{11}^{-1} са нули. Сега от (1.44)и (1.53) стигаме до следната оценка:

$$|\mathcal{A}^{-1}| = \begin{pmatrix} |B_{11}| & |B_{12}| \\ |B_{21}| & |B_{22}| \end{pmatrix} \geq \begin{pmatrix} |A_{11}^{-1} + H| & 0 \\ 0 & 0 \end{pmatrix} \geq \begin{pmatrix} |A_{11}^{-1}| & 0 \\ 0 & 0 \end{pmatrix}, \quad (1.54)$$

където блоковете B_{ij} съответстват на блочното разделяне дефинирано в (1.3), но тогава е изпълнено

$$|A_{11}^{-1}| \leq |B_{11}|, \quad (1.55)$$

и следователно

$$\begin{aligned} \|R\|_\infty &\leq \||A_{11}^{-1}||A_{12}|\|_\infty \leq \||B_{11}||A_{12}|\|_\infty \leq \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty \\ &= cond(\mathcal{A}) = cond(A). \end{aligned} \quad (1.56)$$

От Лема 1.4 и (1.56) получаваме първото неравенство в твърдението на теоремата.

За второто неравенство от (1.55) имаме, че

$$|A_{11}^{-1}||A_{11}| \leq |B_{11}||A_{11}|. \tag{1.57}$$

Вземайки норма безкрайност в (1.57), стигаме до

$$cond(A_{11}) \leq \||B_{11}||A_{11}|\|_\infty \leq cond(\mathcal{A}) = cond(A),$$

където отново сме използвали (1.23).

(b) Нека да предположим, че A е неособена M-матрица. Тогава $A_{11}^{-1} \geq 0, A_{12} \leq 0,$
$A_{21} \leq 0$. От Теорема 1.2 имаме, че S е също M-матрица, т.е. $S^{-1} \geq 0$ тогава за матрицата H е изпълнено

$$H = A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} \geq 0.$$

Това означава, че (1.53) е валидно отново и от тук нататък втората част *(b)* на теоремата се доказва по същия начин. $\Diamond$

Теорема 1.6 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена тридиагонална матрица и нека $k_1 cond(A_{11})f(\rho_0) < 1$. Тогава ако A е и матрица с диагонално преобладаване по редове, то е изпълнено*

$$\|\hat{R}\|_\infty \leq \frac{1}{1 - k_1 cond(A_{11})f(\rho_0)} \leq \frac{1}{1 - 2k_1 cond(A)f(\rho_0)}.$$

Доказателство. Както вече видяхме при доказателството на Теорема 1.3, за произволен i-ти ред на матрицата R, $i = 1, \ldots, s(k-1)$ е изпълнено (виж (1.43))

$$|p_i| + |q_i| \leq 1.$$

Тъй като p_i и q_i са единствените ненулеви елементи в i-ия ред на матрицата R то ясно е че

$$\|R\|_\infty \leq 1. \tag{1.58}$$

Тогава първото неравенство в твърдението на теоремата следва от Лема 1.4. За да докажем второто неравенство, нека да разгледаме най-напред следното матрично произведение

$$|A_{11}^{-1}||A_{11}| = |A_{11}^{-1} + H - H||A_{11}| \leq |A_{11}^{-1} + H||A_{11}| + |H||A_{11}|, \tag{1.59}$$

и нека да оценим първия член в дясната част на (1.59) по норма безкрайност

$$\||A_{11}^{-1} + H||A_{11}|\|_\infty \leq \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty = cond(\mathcal{A}), \tag{1.60}$$

За втория член в дясната част на (1.59), имайки предвид (1.44) и (1.54), получаваме

$$|H||A_{11}| = |RS^{-1}A_{21}A_{11}^{-1}||A_{11}| \leq |R||S^{-1}A_{21}A_{11}^{-1}||A_{11}| = |R||B_{21}||A_{11}|. \qquad (1.61)$$

Вземайки норма безкрайност в двете страни на (1.61) и отчитайки (1.58) получаваме

$$\||H||A_{11}|\|_\infty \leq \||B_{21}||A_{11}|\|_\infty \leq \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty = cond(\mathcal{A}) = cond(A). \qquad (1.62)$$

Сега от (1.59), (1.60) и (1.62) стигаме до

$$cond(A_{11}) \leq cond(A) + cond(A) = 2cond(A).$$

Така и второто неравенство е доказано. $\Diamond$

Забележка. Когато A е матрица с диагонално преобладаване по стълбове можем да направим следното разлагане

$$\mathcal{A} = LU = \begin{pmatrix} I_{s(k-1)} & 0 \\ A_{21}A_{11}^{-1} & I_{s-1} \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & S \end{pmatrix}.$$

Тогава не е трудно да се разгледа модификация на изследвания алгоритъм, съответстваща на това разлагане. За нейната реализация ще бъдат необходими същият брой стъпки във времето, както в разглеждания случай на диагонално преобладаване по редове. Ще бъдат изпълнени и аналогични оценки. Тази модификация по същество не се различава от разглеждания случай и ето защо няма да се спираме подробно върху нея.

В следващата теорема ще използваме факта че

$$\|\mathcal{A}^{-1}\|_2\|\mathcal{A}\|_2 = \|A^{-1}\|_2\|A\|_2,$$

във валидността на който бихме могли да се убедим по същия начин както в (1.23).

Теорема 1.7 *Нека $A \in \mathcal{R}^{n\times n}$ е тридиагонална матрица и нека*

$$k_1(k-1)cond_2(A_{11})f(\rho_0) < 1,$$

където $cond_2(A) = \|A^{-1}\|_2\|A\|_2$. Тогава, ако A е и симетрична положително определена, то

$$\|\hat{R}\|_\infty \leq \frac{\sqrt{(s-1)cond_2(A)}}{1 - k_1 cond(A_{11})f(\rho_0)} \leq \frac{\sqrt{(s-1)cond_2(A)}}{1 - k_1(k-1)cond_2(A)f(\rho_0)}.$$

Доказателство. Вземайки предвид Лема 1.4, Лема 10.12 от [36] и някои известни елементарни свойства на нормите, получаваме

$$\begin{aligned}\|\hat{R}\|_\infty &\leq \frac{\|R\|_\infty}{1-k_1 cond(A_{11})f(\rho_0)} \leq \frac{\sqrt{(s-1)}\|R\|_2}{1-k_1 cond(A_{11})f(\rho_0)} \\ &= \frac{\sqrt{(s-1)}\|A_{11}^{-1}A_{12}\|_2}{1-k_1 cond(A_{11})f(\rho_0)} \leq \frac{\sqrt{(s-1)cond_2(\mathcal{A})}}{1-k_1 cond(A_{11})f(\rho_0)} \\ &= \frac{\sqrt{(s-1)cond_2(A)}}{1-k_1 cond(A_{11})f(\rho_0)},\end{aligned}$$

което е точно първото неравенство в твърдението на теоремата.

За доказателството на второто неравенство отново можем да използваме известни релации между нормите:

$$\begin{aligned}cond(A_{11}) &= \||A_{11}^{-1}||A_{11}|\|_\infty \\ &= \max_{1\leq i\leq s}\|B_i^{-1}\|_\infty\|B_i\|_\infty \\ &\leq \max_{1\leq i\leq s}\sqrt{k-1}\|B_i^{-1}\|_2\sqrt{k-1}\|B_i\|_2 \\ &\leq (k-1)\|A_{11}^{-1}\|_2\|A_{11}\|_2 = (k-1)cond_2(A_{11}) \\ &\leq (k-1)cond_2(\mathcal{A}) = (k-1)cond_2(A). \qquad \Diamond\end{aligned}$$

Теореми 1.5 - 1.7 показват, че $\|\hat{R}\|_\infty$ е ограничена от не големи константи за разглежданите четири класа от матрици, стига матрицата A да е добре обусловена. Същото нещо се отнасяше и до константата K_2. Но както отбелязахме, това важи за точната матрица S. На практика обаче, трябва да се има предвид изчислената матрица $\hat{S}$. Ето защо е важно да се знае каква е разликата между S и $\hat{S}$. Понататък в Теорема 1.8 е даден отговор на този въпрос.

Теорема 1.8 *За общата грешка $\Omega S = \hat{S} - S$ в изчислената редуцирана матрица $\hat{S}$ е изпълнено*

$$\frac{\|\Omega S\|_\infty}{\|S\|_\infty} \leq K_1 cond(A) r f(\rho_0).$$

Доказателство. За изчислената редуцирана матрица $\hat{S}$ имаме

$$\begin{aligned}\hat{S} &= A_{22} - A_{21}\hat{R} + \delta S = A_{22} - A_{21}(R+\delta R) + \delta S \\ &= A_{22} - A_{21}R - A_{21}\delta R + \delta S = S + \Omega S,\end{aligned}$$

където сме означили общата грешка в $\hat{S}$ с $\Omega S = -A_{21}\delta R + \delta S$. Използвайки направените дефиниции (1.9) и (1.44), съответно на матриците E и L^{-1},

нека да намерим тяхното произведение

$$L^{-1}E = \begin{pmatrix} 0 & \delta R \\ 0 & \Omega S \end{pmatrix}. \tag{1.63}$$

От друга страна имаме

$$L^{-1}E = UU^{-1}L^{-1}E = U\mathcal{A}^{-1}E = \begin{pmatrix} I_{s(k-1)} & R \\ 0 & S \end{pmatrix} \mathcal{A}^{-1}E. \tag{1.64}$$

Тогава от (1.63) и (1.64) за блока в долния десен ъгъл на (1.63) ще имаме, че

$$\|\Omega S\|_\infty \leq \|\mathcal{A}^{-1}E\|_\infty \|S\|_\infty. \tag{1.65}$$

Освен това, използвайки Лема 1.1, лесно се вижда, че е изпълнено

$$|\mathcal{A}^{-1}E| \leq K_1 |\mathcal{A}^{-1}|\,|\mathcal{A}|\,|N| f(\rho_0),$$

и ако сравним двете страни по норма безкрайност и заместим (1.22) стигаме до

$$\|\mathcal{A}^{-1}E\|_\infty \leq K_1 cond(\mathcal{A}) r f(\rho_0) = K_1 cond(A) r f(\rho_0). \tag{1.66}$$

Накрая, от (1.65) и (1.66) получаваме твърдението на лемата

$$\frac{\|\Omega S\|_\infty}{\|S\|_\infty} \leq K_1 cond(A) r f(\rho_0). \qquad \Diamond$$

Следователно общата грешка в S зависи главно от обусловеността на A, т.е. щом A е добре обусловена то и компютърно пресметнатата $\hat{S}$ е достатъчно близка до S. Тогава и константата k_2 също става близка до теоретичната константа намерена в [35]. Така в резултат на представените в този раздел изследвания можем да направим извода, че щом матрицата на решаваната тридиагонална система принадлежи към някои от разглежданите специални класове от матрици, то алгоритъмът е числено устойчив. Същият резултат е получен и за метода на Гаус в [35]. Това показва, че паралелният алгоритъм на Уанг може да бъде използван безопасно в същите случаи, както и методът на Гаус, което е едно добро изключение от общото хипотетично правило, че паралелните алгоритми са по-неустойчиви от последователните за една и съща задача.

1.4 Стабилизиране на алгоритъма

Както вече отбелязахме, в този раздел ще отделим специално внимание на случая, когато някои от блоковете B_i са особени или лошо обусловени,

а матрицата A е добре обусловена. В този случай, макар и изходната матрица да е добре обусловена, алгоритъмът може да прекъсне (поради препълване при деление на нула) или да се получи взрив на грешките от закръгляване (при деление на числа близки до нула). Следователно, наложителни са някакви мерки за подобряване на числената устойчивост на алгоритъма. Именно с такава цел в [9] авторите предлагат да се използва QR разлагане на всеки блок B_i и при необходимост да се прави ново блочно разделяне на изходната матрица, така че новите блокове B_i са добре обусловени. Този подход, обаче води до съществено увеличаване броя на аритметичните операции и усложнява алгоритъма. Ето защо ще приложим друг подход, който освен че е прост за реализация е и достатъчно ефективен в конкретния случай. Той се изразява в изкуствено смущаване (при необходимост) на някои данни. По-конкретно, при решаване на (1.4) се налага да се дели на елементите $u_j^{(1)}$ за $j = 1, 2, \ldots, s(k-1)$. При това деление е достатъчно дори и един от елементите $u_j^{(1)}$ да е равен или близък до нула, за да се получи прекъсване на алгоритъма или взрив на грешките от закръгляване. В такива случаи нека да смутим изкуствено тези елементи с достатъчно малко число δ_0, така че да ги отдалечим достатъчно от нулата. Стабилизиращата стъпка можем да представим по следния начин:

```
if (|u_j^(1)| < δ_0)
   if (u_j^(1) = 0)
      u_j^(1) = δ_0;
   else
      u_j^(1) = u_j^(1) + sign(u_j^(1))δ_0;
   end
end
```

В резултат на добавяне на тази стъпка към алгоритъма, обаче се получава и допълнително смутено решение $\hat{x}$, което може да се доуточни чрез използване на стандартна процедура на итерационно уточняване (виж [30]) с малка модификация:

$$
\begin{aligned}
&x^{(0)} = \hat{x};\\
&\textbf{for } k = 1, 2, \ldots\\
&\quad r^{(k-1)} = b - Ax^{(k-1)};\\
&\quad (A + \Delta)y^{(k)} = r^{(k-1)};\\
&\quad x^{(k)} = x^{(k-1)} + y^{(k)};\\
&\textbf{end}
\end{aligned}
$$

Следователно, необходимо е да се решава няколко пъти система с матрица $A + \Delta$ и различни десни части, където с Δ сме означили матрицата от допълнително направените смущения (в това именно се изразява модификацията на използваната процедура). Естествено възниква въпросът за това колко пъти е необходимо да се решава тази система и каква би могла да бъде оптималната стойност на δ_0? Отговор на тези въпроси можем да дадем въз основа на нашия практически опит. Изводът, до който сме стигнали след редица числени експерименти е, че когато $\delta_0 = \sqrt{\rho_0} \approx 10^{-8}$ (в среда с двойна точност) смутеното решение е много близко до точното и е необходима само една стъпка на итерационно уточняване. Пример е разгледан в Раздел 1.5.

Що се отнася до сходимостта на използваната процедура, то имайки предвид изследванията направени в [72], можем да твърдим, че достатъчно условие за сходимост е условието от вида:

$$const.\ cond(A)\delta_0 < 1.$$

Използваният критерий за край е представен в следващия раздел.

1.5 Числени експерименти

Числените експерименти в този раздел са направени, използвайки програмната среда MATLAB, където машинната точност е $\rho_0 \approx 2.22\text{E}{-16}$. При компютърната реализация на алгоритъма са измервани два типа грешки:

1. Правата грешка в относителен смисъл

$$FE = \frac{\|\hat{x} - x\|_\infty}{\|\hat{x}\|_\infty},$$

където $\hat{x}$ е изчисленото решение.

2. Покомпонентната обратна грешка в елементите на ΔA отново в относителен смисъл (виж [36]) по формулата

$$BE = \max_{1 \le i \le n} \frac{(|A\hat{x} - d|)_i}{(|A||\hat{x}| + |d|)_i}.$$

ε	1E–5	1E–10	1E–15
BE	8.73E–11	1.19E–5	0.42
FE	1.74E–10	2.38E–5	2.06

Таблица 1.1: Права и обратна грешка в Пример 1.1, при $k = 6, s = 10$, за различни стойности на ε.

k	6	56	256	556
BE	1.44E–15	1.11E–16	1.66E–16	1.14E–16
FE	3.33E–15	1.99E–15	1.31E–14	1.55E–15

Таблица 1.2: Права и обратна грешка в Пример 1.2, при $s = 10$ и $\varepsilon = 1E - 16$, за различни стойности на k.

x	x_α	e	$rand$	$randn$
BE	4.67E–16	1.17E–16	1.59E–16	2.61E–16
FE	1.54E–9	2.21E–8	1.15E–7	6.95E–7
$cond(A, \hat{x})$	$3.03E+7$	$8.99E+8$	$9.55E+8$	$4.72E+8$
$cond^*(A, x^*)$	$1.97E+8$	$8.99E+8$	$1.22E+9$	$1.71E+9$
r	1	1	1	1

Таблица 1.3: Права и обратна грешка в Пример 1.3, при $k = 5, s = 3, \varepsilon = .009$, за различни точни решения. Дадени са също и съответните стойности на числата на обусловеност и на множителя r.

Да разгледаме следните примери.

Пример 1.1 Нека матрицата има вида $A = \text{tridiag}(1, b, 1)$, където $b = (\varepsilon, \ldots, \varepsilon, 2)$. Фиксирайки b по този начин тя става много добре обусловена. В качеството на точно решение да изберем $x = (1, 1, \ldots, 1)^T$. При компютърната реализация на този пример получаваме, че правата и обратна грешка нарастват значително, когато $\varepsilon \to 0$, макар че A е много

добре обусловена и във всички етапи на алгоритъма Гаусовото изключване е направено с частичен избор на главен елемент, т.е. костантите k_1 и k_2 са малки. Обяснението на това е, че $\|\hat{R}\|_\infty$ нараства неограничено, когато $\varepsilon \to 0$. Резултатите за различни стойности на ε са представени в Таблица 1.1. Основният извод от този пример е, че той потвърждава теоретично направените изводи в частта им относно константата r, ограничаваща ръста на елементите при пресмятане на $\hat{R}$. Но какво става, когато заедно с $\varepsilon \to 0$ имаме и $cond^*(A, x^*) \to 0$? Отговор на този въпрос е даден в следващия пример.

Пример 1.2 Нека A е матрицата от Пример 1.1, при което сме избрали $\varepsilon =$ 1E-16 (число по-малко от машинната точност). Следователно A отново е много добре обусловена. За да осигурим $cond^*(A, x^*) \approx 0$, нека точното решение да е

$$x = (1, \ldots, 1, 0; 1, \ldots, 1, 0; \ldots, 1, \ldots, 1, 0; 1, \ldots, 1)^T,$$

където $x_k = x_{2k} = \ldots = x_{(s-1)k} = 0$. При компютърното решаване на поставената задача отново във всички етапи на алгоритъма Гаусовото изключване е направено с частичен избор на главен елемент, т.е. k_1 и k_2 са малки. В същото време $r \approx$ 1E+16, но както можем да видим в Таблица 1.2 грешките в решението са много малки. Причината за това е, че $cond^*(A, x^*) \approx 0$, откъдето и влиянието на r не е съществено, макар че блоковете B_i са почти особени. Следователно, както и показахме, големи r не водят непременно до големи грешки.

Пример 1.3 Нека да разгледаме един пример от [26], който е използван и в [35]. Матрицата A е дефинирана по следния начин:

$$a_i = \begin{cases} -\varepsilon/h^2, & 1 \le i \le m, \\ -\varepsilon/h^2 + (0.5 - ih)/h^2, & m+1 \le i \le n, \end{cases}$$

$$c_i = \begin{cases} -\varepsilon/h^2 - (0.5 - ih)/h^2, & 1 \le i \le m, \\ -\varepsilon/h^2, & m+1 \le i \le n, \end{cases}$$

и $b_i = -a_i - c_i, i = 1, \ldots, n$, където $m = \lfloor (n+1)/2 \rfloor, h = 1/(n+1), \varepsilon > 0$. Това е една неособена с диагонално преобладаване по редове M-матрица. Имайки предвид теоретично направените изводи, трябва да очакваме, че каквито и точни решения да избираме, грешката в решението ще зависи главно от обусловеността на матрицата. Нека да изберем различни точни решения по следния начин: $x_\alpha = (1, \alpha, \alpha^2, \ldots, 10^{-5})^T$, $\alpha = 10^{-5/(n-1)}$, $e = (1, 1, \ldots, 1)^T$, 'rand' и 'randn', където последните две са генерирани от съответните фукции в MATLAB, като първата от тях дава равномерно разпределени случайни числа в интервала от нула до едно, а втората

$\varepsilon \backslash \delta_0$	$\delta_0 = 0$	$\delta_0 =$ 1E–6	$\delta_0 =$ 1E–7	$\delta_0 =$ 1E–8	$\delta_0 =$ 1E–9	$\delta_0 =$ 1E–10
$\varepsilon = 0$						
BE	∞	3.34E–15	2.22E–16	1.11E–16	6.66E–16	9.99E–16
FE	∞	1.05E–14	3.33E–15	1.22E–15	2.44E–15	9.88E–15
$\varepsilon =$ 1E–14						
BE	2.40E–3	6.22E–15	1.29E–15	3.33E–16	5.54E–16	8.99E–16
FE	1.95E–2	1.57E–14	8.26E–15	6.66E–15	1.22E–14	6.32E–14

Таблица 1.4: Права и обратна грешка в Пример 1.4, при $k = 102$ и $s = 8$, за различни стойности на ε и δ_0.

нормално разпределени случайни числа по цялата числова ос. Получените резултати са представени в Таблица 1.3. Прави впечатление, че обратната грешка BE е малка. Това се дължи на факта, че матрицата A е M-матрица с диагонално преобладаване по редове. В същото време, обаче, правата грешка FE е по-голяма. Обяснението е, че матрицата A е не така добре обусловена, както може да бъде видяно в Таблица 1.3. Освен това можем да видим, че правата грешка е почти равна на теоретичната оценка получена в Теорема 1.1, което показва, че тази оценка е достижима.

Пример 1.4 Нека изходната матрица A е същата от Пример 1.1. Както отбелязахме, тази матрица е много добре обусловена. В същото време ако размерността $k - 1$ на блоковете B_i е нечетно число, т.е. k е четно, те стават лошо обусловени. Целта сега ще бъде да покажем в този случай какви резултати могат да се получат прилагайки подхода със смущения. За да подобрим полученото решение използваме процедура на итерационно уточняване, при което итерациите спираме когато поне едно от следните две условия са изпълнят:

1. $\|Ax^{(k)} - d\|_\infty / \|d\|_\infty \leq 1000\rho_0$;
2. Броят на итерациите е > 10;

Нека отново в качеството на точно решение да изберем $x = (1, 1, \ldots, 1)^T$. Получените резултати за различни стойности на ε и δ_0 (за $\delta_0 = 0$ се получава оригиналния алгоритъм без стабилизация), при избрани $k = 102, s = 8$ са представени в Таблица 1.4. Броят на итерациите навсякъде е равен на единица. Въз основа на тези и множество други числени експерименти бихме могли да препоръчаме като оптимална стойност $\delta_0 = 10^{-8}$, при която се получават минимални права и обратна грешки и е необходима само една стъпка на итерационно уточняване.

k	S_2	E_2	S_4	E_4	S_6	E_6	S_8	E_8
100	0.04	0.02	0.48	0.12	0.66	0.11	0.72	0.09
500	0.16	0.08	1.04	0.26	1.38	0.23	1.60	0.20
1000	0.22	0.11	1.28	0.32	1.68	0.28	2.00	0.25
2000	0.28	0.14	1.44	0.36	1.92	0.32	2.32	0.29
4000	0.34	0.17	1.64	0.41	2.10	0.35	2.48	0.31
5000	0.40	0.20	1.84	0.46	2.22	0.37	2.64	0.33
6000	0.48	0.24	2.04	0.51	2.34	0.39	2.96	0.37
8000	0.54	0.27	2.28	0.57	2.70	0.45	3.52	0.44
10000	0.64	0.32	2.52	0.63	3.18	0.53	4.32	0.54
20000	0.78	0.39	2.72	0.68	3.72	0.62	4.88	0.61

Таблица 1.5: Паралелно ускорение и ефективност за стабилизирания алгоритъм на 2, 4, 6 и 8 процесора.

1.6 Паралелна реализация

В този раздел са представени и анализирани резултатите при паралелна реализация на изследвания алгоритъм, използвайки PVM (Parallel Virtual Machine).

PVM е софтуерна система, създадена и развивана в Emory University и Oak Ridge National Laboratory в САЩ. Тя предлага унифицирана схема на работа при разработването на паралелни програми, които могат да се изпълняват върху включени в локална мрежа хетерогенни компютри. Основен принцип при работа с PVM е, че главната програма, т. нар. Master стартира определен брой процеси, като се грижи за обмена на данни и синхронизацията между тях, чрез библиотека от стандартни интерфейсни процедури и функции. При реална паралелна реализация всеки един от тези процеси се изпълнява на отделен процесор, т.е. броят на процесите следва да бъде равен на броя на включените в мрежата процесори. Съществува възможност да се симулира паралелно изпълнение в случая, когато броят на процесорите е по-малък от броя на процесите. От тази гледна точка PVM е удобна, както за реална паралелна реализация, така и за подготовка за такава например на скъп паралелен компютър. Повече информация за PVM може да бъде намерена в [28].

Числените експерименти в този раздел са направени, използвайки локална мрежа (Lokal Area Network), включваща 8 работни станции IBM RISC 6000/250 (избрана е такава мрежа поради липса на достъп до по-добри архитектури). Реализирани са двата варианта на алгоритъма стабилизи-

k	S_2	E_2	S_4	E_4	S_6	E_6	S_8	E_8
100	0.02	0.01	0.40	0.10	0.48	0.08	0.56	0.07
500	0.14	0.07	0.88	0.22	1.02	0.17	1.28	0.16
1000	0.20	0.10	1.12	0.28	1.44	0.24	1.76	0.22
2000	0.24	0.12	1.32	0.33	1.74	0.29	2.16	0.27
4000	0.32	0.16	1.56	0.39	1.92	0.32	2.40	0.30
5000	0.38	0.19	1.68	0.42	2.04	0.34	2.56	0.32
6000	0.44	0.22	1.88	0.47	2.28	0.38	2.88	0.36
8000	0.48	0.24	2.12	0.53	2.64	0.44	3.52	0.44
10000	0.56	0.28	2.36	0.59	3.00	0.50	3.92	0.49
20000	0.72	0.36	2.62	0.65	3.48	0.58	4.64	0.58

Таблица 1.6: Паралелно ускорение и ефективност за оригиналния алгоритъм (без стабилизация) на 2, 4, 6 и 8 процесора.

ран и нестабилизиран. Тестовете са направени, използвайки 2, 4, 6 и 8 процесора, като броят на процесорите е равен на броя на стартираните процеси, равен на s - броят на блоковете B_i в изходната матрица.

Обект на нашето внимание са:

- Паралелното ускорение на алгоритъма $S_p = t_1/t_p,\ 0 \leq S_p \leq p$, където t_1 е времето за последователно изпълнение на един процесор на метода на Гаус (като алгоритъм изискващ най-малко време за последователното си изпълнение), а t_p е времето за паралелното изпълнение на изследвания алгоритъм на p процесора. При това нека да уточним, че когато измерваме ускорението на стабилизирания алгоритъм сравнението се прави с метода на Гаус с избор на главен елемент, а в случая на нестабилизиран алгоритъм с метода на Гаус без избор на главен елемент.

- Ефективността на алгоритъма $E_p = S_p/p,\ 0 \leq E_p \leq 1$.

Получените резултати за S_p и E_p при стабилизирания вариант са представени в Таблица 1.5, а за нестабилизирания в Таблица 1.6. Нека да напомним, че размерът на цялата система е $n = ks - 1$.

Сравнявайки паралелната реализация на двата варианта на изследвания алгоритъм се вижда, че ускорението и ефективността при стабилизирания вариант са по-добри. Обяснението на това е, че при стабилизирания вариант на алгоритъма отделните процесори извършват по-голям

обем от аритметични операции. Освен това и при двата варианта на алгоритъма най-добри резултати по отношение на ефективността и ускорението се получават при четири процесора. Разпаралелването на алгоритъма върху по-голям брой процесори води до по-голямо ускорение като абсолютна стойност и по-малко в относителен смисъл, при което ефективността намалява. Това може да се обясни с ниската пропусквателна способност на използвания Ethernet network protocol, която забавя в известна степен комуникациите. Както може да се очаква с увеличаване на размерността на решаваната система нарастват и стойностите на E_p и S_p. При малки размерности разпаралелването е необосновано поради големите служебни времена (overhead time) за стартиране и прекратяване на процесите и за буфериране на данните. Ясно е, че по отношение на получените резултати за E_p и S_p има какво още да се желае, в смисъл те да се доближат още повече до теоретичните нива: 1 за ефективността и p за ускорението. Някои възможни варианти за повишаване на ускорението и ефективността са:

- паралелната реализация да се направи върху Fiber Distributed Data Interface (FDDI) мрежа от станции, която се характеризира с намалени служебни времена и повишена пропусквателна способност.
- паралелната реализация да се направи използвайки мощен съвременен паралелен компютър, например от типа на Cray.

За съжаление това не е направено поради липса на достъп до подобен тип архитектури.

Глава 2

Изследване на числената устойчивост на метода на Уанг за решаване на лентови системи линейни уравнения

Основните резултати, представени в тази глава, са публикувани в статията:
Yalamov, P., V. Pavlov. Backward Stabilty of a Parallel Partitioning Algorithm for Banded Linear Systems. *Proc. of 4th International Conference on Numerical Methods and Applications (Eds. O. Iliev et. al.), Sofia, August 19–23, 1998*, World Scientific Publ., 655–663, 1999.

В тази глава е разгледано обобщение на метода на Уанг [64] в случая на решаване на лентови системи линейни уравнения и е представен анализ на числената устойчивост на алгоритъма в този случай. При това хронологията на излагане на изследванията е като тази в предходната глава, където беше изследван същият метод във важния частен случай на решаване на тридиагонални системи. Нека да отбележим, че в настоящата глава, навсякъде където доказателствата на твърденията са напълно аналогични на тези във вече изследвания тридиагонален случай, са пропускани, а там където нещата са специфични или не съвсем очевидни, са представяни подробно.

Главата се състои от четири раздела. В Раздел 2.1 е направено описание на обобщения вариант на алгоритъма за решаване на лентови системи линейни уравнения. Най-напред разглежданата система се структурира в блочно тридиагонален вид, след което отчитайки спецификата

на решаваната задача се пренася алгоритъма от тридиагоналния случай (описан в Раздел 1.1). При това трите негови етапа се запазват. Понататък, в Раздел 2.2 е представен пълен анализ на разпространението на грешките от закръгляване при компютърна реализация на описания алгоритъм. Съществени негови особености са, че както и в Раздел 1.2 той е покомпонентен, а също и че е използван комбиниран подход на прав и обратен анализ в отделните етапи на алгоритъма. Отново навсякъде извежданите оценки са точни, т.е. не са пренебрегвани високите (след първия) порядъци относно машинната точност. Както и при тридиагоналния случай, трите етапа на алгоритъма са анализирани съответно в три леми, естествено отчитайки особеностите на алгоритъма в лентовия случай. Основният резултат в раздела съдържа оценки за обратната грешка (в покомпоненетен смисъл) и за правата грешка в решението на системата (в относителен смисъл, по норма безкрайност). Получените оценки показват, че както и в тридиагоналния случай, грешката в решението на системата зависи по подобен начин от нейната обусловеност и от устойчивостта на алгоритъма. Влияние върху устойчивостта на алгоритъма оказва ръстът на елементите в отделните етапи на алгоритъма. В общия случай, дори и решаваната система да е добре обусловена, този ръст може да е неограничен и в крайна сметка полученото решение да се различава съществено от точното. Поради тази причина в Раздел 2.3 са разгледани някои специални класове от лентови матрици: матрици с диагонално преобладаване (по редове или стълбове), симетрични и положително определени матрици и M-матрици. За тези класове ръстът на елементите в отделните етапи на алгоритъма е ограничен от малки константи. Това дава основание, както и в тридиагоналния случай, да се направи важният извод, че за разглежданите специални класове от матрици алгоритъмът е числено устойчив, а грешката в решението на системата зависи главно от обусловеността на решаваната система. Накрая в Раздел 2.4 са представени числени експерименти, които потвърждават че теоретично получените оценки са почти достижими.

2.1 Описание на алгоритъма на Уанг

Нека да разгледаме следната лентова система линейни уравнения:

$$Ax = d, \tag{2.1}$$

където матрицата $A \in \mathcal{R}^{n\times n}$ има ширина на лентата $2j+1$ или с други думи броят на нейните ненулеви диагонали над и под главния диагонал е един и същ и е равен на j (това изискване не е съществено за изследванията,

които ще изложим понататък). При това естествено е да предполагаме че $2j+1 << n$, т.е. A не е близка до плътна матрица. Нека още за простота да допуснем че $n = ks - j$, където s е броят на паралелните процесори, които бихме желали да използваме, а k е произволно цяло положително число.

Разглежданата система можем да структурираме в блочно тридиагонален вид по следния начин:

$$\begin{pmatrix} B_1 & \bar{c}_1 & & & & & & \\ a_k & b_k & c_k & & & & & \\ & \bar{a}_2 & B_2 & \bar{c}_2 & & & & \\ & & a_{2k} & b_{2k} & c_{2k} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \bar{a}_{s-1} & B_{s-1} & \bar{c}_{s-1} & \\ & & & & & a_{(s-1)k} & b_{(s-1)k} & c_{(s-1)k} \\ & & & & & & \bar{a}_s & B_s \end{pmatrix} \begin{pmatrix} X_1 \\ x_k \\ X_2 \\ x_{2k} \\ \vdots \\ X_{s-1} \\ x_{(s-1)k} \\ X_s \end{pmatrix} = \begin{pmatrix} D_1 \\ d_k \\ D_2 \\ d_{2k} \\ \vdots \\ D_{s-1} \\ d_{(s-1)k} \\ D_s \end{pmatrix},$$

където $B_i \in \mathcal{R}^{(k-j)\times(k-j)}, i = 1, 2, \ldots, s$, са лентови матрици със същата ширина на лентата като на A, но от няколко (от порядъка на s) пъти по-малка размерност, $\bar{a}_i, \bar{c}_i$ са матрици от следния вид:

$$\begin{aligned} \bar{a}_i &= (a_{(i-1)k+1}, 0, \ldots, 0)^T \in \mathcal{R}^{(k-j)\times j}, \quad i = 2, \ldots, s, \\ \bar{c}_i &= (0, \ldots, 0, c_{ik-1})^T \in \mathcal{R}^{(k-j)\times j}, \quad i = 1, \ldots, s-1, \end{aligned}$$

чиито елементи $a_{(i-1)k+1}, c_{ik-1} \in \mathcal{R}^{j\times j}$. Същото важи и за елементите a_{ik}, b_{ik}, c_{ik}, т.е. $a_{ik}, b_{ik}, c_{ik} \in \mathcal{R}^{j\times j}, i = 1, 2, \ldots, s-1$. Накрая, за компонентите на вектора стълб от неизвестни и на дясната част в горното представяне е изпълнено

$$X_i, D_i \in \mathcal{R}^{(k-j)\times 1}, \ i = 1, 2, \ldots, s, \quad x_{ik}, d_{ik} \in \mathcal{R}^{j\times 1}, \ i = 1, 2, \ldots, s-1.$$

Нека, както и в предишната глава, да представим алгоритъма в блочна форма. За тази цел най-напред да дефинираме следната пермутация на числата $[1, \ldots, sk-j]$:

$$\begin{aligned} &[1 : k-j; \ldots; (i-1)k+1 : ik-j; \ldots; (s-1)k+1 : sk-j; \\ &k-j+1 : k; \ldots; ik-j+1 : ik; \ldots; (s-1)k-j+1 : (s-1)k]. \end{aligned}$$

Имайки предвид, че $A \in \mathcal{R}^{n\times n}$, където $n = sk-j$, да приложим същата пермутация към редовете и стълбовете на матрицата A, при това да означим съответстващата пермутационна матрица с $\mathcal{P}$. Тогава получаваме:

$$\mathcal{A}\mathcal{P}x = \mathcal{P}d, \quad \text{където} \quad \mathcal{A} = \mathcal{P}A\mathcal{P}^T = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{2.2}$$

с елементи $A_{11} = \mathrm{diag}\{B_1, B_2, \ldots, B_s\} \in \mathcal{R}^{s(k-j)\times s(k-j)}$,

$$A_{12} = \begin{pmatrix} \bar{c}_1 & & & \\ \bar{a}_2 & \bar{c}_2 & & \\ & \ddots & \ddots & \\ & & \ddots & \bar{c}_{s-1} \\ & & & \bar{a}_s \end{pmatrix} \in \mathcal{R}^{s(k-j)\times j(s-1)},$$

$$A_{21} = \begin{pmatrix} 0 & \cdots & a_k & c_k & \cdots & 0 & & & & \\ & & & 0 & \cdots & a_{2k} & c_{2k} & \cdots & 0 & \\ & & & & \ddots & & \ddots & \ddots & & \ddots \\ & & & & & & 0 & \cdots & a_{(s-1)k} & c_{(s-1)k} & \cdots & 0 \end{pmatrix} \in \mathcal{R}^{j(s-1)\times s(k-j)},$$

и $A_{22} = \mathrm{diag}(b_k, b_{2k}, \ldots, b_{(s-1)k}) \in \mathcal{R}^{j(s-1)\times j(s-1)}$.

Понататък ще правим разлика между двете матрици A (оригиналната) и $\mathcal{A}$ (пермутираната), както в предишната глава.

Алгоритъмът би могъл да бъде структуриран по същия начин, както в тридиагоналния случай (Раздел 1.1):

Eman 1. Получаване на блочна LU-факторизация на $\mathcal{A}$

$$\mathcal{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = LU = \begin{pmatrix} A_{11} & 0 \\ A_{21} & I_{j(s-1)} \end{pmatrix} \begin{pmatrix} I_{s(k-j)} & R \\ 0 & S \end{pmatrix} \tag{2.3}$$

посредством следните стъпки:

1. Получаване на LU-факторизация на A_{11} (ако е необходимо използвайки частичен избор на главен елемент)

$$A_{11} = \mathcal{P}_1 L_1 U_1,$$

където $\mathcal{P}_1$ е пермутационна матрица, L_1 е долнотриъгълна матрица с единици по главния диагонал, а U_1 е горнотриъгълна матрица.

2. Решава се

$$A_{11} R = A_{12}, \tag{2.4}$$

използвайки вече получената в предишната стъпка LU-факторизация, след което се конструира т. нар. редуцирана матрица

$$S = A_{22} - A_{21} R. \tag{2.5}$$

Всъщност S е точно допълнението на Шур на A_{11} в $\mathcal{A}$.

Естествено в този етап считаме, че разлагането (2.3) съществува, т.е. предполагаме, че A и A_{11} са неособени матрици. Възможно е обаче A_{11} да бъде особена или някой от блоковете B_i да са особени и това би довело до прекъсване на алгоритъма или до взрив на грешките от закръгляване. Коментар за изход от тази опасна ситуация, в частния случай на тридиагонални системи, вече направихме в Раздел 1.4. Тъй като идеята се пренася напълно аналогично в лентовия случай и резултатите, които се получават не са нови, в тази глава няма да отделяме специално място на възможната стабилизация на алгоритъма.

Етап 2. Решава се $Ly = d$, използвайки вече получената в Етап 1 матрица L.

Етап 3. Решава се $Ux = y$, като най-напред се решава редуцираната система (с матрицата S), използвайки Гаусово изключване (ако е необходимо с избор на главен елемент), в резултат на което се намират блочните компоненти $x_k, x_{2k}, \ldots, x_{(s-1)k}$ на решението. След това посредством обратна субституция в блочна форма се намират всички останали компоненти.

Ако вземем предвид структурата на A_{11} и A_{12}, и (2.4), то лесно се вижда, че матрицата R е също структурирана и има следния вид:

$$R = \begin{pmatrix} p^{(1)} & & & \\ q^{(2)} & p^{(2)} & & \\ & \ddots & \ddots & \\ & & \ddots & p^{(s-1)} \\ & & & q^{(s)} \end{pmatrix} \in \mathcal{R}^{s(k-1)\times(s-1)}, \tag{2.6}$$

където

$$\begin{aligned} p^{(i)} &= (p_{(i-1)k+1}, p_{(i-1)k+2}, \ldots, p_{ik-1})^T \in \mathcal{R}^{(k-j)\times j}, \\ q^{(i)} &= (q_{(i-1)k+1}, q_{(i-1)k+2}, \ldots, q_{ik-1})^T \in \mathcal{R}^{(k-j)\times j}. \end{aligned}$$

Що се отнася до редуцираната матрица, то специфично свойство на алгоритъма е, че тя е отново блочно тридиагонална (може да бъде разглеждана и като лентова с ширина на лентата $4j-1$)

$$S = \begin{pmatrix} v_1 & w_1 & & & \\ u_2 & v_2 & w_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & w_{s-2} \\ & & & u_{s-1} & v_{s-1} \end{pmatrix} \in \mathcal{R}^{j(s-1)\times j(s-1)},$$

където елементите на S са плътни матрици от размерност $j \times j$, които се получават по следните формули:

$$u_i = -a_{ik}q_{ik-1}, \quad v_i = b_{ik} - a_{ik}p_{ik-1} - c_{ik}q_{ik+1}, \quad w_i = -c_{ik}p_{ik+1}. \tag{2.7}$$

Имайки предвид структурирания вид на алгоритъма и блочната структура на A_{11} и R нека да отбележим, че както и в тридиагоналния случай алгоритъмът запазва добрите си паралелни свойства и при решаване на лентови системи. По-точно всички изчисления, с изключение на решаването на редуцираната система могат да се направят паралелно, при това за да се получи естествено разпаралелване следва числото s да бъде равно на броя на използваните процесори.

2.2 Анализ на грешките от закръгляване

В този раздел ще проследим разпространението на грешките от закръгляване от началото до края на изчислителния процес при компютърна реализация на описания алгоритъм.

В предложения анализ на грешките от закръгляване ще използваме някои резултати, представени от Хайам в [36]. По-точно ако за решаване на лентовата система $Ax = d$ е използвано Гаусово изключване и ако $\hat{L}\hat{U}$ е компютърно полученото LU разлагане, а $\hat{x}$ е компютърно пресметнатото решение, то еквивалентно смутената задача има вида (виж Теорема 9.4 в [36]) $(A + \Delta A)\hat{x} = d$, при което

$$|\Delta A| \leq f(\rho_0)|\hat{L}||\hat{U}|, \quad f(\rho_0) = \gamma_{j+1} + \gamma_{2j+1}, \tag{2.8}$$

където $\gamma_n = n\rho_0/(1 - n\rho_0)$. Да допуснем, че $|\hat{L}||\hat{U}| \leq K|A|$, където K е константа, която е горна граница за ръста на елементите при намиране на LU разлагане на матрицата A. Тогава имайки предвид (2.8) получаваме

$$|\Delta A| \leq K|A|f(\rho_0). \tag{2.9}$$

Нека да отбележим че дори и A да е добре обусловена е възможно ръстът на елементите при намиране на нейното LU разлагане да бъде неограничен или да е ограничен, но константата K е твърде голяма. В такива случаи обикновено се препоръчва ако е възможно да бъдат отделени класове от матрици, за които ръстът на елементите е ограничен. За конкретния случай подобно нещо е направено в [36], където са намерени ограничения за K (отделените в [36] класове са разгледани и в Раздел 2.3).

Нека сега да представим нашите изследвания относно числената устойчивост на алгоритъма. В следващите три леми са анализирани отделните негови етапи, след което в Теорема 2.1 е представен и основният резултат в този раздел.

Лема 2.1 *Ако е изпълнено* $L\hat{U} = \mathcal{A} + E$, *тогава за обратната грешка* E, *получена от изчисленията за намиране на блочната* LU *факторизация на* $\mathcal{A}$, *е валидна оценката*

$$|E| \leq K_1 |\mathcal{A}||N| f(\rho_0), \quad K_1 = \max\{k_1, 1\},$$

където k_1 *ограничава ръста на елементите при получаване на* LU*-факторизацията на матрицата* A_{11} *(Етап 1 на алгоритъма), а* N *е следната матрица*

$$N = \begin{pmatrix} 0 & \hat{R} \\ 0 & I_{j(s-1)} \end{pmatrix}.$$

Доказателство. Доказателството е аналогично на това направено в Лема 1.1. Все пак за по-голяма яснота ще повторим основните и специфичните за разглеждания случай моменти. Най-напред по същия начин, както в Лема 1.1 намираме израз за обратната грешка E от изчисленията за получаване на блочната LU факторизация на $\mathcal{A}$

$$E = \begin{pmatrix} 0 & A_{11}\delta R \\ 0 & \delta S \end{pmatrix}. \tag{2.10}$$

За да оценим E е необходимо да получим оценка за нейните ненулеви елементи. При това оценката за елемента $A_{11}\delta R$ се запазва и се извежда по същия начин, както в Лема 1.1.

$$|A_{11}\delta R| \leq k_1 |A_{11}||\hat{R}| f(\rho_0). \tag{2.11}$$

Що се отнася до другия ненулев елемент δS на матрицата E, то имайки предвид израза (2.5), по който се пресмята S получаваме следната оценка за грешката δS

$$|\delta S| \leq |E_1| + |E_2|, \tag{2.12}$$

където E_1 и E_2 са съответно грешките от матричното произведение $A_{21}\,\hat{R}$ и матричната сума $A_{22} + A_{21}\,\hat{R}$. Тогава замествайки в (2.12) добре известните оценки за грешката при компютърно пресмятане на сума и произведение на матрици (виж [36]), получаваме

$$\begin{aligned} |\delta S| &\leq \gamma_{j+1} |A_{21}|\,|\hat{R}| + (|A_{22}| + |A_{21}|\,|\hat{R}|)\rho_0 \\ &\leq (|A_{22}| + |A_{21}||\hat{R}|) g(\rho_0), \end{aligned} \tag{2.13}$$

където $g(\rho_0) = \gamma_{j+1} + \rho_0$, при което е очевидно че $g(\rho_0) \leq f(\rho_0)$. Тогава от (2.10), (2.11) и (2.13) намираме търсената оценка

$$|E| \leq \begin{pmatrix} 0 & k_1 |A_{11}||\hat{R}| f(\rho_0) \\ 0 & (|A_{22}| + |A_{21}||\hat{R}|) g(\rho_0) \end{pmatrix}$$

$$\begin{aligned} &\leq K_1 \begin{pmatrix} |A_{11}| & 0 \\ |A_{21}| & |A_{22}| \end{pmatrix} \begin{pmatrix} 0 & |\hat{R}| \\ 0 & I_{j(s-1)} \end{pmatrix} f(\rho_0) \\ &\leq K_1 |\mathcal{A}| \begin{pmatrix} 0 & |\hat{R}| \\ 0 & I_{j(s-1)} \end{pmatrix} f(\rho_0) \\ &= K_1 |\mathcal{A}||N| f(\rho_0), \end{aligned}$$

където $K_1 = \max\{k_1, 1\}$ и $N = \begin{pmatrix} 0 & \hat{R} \\ 0 & I_{j(s-1)} \end{pmatrix}$. ◊

След като получихме оценка за грешката, допусната при блочната LU факторизация на $\mathcal{A}$, нека понататък да анализираме и грешката, която се допуска при намиране на решението на системата с двете блочно триъгълни матрици L и $\hat{U}$.

Лема 2.2 *При решаване на система с блочно долнотриъгълната матрица L е в сила $(L + \Delta L)\hat{y} = d$, където за матрицата от еквивалентните смущения ΔL е валидна оценката*

$$|\Delta L| \leq K_1 |L| f(\rho_0).$$

Доказателство. Имайки предвид вида на L (виж (2.3)), то за матрицата от еквивалентни смущения ΔL получаваме

$$|\Delta L| = \begin{pmatrix} |\Delta A_{11}| & 0 \\ |\Delta A_{21}| & |\Delta I_{j(s-1)}| \end{pmatrix}, \tag{2.14}$$

където ΔA_{11} е смущението от Етап 1 (решаване на лентова система), $\Delta A_{21}, \Delta I_{j(s-1)}$ са смущения, идващи от изключването на A_{21} (това очевидно е еквивалентно на решаване на триъгълна система). За блочно диагоналната матрица A_{11}, отчитайки че тя е и лентова ако приложим (2.9), получаваме

$$|\Delta A_{11}| \leq k_1 |A_{11}| f(\rho_0). \tag{2.15}$$

Оценки за останалите ненулеви елементи на матрицата $|\Delta L|$ получаваме като използваме направения в [36] анализ на грешката при решаване на триъгълна система и вземем под внимание ширината на лентата на съответните матрици

$$|\Delta A_{21}| \leq \gamma_{j+1} |A_{21}|, \tag{2.16}$$

$$|\Delta I_{j(s-1)}| \leq \gamma_{j+1} I_{j(s-1)}. \tag{2.17}$$

След като получихме оценки за всички ненулеви елементи на матрицата ΔL, то от (2.14), (2.15), (2.16) и (2.17) стигаме до твърдението на лемата

$$|\Delta L| \leq K_1 |L| f(\rho_0). \quad \Diamond$$

Лема 2.3 *При решаване на система с блочно горнотриъгълната матрица $\hat{U}$ е в сила $(\hat{U}+\Delta\hat{U})\hat{x}=\hat{y}$, където за матрицата от еквивалентните смущения $\Delta\hat{U}$ е валидна оценката*

$$|\Delta\hat{U}| \leq K_2|\hat{U}|h(\rho_0), \quad K_2=\max\{k_2,1\}, \quad h(\rho_0)=\gamma_{2j-1}+\gamma_{4j-1},$$

където k_2 ограничава ръста на елементите при прилагане на Гаусово изключване за намиране на решение на редуцираната система (Етап 3 на алгоритъма).

Доказателство. Доказателството ще направим по подобен начин като при предходната лема. Изхождайки от вида на $\hat{U}$ (виж (2.3)), то за матрицата от еквивалентни смущения $\Delta\hat{U}$ получаваме

$$|\Delta\hat{U}| = \begin{pmatrix} |\Delta I_{s(k-j)}| & |\Delta\hat{R}| \\ 0 & |\Delta\hat{S}| \end{pmatrix}, \tag{2.18}$$

където $\hat{S}=\tilde{S}+\delta S$, при което $\hat{S}$ е лентова с ширина на лентата $4j-1$. Имайки предвид отново Теорема 9.4 от [36] и (2.9) получаваме

$$|\Delta\hat{S}| \leq (\gamma_{2j-1}+\gamma_{4j-1})|\hat{S}|K_2. \tag{2.19}$$

Оценки за останалите ненулеви елементи на матрицата $|\Delta\hat{U}|$ получаваме като използваме направения в [36] анализ на грешката при решаване на триъгълна система и вземем под внимание ширината на лентата на съответните матрици

$$|\Delta\hat{R}| \leq \gamma_{2j+1}|\hat{R}|, \tag{2.20}$$

$$|\Delta I_{s(k-j)}| \leq \gamma_{2j} I_{s(k-j)}. \tag{2.21}$$

След като получихме оценки за всички ненулеви елементи на матрицата $\Delta\hat{U}$, то от (2.18), (2.19), (2.20) и (2.21) стигаме до твърдението на лемата

$$|\Delta\hat{U}| \leq K_2|\hat{U}|h(\rho_0), \quad h(\rho_0)=\gamma_{2j-1}+\gamma_{4j-1}. \qquad \Diamond$$

Преди да представим основния резултат, относно устойчивостта на целия алгоритъм, ще направим някои уточнения. Нека да разгледаме $l-$та компонента на произведението $|N||\mathcal{P}\hat{x}|, l=1,2,\ldots,s(k-j)$. Имайки предвид вида на R (виж (2.6)), разглеждана като блочна матрица, и по-точно факта, че във всеки ред на тази матрица има само два ненулеви блочни елемента, то за тази $l-$та компонента е изпълнено

$$\begin{aligned}(|N||\mathcal{P}\hat{x}|)_l &\leq \| \, |\hat{R}_{l,i}|\,|\hat{x}_{ik}| + |\hat{R}_{l,i+1}|\,|\hat{x}_{(i+1)k}| \, \|_\infty \\ &\leq \|\hat{R}\|_\infty \max\left\{\|\hat{x}_{ik}\|_\infty, \|\hat{x}_{(i+1)k}\|_\infty\right\}, \end{aligned} \tag{2.22}$$

за някое i. Нека да дефинираме вектор $\mathcal{P}x^*$ по следния начин:

$$\mathcal{P}x^* = [(x_1^*)^T \ (x_2^*)^T]^T,$$

където

$$x_1^* = (\|\hat{x}_k\|_\infty e, \max\{\|\hat{x}_k\|_\infty, \|\hat{x}_{2k}\|_\infty\} e, \ldots, \max\{\|\hat{x}_{(s-2)k}\|_\infty, \|\hat{x}_{(s-1)k}\|_\infty\} e)^T,$$

$$x_2^* = (|\hat{x}_k^T|, \ldots, |\hat{x}_{(s-1)k}^T|)^T,$$

$e = (1, 1, \ldots, 1) \in \mathcal{R}^{1\times(k-j)}$. Тогава можем да определим и вектор x^*

$$x^* = (\|\hat{x}_k\|_\infty e, |\hat{x}_k^T|, \max\{\|\hat{x}_k\|_\infty, \|\hat{x}_{2k}\|_\infty\} e, \ldots, |\hat{x}_{(s-1)k}^T|, \max\{\|\hat{x}_{(s-2)k}\|_\infty, \|\hat{x}_{(s-1)k}\|_\infty\} e)^T.$$

Изхождайки от (2.22) и от вида на x^* получаваме

$$|N||\mathcal{P}\hat{x}| \leq \begin{pmatrix} \|R\|_\infty x_1^* \\ x_2^* \end{pmatrix} \leq r\mathcal{P}x^*, \tag{2.23}$$

където $r = \max\left\{\|\hat{R}\|_\infty, 1\right\}$.

Сега, използвайки така дефинирания вектор x^*, можем да въведем следното число на обусловеност:

$$cond^*(\mathcal{A}, \mathcal{P}x^*) = \frac{\| \ |\mathcal{A}^{-1}| \ |\mathcal{A}| \ \mathcal{P}x^* \ \|_\infty}{\|\mathcal{P}\hat{x}\|_\infty}.$$

Лесно може да се провери че

$$cond^*(\mathcal{A}, x^*) \leq cond(\mathcal{A}),$$

където $cond(\mathcal{A}) = \||\mathcal{A}^{-1}||\mathcal{A}|\|_\infty$. Връзката между обусловеността на пермутираната и оригиналната матрица, както и в Раздел 1.2, се запазва

$$cond(\mathcal{A}) = cond(A), \tag{2.24}$$

$$cond^*(\mathcal{A}, \mathcal{P}x^*) = cond^*(A, x^*), \tag{2.25}$$

$$cond(\mathcal{A}, \mathcal{P}\hat{x}) = cond(A, \hat{x}), \tag{2.26}$$

където $cond(A, \hat{x})$ е вече въведеното в Раздел 1.2 число на обусловеност на Скийл [56].

Смисълът на така дефинираното число на обусловеност $cond^*(A, x^*)$ е същият, както в Глава 1. Пример, който потвърждава това, е представен в Раздел 2.4.

В следващата теорема, използвайки Леми 2.1 - 2.3, ще представим основния резултат в този раздел.

Теорема 2.1 *При реализация на алгоритъма на Уанг за решаване на системата (2.1) е в сила* $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x} = \mathcal{P}d$, *като*

$$|\Delta\mathcal{A}| \le |\mathcal{A}|h_1(\rho_0) + |\mathcal{A}||N|h_2(\rho_0),$$

а

$$\begin{aligned} h_1(\rho_0) &= K_1 f(\rho_0) + K_2 h(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + K_1 f(\rho_0)g(\rho_0) + K_2 h(\rho_0)g(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0), \\ h_2(\rho_0) &= 3K_1 f(\rho_0) + 2K_2 h(\rho_0) + 2K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + 3K_1 f(\rho_0)g(\rho_0) + 3K_2 h(\rho_0)g(\rho_0) + 3K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0) \\ &\quad + K_1 f(\rho_0)g^2(\rho_0) + K_2 h(\rho_0)g^2(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g^2(\rho_0). \end{aligned}$$

Освен това за правата грешка в решението на системата е изпълнено

$$\frac{\|\delta x\|}{\|\hat{x}\|} = \frac{\|\hat{x}-x\|_\infty}{\|\hat{x}\|_\infty} \le cond(A,\hat{x})h_1(\rho_0) + cond^*(A,x^*)rh_2(\rho_0).$$

Доказателство. Идеята за доказателство на тази теорема е същата, както при Теорема 1.1. Най-напред оценката (1.26) се запазва

$$|\Delta\mathcal{A}| \le |E| + |\Delta L||\hat{U}| + |L||\Delta\hat{U}| + |\Delta L||\Delta\hat{U}|. \tag{2.27}$$

Сега от Леми 2.1 - 2.3 и (2.27) получаваме, че

$$\begin{aligned} |\Delta\mathcal{A}| &\le K_1|\mathcal{A}||N|f(\rho_0) + K_1|L||\hat{U}|f(\rho_0) \\ &\quad + K_2|L||\hat{U}|h(\rho_0) + K_1K_2|L||\hat{U}|f(\rho_0)h(\rho_0) \\ &= K_1|\mathcal{A}||N|f(\rho_0) + (K_1 f(\rho_0) + K_2 h(\rho_0) \\ &\quad + K_1K_2 f(\rho_0)h(\rho_0))\,|L||\hat{U}|, \end{aligned} \tag{2.28}$$

но

$$\begin{aligned} |L||\hat{U}| &= \begin{pmatrix} |A_{11}| & 0 \\ |A_{21}| & I_{j(s-1)} \end{pmatrix} \begin{pmatrix} I_{s(k-j)} & |\hat{R}| \\ 0 & |\hat{S}| \end{pmatrix} \\ &= \begin{pmatrix} |A_{11}| & |A_{11}||\hat{R}| \\ |A_{21}| & |A_{21}||\hat{R}| + |\hat{S}| \end{pmatrix} \\ &\le \begin{pmatrix} |A_{11}| & |A_{11}||\hat{R}| + |A_{12}| \\ |A_{21}| & |A_{22}| + 2|A_{21}||\hat{R}| + |\delta S| \end{pmatrix}, \end{aligned} \tag{2.29}$$

където сме използвали факта че $\hat{S} = A_{22} - A_{21}\hat{R} + \delta S$. Ако заместим (2.13) в (2.29) и направим някои преобразувания стигаме до

$$|L||\hat{U}| \le \begin{pmatrix} |A_{11}| & |A_{11}||\hat{R}| + |A_{12}| \\ |A_{21}| & |A_{22}|(1+g(\rho_0)) + |A_{21}||\hat{R}|(2+g(\rho_0)) \end{pmatrix}$$

$$\begin{aligned} &\leq |\mathcal{A}|(1+g(\rho_0)) + \begin{pmatrix} 0 & |A_{11}||\hat{R}| \\ 0 & |A_{21}||\hat{R}| \end{pmatrix}(2+g(\rho_0)) \\ &\leq |\mathcal{A}|(1+g(\rho_0)) + |\mathcal{A}| \begin{pmatrix} 0 & |\hat{R}| \\ 0 & I_{j(s-1)} \end{pmatrix}(2+g(\rho_0)) \\ &= |\mathcal{A}|(1+g(\rho_0)) + |\mathcal{A}||N|(2+g(\rho_0)). \end{aligned} \tag{2.30}$$

Сега от (2.28) и (2.30) получаваме следната оценка за матрицата от еквивалентни смущения $\Delta\mathcal{A}$ при реализация на целия алгоритъм:

$$\begin{aligned} |\Delta\mathcal{A}| &\leq K_1|\mathcal{A}||N|f(\rho_0) + (K_1 f(\rho_0) + K_2 h(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)) \\ &\quad [|\mathcal{A}|(1+g(\rho_0)) + |\mathcal{A}||N|(2+g(\rho_0))] \\ &= |\mathcal{A}|h_1(\rho_0) + |\mathcal{A}||N|h_2(\rho_0), \end{aligned} \tag{2.31}$$

където

$$\begin{aligned} h_1(\rho_0) &= K_1 f(\rho_0) + K_2 h(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + K_1 f(\rho_0)g(\rho_0) + K_2 h(\rho_0)g(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0), \\ h_2(\rho_0) &= 3K_1 f(\rho_0) + 2K_2 h(\rho_0) + 2K_1K_2 f(\rho_0)h(\rho_0) \\ &\quad + 3K_1 f(\rho_0)g(\rho_0) + 3K_2 h(\rho_0)g(\rho_0) + 3K_1K_2 f(\rho_0)h(\rho_0)g(\rho_0) \\ &\quad + K_1 f(\rho_0)g^2(\rho_0) + K_2 h(\rho_0)g^2(\rho_0) + K_1K_2 f(\rho_0)h(\rho_0)g^2(\rho_0). \end{aligned}$$

Остава да оценим правата грешка $\mathcal{P}(\hat{x}-x)$. От представянето $(\mathcal{A}+\Delta\mathcal{A})\mathcal{P}\hat{x} = \mathcal{P}d$ и (2.2) лесно получаваме израз, който я съдържа

$$\mathcal{P}(\hat{x}-x) = -\mathcal{A}^{-1}\Delta\mathcal{A}\mathcal{P}\hat{x}. \tag{2.32}$$

Тогава от (2.31) и (2.32) получаваме

$$|\mathcal{P}(\hat{x}-x)| \leq |\mathcal{A}^{-1}||\mathcal{A}||\mathcal{P}\hat{x}|h_1(\rho_0) + |\mathcal{A}^{-1}||\mathcal{A}||N||\mathcal{P}\hat{x}|h_2(\rho_0). \tag{2.33}$$

Накрая от (2.23) и (2.33) стигаме до

$$\begin{aligned} \frac{\|\delta x\|_\infty}{\|\hat{x}\|_\infty} &\leq \frac{\| \, |\mathcal{A}^{-1}| \, |\mathcal{A}| \, |\mathcal{P}\hat{x}| \, \|_\infty h_1(\rho_0) + \| \, |\mathcal{A}^{-1}| \, |\mathcal{A}| \, \mathcal{P}x^* \, \|_\infty r h_2(\rho_0)}{\|\hat{x}\|_\infty} \\ &= cond(\mathcal{A}, \mathcal{P}\hat{x})h_1(\rho_0) + cond^*(\mathcal{A}, \mathcal{P}x^*)r h_2(\rho_0). \end{aligned} \tag{2.34}$$

Остава да вземем под внимание (2.25) и (2.26), откъдето получаваме и окончателната оценка в относителен смисъл за грешката в решението на системата. $\Diamond$

Получените оценки (2.31) и (2.34) показват, че грешката в решението на системата зависи от нейната обусловеност и от устойчивостта на алгоритъма (имат се предвид функциите $h_1(\rho_0)$ и $h_2(\rho_0)$). От своя страна

тези функции са малки (от порядък $\mathcal{O}(j\rho_0)$), стига константите K_1, K_2 и r да бъдат малки. Удовлетворителни ограничения за тези константи могат да бъдат намерени ако матрицата A принадлежи към някои от следните специални класове от матрици: матрици с диагонално преобладаване по редове или стълбове, симетрични и положително определени матрици, M-матрици. Известно е, че за посочените класове от плътни матрици методът на Гаус е числено устойчив [36]. При това ако решаваме една такава система с матрица T например, използвайки Гаусово изключване, то е в сила следната оценка [36, стр. 176]:

$$|\hat{L}|\,|\hat{U}| \leq \frac{1}{1-\gamma_n}|T|,$$

където $\gamma_n = n\rho_0/(1-n\rho_0)$, а n е размерността на системата. В случая, когато T е и лентова с ширина на лентата j оценката добива вида (виж [36, стр. 182]):

$$|\hat{L}|\,|\hat{U}| \leq \frac{1}{1-\gamma_{j+1}}|T|, \tag{2.35}$$

Следователно, щом A принадлежи към някои от посочените специални класове, ограничение за K_1 е известно (т.е. $K = 1/(1-\gamma_{j+1})$, виж (2.9)). Ако успеем да покажем, че редуцираната матрица S запазва свойствата на A, то това автоматично ще означава, че за K_2 ще е валидно същото ограничение. Върху този проблем, както и върху оценяване на $\|\hat{R}\|_\infty$, за посочените специални класове е посветен следващият раздел.

2.3 Специални класове от матрици

Вече стана ясно, че в този раздел ще отделим специално внимание на случая, когато матрицата A принадлежи към някои от следните специални класове: матрици с диагонално преобладаване по редове, симетрични и положително определени матрици или M-матрици. За тези класове ще докажем, че константите K_2 и r са достатъчно малки и следователно числената устойчивост на алгоритъма ще зависи главно от обусловеността на решаваната лентова система.

Нека да отбележим, че в предишната глава (Раздел 1.3) към посочените тук класове беше добавен и класа от тотално неотрицателните матрици. За съжаление обаче, при разглеждания тук лентов случай, се оказва, че ако A е тотално неотрицателна матрица, то това не означава че редуцираната матрица S е от същия тип. Кратко пояснение на този факт ще направим понататък.

Нека да напомним, че както и в тридиагоналния случай, щом оригиналната матрица A е симетрична и положително определена, с диагонално преобладаване по редове или M-матрица, то и пермутираната матрица $\mathcal{A}$ е от същия тип.

За да намерим оценки на $\|\hat{R}\|_\infty$ и k_2, е необходимо да покажем че редуцираната матрица S запазва свойствата на изходната матрица A. Това ще направим най-напред в точна аритметика, т.е. ще считаме, че редуцираната матрица S е пресметната точно. След като покажем, че S запазва свойствата на A, това ще ни даде възможност да използваме известната оценка (2.35) за ръста на елементите при решаване по метода на Гаус на лентови системи от посочените специални видове. Така ръстът на константата k_2 ще бъде ограничен. Що се отнася до $\|\hat{R}\|_\infty$, то при точно пресметната S, ще намерим оценки за този член, зависещи от обусловеността на A. Накрая естествено ще изведем и оценка за общата грешка в матрицата S, при което ще покажем, че тази грешка зависи отново главно от обусловеността на матрицата A. Следователно, щом A е добре обусловена, компютърно пресметнатата редуцирана матрица ще бъде достатъчно близка до точната и всички наши разсъждения ще запазят своята валидност.

Имайки предвид Теорема 1.2, остава да докажем, че ако A е матрица с диагонално преобладаване по редове, то и S е от същия тип. Случаят, когато A е матрица с блочно диагонално пробладаване по редове е разгледан в [36], където е показано, че и S е от същия тип. Тук ние ще разширим класа от матрици с блочно диагонално преобладаване по редове, като разгледаме матрици със стандартно диагонално преобладаване по редове.

Теорема 2.2 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена лентова матрица с диагонално преобладаване по редове. Тогава редуцираната матрица S е от същия тип.*

Доказателство. Нека да конструираме матрицата $\mathcal{B}_i^{(1)} = (B_i, \bar{a}_i, \bar{c}_i)$. Очевидно е, че тя притежава свойството на диагонално преобладаване по редове, тъй като по условие A е матрица, притежаваща същото свойство. Въпросът, който възниква е дали $\mathcal{B}_i^{(1)}$ запазва това свойство и след прилагане на Гаусово изключване за обръщане на матрицата B_i? Подобен проблем, в случая на произволна плътна матрица, е изследван в [30], където е показано че диагоналното преобладаване (в класически смисъл) се запазва след правия ход на метода на Гаус (за обратния ход твърдението следва по аналогия). Следователно можем да твърдим, че разлежданата матрица $\mathcal{B}_i^{(1)}$ запазва свойството на диагонално преобладаване по редове и след прилагане на Гаусово изключване за обръщане на матрицата B_i.

Тогава ако означим с $\mathcal{B}_i^{(2)}$ матрицата, която се получава след обръщането на B_i, то съобразявайки се с въведените означения е валидно следното представяне

$$\mathcal{B}_i^{(2)} = (I_{k-j}, q^{(i)}, p^{(i)}). \tag{2.36}$$

Сега ще покажем и че редуцираната матрица S също запазва свойството на диагонално преобладаване по редове. Нека да разгледаме произволен l-ти ред от тази матрица, който да означим по следния начин:

$$0, \ldots, 0, u_1^{(l)}, \ldots, u_j^{(l)}, v_1^{(l)}, \ldots, v_j^{(l)}, w_1^{(l)}, \ldots, w_j^{(l)}, 0, \ldots, 0,$$

и нека без ограничение на общността да предположим, че в този ред елементът, който стои на главния диагонал е $v_1^{(l)}$. Тогава нека да разпишем формулите (2.7), по които се пресмятат елементите на S в скаларна форма (за простота индексите са записани в непълен вид):

$$\begin{aligned}
v_1^{(l)} &= b_1^{(l)} - a^{(l)} p_{1-}^{(l)} - c^{(l)} q_{1+}^{(l)}, \\
v_i^{(l)} &= b_i^{(l)} - a^{(l)} p_{i-}^{(l)} - c^{(l)} q_{i+}^{(l)}, \quad i = 2, \ldots, s-1, \\
u_i^{(l)} &= -a^{(l)} q_{i-}^{(l)}, \quad i = 2, \ldots, s-1, \\
w_i^{(l)} &= -c^{(l)} p_{i+}^{(l)}, \quad i = 1, \ldots, s-2,
\end{aligned} \tag{2.37}$$

където $a^{(l)}, b^{(l)}, c^{(l)} \in \mathcal{R}^{1\times j}$, $p_{i-}^{(l)}, p_{i+}^{(l)}, q_{i-}^{(l)}, q_{i+}^{(l)} \in \mathcal{R}^{j\times 1}$. Вече показахме, че $\mathcal{B}_i^{(2)}$ е матрица с диагонално преобладаване по редове, т.е. изпълнено е

$$\sum_{i=1}^{j} |p_{i-}^{(l)}| + \sum_{i=1}^{j} |q_{i-}^{(l)}| \leq e, \tag{2.38}$$

$$\sum_{i=1}^{j} |p_{i+}^{(l)}| + \sum_{i=1}^{j} |q_{i+}^{(l)}| \leq e, \tag{2.39}$$

където векторът $e = (1, 1, \ldots, 1)^T$ е от размерност j. Сега като използваме формулите (2.37) за пресмятане на елементите на S, а също и (2.38), (2.39), получаваме

$$\begin{aligned}
|v_1| &\geq |b_1^{(l)}| - |a^{(l)}||p_{1-}^{(l)}| - |c^{(l)}||q_{1+}^{(l)}| \\
&\geq |b_1^{(l)}| - |a^{(l)}| \left(e - \sum_{i=2}^{j} |p_{i-}^{(l)}| - \sum_{i=1}^{j} |q_{i-}^{(l)}| \right) - |c^{(l)}| \left(e - \sum_{i=1}^{j} |p_{i+}^{(l)}| - \sum_{i=2}^{j} |q_{i+}^{(l)}| \right) \\
&\geq |b_1^{(l)}| - |a^{(l)}|e + |a^{(l)}| \sum_{i=2}^{j} |p_{i-}^{(l)}| + |a^{(l)}| \sum_{i=1}^{j} |q_{i-}^{(l)}| \\
&\quad -|c^{(l)}|e + |c^{(l)}| \sum_{i=1}^{j} |p_{i+}^{(l)}| + |c^{(l)}| \sum_{i=2}^{j} |q_{i+}^{(l)}|
\end{aligned}$$

$$\geq |b_1^{(l)}| - \sum_{i=2}^{j} |b_i^{(l)}| - |a^{(l)}|e - |c^{(l)}|e + \sum_{i=2}^{j} |v_i^{(l)}| + \sum_{i=1}^{j} |u_i^{(l)}| + \sum_{i=1}^{j} |w_i^{(l)}|, \quad (2.40)$$

но тъй като A е матрица с диагонално преобладаване по редове е валидно

$$|b_1^{(l)}| - \sum_{i=2}^{j} |b_i^{(l)}| - |a^{(l)}|e - |c^{(l)}|e \geq 0. \quad (2.41)$$

Тогава от (2.40) и (2.41) получаваме

$$|v_1^{(l)}| \geq \sum_{i=2}^{j} |v_i^{(l)}| + \sum_{i=1}^{j} |u_i^{(l)}| + \sum_{i=1}^{j} |w_i^{(l)}|,$$

което всъщност показва, че свойството на диагонално преобладаване по редове се удовлетворява за произволен ред на матрицата S. Следователно, можем да направим и извода, че S е също матрица с диагонално преобладаване по редове, както и A. $\Diamond$

Така, въз основа на Теорема 1.2 и Теорема 2.2, можем да направим извода, че S е от същия тип, както и A. Но както вече отбелязахме, за разглежданите три специални класа от матрици, методът на Гаус е устойчив. Следователно при решаване на редуцираната система ръстът на елементите е ограничен, т.е. K_2 е малка. Разбира се тук става въпрос за точната матрица S. Ясно е, че компютърно пресметнатата редуцирана матрица се различава от точната. Ето защо в края на този раздел ще изведем и оценка за общата грешка в S, като ще покажем, че тази грешка зависи главно от обусловеността на A.

Нека сега да отбележим, че в тридиагоналния случай (предишната глава) беше разгледан и специалният клас на тотално неотрицателните матрици. За съжаление обаче, в лентовия случай се оказва че щом A е тотално неотрицателна матрица, то това не е изпълнено за редуцираната матрица. Това лесно можем да проверим като за конкретен пример изходим от дефиницията за знаково регулярни матрици. Нека най-напред подобно на (1.44) да намерим $\mathcal{A}^{-1}$

$$\begin{aligned} \mathcal{A}^{-1} = U^{-1}L^{-1} &= \begin{pmatrix} I_{s(k-j)} & -RS^{-1} \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} A_{11}^{-1} & 0 \\ -A_{21}A_{11}^{-1} & I_{j(s-1)} \end{pmatrix} \\ &= \begin{pmatrix} A_{11}^{-1} + H & -RS^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{pmatrix}, \end{aligned} \quad (2.42)$$

където $H = RS^{-1}A_{21}A_{11}^{-1} = A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1}$. Нека сега за конкретните стойности $k = 8, s = 4, j = 2$ да формираме матрицата $\mathcal{A}^{-1}$ и като вземем предвид спецификата на алгоритъма в лентовия случай, да отчетем направените разместванията на нейните редове и стълбове. Тогава за матрицата

S^{-1}, намираща се в долния десен ъгъл на $\mathcal{A}^{-1}$ записана в знакова форма, получаваме

$$\begin{pmatrix} + & - & - & + & + & - \\ - & + & + & - & - & + \\ - & + & + & - & - & + \\ + & - & - & + & + & - \\ + & - & - & + & + & - \\ - & + & + & - & - & + \end{pmatrix}.$$

Следователно, за конкретния пример S^{-1} не е знаково регулярна матрица, което пък автоматично означава, че S не е тотално неотрицателна. Ето защо в крайна сметка класът на тотално неотрицателните матрици отпада от разглежданията в лентовия случай.

Както видяхме в Теорема 2.1, оценката за правата грешка в решението зависи не само от константите K_1 и K_2, но също и от множителя r, който измерва ръста на елементите при пресмятане на $\hat{R}$. В случай обаче, когато някои от блоковете B_i са лошо обусловени (макар цялата матрица A да е добре обусловена), това би могло да доведе до значителен ръст на елементите в $\hat{R}$, което ще означава, че $\|\hat{R}\|_\infty$ става голяма, т.е. множителят r нараства. Това пък от своя страна може да доведе до големи грешки за добре обусловени матрици. Следователно, необходимо е да оценим r (респективно $\|\hat{R}\|_\infty$). Понататък ще покажем, че $\|\hat{R}\|_\infty$ е ограничена от не големи константи в случай на гореспоменатите класове от матрици. За тази цел ще ни бъде необходима следната лема:

Лема 2.4 *Нека е изпълнено, че* $k_1 cond(A_{11}) f(\rho_0) < 1$. *Тогава е валидна следната оценка*

$$\|\hat{R}\|_\infty \leq \frac{\|R\|_\infty}{1 - k_1 cond(A_{11}) f(\rho_0)}.$$

Доказателство. Доказателството на тази лема е напълно аналогично на това направено на Лема 1.4. $\Diamond$

Следователно, за да намерим оценки за $\|\hat{R}\|_\infty$, понататък ще бъде достатъчно да намерим такива само за $\|R\|_\infty$. Освен това от Лема 2.4 се вижда, че оценката за $\|\hat{R}\|_\infty$ зависи от големината на $cond(A_{11})$. Ето защо е небходимо да получим оценки и за това число на обусловеност. По-точно ще покажем също, че A_{11} е по-добре обусловена от A за разглежданите специални класове от матрици.

Теорема 2.3 *Нека* $A \in \mathcal{R}^{n\times n}$ *е неособена лентова M-матрица и нека*

$k_1 cond(A) f(\rho_0) < 1$. *Тогава е валидна следната оценка:*

$$\|\hat{R}\|_\infty \leq \frac{cond(A)}{1 - k_1 cond(A_{11}) f(\rho_0)} \leq \frac{cond(A)}{1 - k_1 cond(A) f(\rho_0)}.$$

Доказателство. Доказателството на тази теорема е напълно аналогично на това направено на Теорема 1.5 (b). ◊

Теорема 2.4 *Нека $A \in \mathcal{R}^{n\times n}$ е неособена лентова матрица и нека $k_1 cond(A) f(\rho_0) < 1$. Тогава ако A е и матрица с диагонално преобладаване по редове, то е изпълнено*

$$\|\hat{R}\|_\infty \leq \frac{1}{1 - k_1 cond(A_{11}) f(\rho_0)} \leq \frac{1}{1 - 2k_1 cond(A) f(\rho_0)}.$$

Доказателство. Доказателството на тази теорема е напълно аналогично на това направено на Теорема 1.6. ◊

Теорема 2.5 *Нека $A \in \mathcal{R}^{n\times n}$ е лентова матрица и нека $k_1(k-1) cond_2(A_{11}) f(\rho_0) < 1$, където $cond_2(A) = \|A^{-1}\|_2 \|A\|_2$. Тогава ако A е и симетрична положително определена, то*

$$\|\hat{R}\|_\infty \leq \frac{\sqrt{j(s-1) cond_2(A)}}{1 - k_1 cond(A_{11}) f(\rho_0)} \leq \frac{\sqrt{j(s-1) cond_2(A)}}{1 - k_1(k-1) cond_2(A) f(\rho_0)}.$$

Доказателство. Доказателството на тази теорема е напълно аналогично на това направено на Теорема 1.7. Единственото нещо, което трябва да се съобрази е новата размерност на матрицата R, поради което е валидна следната оценка:

$$\|R\|_\infty \leq \sqrt{j(s-1)} \|R\|_2. \qquad \Diamond$$

Теореми 2.3 - 2.5 показват, че $\|\hat{R}\|_\infty$ е ограничена от не големи константи за разглежданите три класа от матрици, стига матрицата A да е добре обусловена. Същото нещо се отнасяше и до константата K_2. Но както отбелязахме, това важи за точната матрица S. На практика обаче, трябва да се има предвид изчислената матрица $\hat{S}$. Ето защо е важно да се знае каква е разликата между S и $\hat{S}$. Понататък в Теорема 2.6 е даден отговор на този въпрос.

Теорема 2.6 *За общата грешка $\Omega S = \hat{S} - S$ в изчислената редуцирана матрица $\hat{S}$ е изпълнено*

$$\frac{\|\Omega S\|_\infty}{\|S\|_\infty} \leq K_1 cond(A) r f(\rho_0).$$

Доказателство. Доказателството на тази теорема е напълно аналогично на това направено на Теорема 1.8. ◊

Така в резултат на представените в този раздел изследвания можем да направим извода, че щом матрицата на решаваната лентова система принадлежи към някои от разглежданите специални класове от матрици, то алгоритъмът е числено устойчив. Същият резултат е получен и за метода на Гаус в [36]. Това показва, че и за решаване на лентови системи паралелният алгоритъм на Уанг може да бъде използван безопасно в същите случаи, както и методът на Гаус, което е едно добро изключение от общото хипотетично правило, че паралелните алгоритми са по-неустойчиви от последователните за една и съща задача.

2.4 Числени експерименти

Числените експерименти в този раздел са направени, използвайки програмната среда MATLAB, където машинната точност е $\rho_0 \approx 2.22\text{E}{-16}$. Грешките в решението, които са измервани, при компютърна реализация на изследвания алгоритъм са същите, както в Раздел 1.5

1. Правата грешка в относителен смисъл

$$FE = \frac{\|\hat{x} - x\|_\infty}{\|\hat{x}\|_\infty},$$

където $\hat{x}$ е изчисленото решение.

2. Покомпонентната обратна грешка в елементите на ΔA отново в относителен смисъл (виж [36]) по формулата

$$BE = \max_{1 \le i \le n} \frac{(|A\hat{x} - d|)_i}{(|A||\hat{x}| + |d|)_i}.$$

ε	1E–5	1E–10	1E–15
BE	9.09E–12	2.38E–7	0.04
FE	3.64E–11	8.52E–7	0.15

Таблица 2.1: Права и обратна грешка в Пример 2.1, при $k = 6, s = 10, j = 2$, за различни стойности на ε.

Да разгледаме следните примери:

Пример 2.1 Нека в качеството на A да вземем следната петдиагонална матрица (ширината на лентата $j = 2$):

$$A = \begin{pmatrix} \varepsilon & \varepsilon & 1 & & & & \\ -\varepsilon & \varepsilon & \varepsilon & 1 & & & \\ 1 & -\varepsilon & \varepsilon & \varepsilon & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -\varepsilon & \varepsilon & \varepsilon & 1 \\ & & & 1 & -\varepsilon & 2 & \varepsilon \\ & & & & 1 & -\varepsilon & 2 \end{pmatrix}.$$

Фиксирайки последните два елемента по главния диагонал равни на 2 A става много добре обусловена. Точното решение избираме отново $x = (1, 1, \ldots, 1)^T$. Ефектът при компютърна реализация на този пример е същия, както в Пример 1.1, т.е. получава се, че правата и обратна грешка нарастват значително, когато $\varepsilon \to 0$, макар че A е много добре обусловена и във всички етапи на алгоритъма Гаусовото изключване е направено с частичен избор на главен елемент, т.е. костантите k_1 и k_2 са малки. Обяснението на това е отново, че $\|\hat{R}\|_\infty$ нараства неограничено, когато $\varepsilon \to 0$. Резултатите за различни стойности на ε са представени в Таблица 2.1. Основният извод от този пример е, че той потвърждава теоретично направените изводи в часта им относно константата r, ограничаваща ръста на елементите при пресмятане на $\hat{R}$. В лентовия случай отново е валиден въпросът: "Какво става, когато заедно с $\varepsilon \to 0$ имаме и $cond^*(A, x^*) \to 0$?" Отговор на този въпрос е даден в следващия пример.

k	6	56	256	556
BE	1.274E–16	3.23E–16	1.46E–16	2.38E–16
FE	2.24E–15	8.88E–15	4.28E–15	1.54E–14

Таблица 2.2: Права и обратна грешка в Пример 2.2, при $s = 10, j = 2$ и $\varepsilon = 1E - 16$, за различни стойности на k.

Пример 2.2 Нека A е матрицата от Пример 2.1, при което сме избрали $\varepsilon =$ 1E-16 (число по-малко от машинната точност). Следователно A отново е много добре обусловена. За да осигурим $cond^*(A, x^*) \approx 0$, нека точното решение да изберем по следния начин:

$$x = (e, nil; e, nil; \ldots, e, nil)^T,$$

където $e = (1, \ldots, 1) \in \mathcal{R}^{1\times(k-j)}, nil = (0, \ldots, 0) \in \mathcal{R}^{1\times j}$. При компютърното решаване на поставената задача отново във всички етапи на алгоритъма

Гаусовото изключване е направено с частичен избор на главен елемент, т.е. k_1 и k_2 са малки. В същото време $r \approx 1E+16$), но както можем да видим в Таблица 2.2 грешките в решението са много малки. Причината за това е, че $cond^*(A, x^*) \approx 0$, откъдето и влиянието на r не е съществено, макар че блоковете B_i са почти особени, т.е. и A_{11} е почти особена. Следователно, както в тридиагоналния, така и в лентовия случай (теоретично и на практика), големи r не водят непременно до големи грешки.

x	x_α	e	$rand$	$randn$
BE	3.62E–16	2.58E–16	4.04E–16	1.44E–16
FE	3.54E–11	2.28E–12	5.15E–11	8.08E–11
$cond(A, \hat{x})$	$3.11E+4$	$6.24E+4$	$7.56E+4$	$7.23E+4$
$cond^*(A, x^*)$	$4.54E+5$	$1.88E+4$	$2.32E+5$	$4.38E+5$
r	8.98	8.98	8.98	8.98

Таблица 2.3: Права и обратна грешка в Пример 2.3, при $k = 60, s = 8, j = 2$, за различни точни решения.

Пример 2.3 Нека сега A да има следния вид:

$$A = \begin{pmatrix} 4 & -1 & -1 & & & \\ -1 & 4 & -1 & -1 & & \\ -1 & -1 & 4 & -1 & -1 & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & -1 & 4 & -1 \\ & & & -1 & -1 & 4 \end{pmatrix}.$$

Тази матрица е една симетрична и положително определена с нестрого диагонално преобладаване матрица. Имайки предвид теоретично направените изводи, трябва да очакваме, че каквито и точни решения да избираме, грешката в решението ще зависи главно от обусловеността на матрицата. Нека да изберем различни точни решения по същия начин, както в Пример 1.3: $x_\alpha = (1, \alpha, \alpha^2, \ldots, 10^{-5})^T$, $\alpha = 10^{-5/(n-1)}$, $e = (1, 1, \ldots, 1)^T$, 'rand' и 'randn', където последните две са генерирани от съответните фукции в MATLAB. Получените резултати са представени в Таблица 2.3. Прави впечатление, че обратната грешка BE е малка. Това се дължи на специалния вид, който има матрицата A. В същото време, обаче, правата грешка FE е по-голяма. Обяснението е, че матрицата A е не така добре обусловена, както може да бъде видяно в Таблица 2.3. Освен това можем да видим, че правата грешка е много близка до теоретичната оценка получена в Теорема 2.1, което показва, че тази оценка е почти достижима.

Глава 3

Един по-устойчив вариант на метода на цикличната редукция за решаване на тридиагонални системи линейни уравнения

Основните резултати, представени в тази глава, са публикувани в статиите:

- Pavlov, V., P. Yalamov. Stabilization by Perturbation of Ill-Conditioned Cyclic Reduction. *International Journal of Computer Mathematics*, 68 (1998), 273–283.

- Pavlov, V. Iterative Refinement for Ill-Conditioned Cyclic Reduction. *Proc. Fifth International Conference on Differential Equations and Applications*, (Eds. S. Bilchev and S. Tersian), Rousse, August 24–29, 1995, 84–95.

Един подходящ и често използван метод за паралелно решаване на тридиагонални системи линейни уравнения е този на цикличната редукция (ЦР) [37]. Същността на алгоритъма (изложение може да се намери и в [1]) на този метод се изразява в изключване на четните (или нечетните) редове на системата на всяка стъпка, в резултат на което се получава нова система относно само нечетните (четните) неизвестни, от размерност половината от размерността на изходната система. И ако n е размерността на решаваната система, то след извършването на $O(\log_2 n)$ такива стъпки се стига до едно уравнение с едно неизвестно, откъдето посредством обратна субституция се намират всички останали неизвестни. В [21] е представена още една версия на метода на цикличната редукция в случай на симетрична матрица с постоянни коефициенти. Някои изследвания върху устойчивостта на тази версия са направени по-късно в [22]. В

последната работа ръстът на елементите е ограничен и така би могло да се очаква, че грешката в решението на системата зависи главно от обусловеността на решаваната система, но за съжаление липсва точна оценка за нея.

Съществува една модификация на метода на цикличната редукция [38], при която на всяка стъпка се намира матрица от специален вид, такава че умножавайки с нея изходната матрица (естествено и дясната част) се получава нова матрица с изместени ненулеви диагонали под и над главния диагонал по направление съответно към долния ляв и горния десен ъгъл на изходната матрица. След извършването на $\lceil \log_2 n \rceil$ такива стъпки (умножения) тези диагонали изчезват и се получава диагонална матрица. Това означава, че предлаганата модификация няма обратен ход, т.е. директно се получава решението на системата.

Предимства при паралелна реализация на модифицирания вариант на метода цикличната редукция, в сравнение с оригиналния, са по-малкия брой паралелни стъпки, поради липса на обратна субституция, както и по-равномерното натоварване на отделните процесори, и следователно в крайна сметка по-добра ефективност. Числената устойчивост на тази модификация, при предположение, че матрицата на разглежданата система е неособена е изследвана в [69]. В тази работа най-напред са изведени оценки за относителните еквивалентни смущения във входните данни при пресмятането поотделно на всяка компонента на решението. След това е направен покомпонентен прав анализ, като в крайна сметка е получена оценка за правата грешка в решението на системата в покомпонентен смисъл. Тази оценка зависи от ръста на елементите и от обусловеността на решаваната система. Понататък ръстът на елементите е ограничен в случай, че изходната матрица принадлежи на някои от следните специални класове: матрици с диагонално преобладаване, симетрични и положително определени, М-матрици или тотално неотрицателни матрици. Откъдето е направен изводът, че за разглежданите специални класове числената устойчивост на алгоритъма зависи главно от обусловеността на решаваната система. Освен това от направените в [69] изследвания, може да се направи и изводът, че числената устойчивост на модифицирания метод на цикличната редукция е доста близка до тази на метода на Гаус (виж [35], където е представен аналогичен анализ). Следователно, разглежданият паралелен метод на цикличната редукция на практика може да бъде използван в същите случаи, както и методът на Гаус (известен като последователен такъв).

Проблеми с реализацията на модифицирания метод на цикличната редукция възникват, когато решаваната система е добре обусловена, но нейната матрица не принадлежи към отделените в [69] специални класове

от матрици. Тогава е възможно да се получи прекъсване на алгоритъма (поради препълване при деление на нула) или да се получи взрив на грешките от закръгляване (при деление на числа близки до нулата). В такива случаи ако се приложи вариант за решаване на системата с избор на главен елемент, това би довело до нарушаване на нейната структура и до много комуникации между отделните процесори. Ето защо е желателно да се избегне подхода на избор на главен елемент и да се потърсят други, по-прости за реализация подходи за подобряване на числената устойчивост на алгоритъма. За решаването на възникналия проблем в настоящата глава отново се прилага, вече използваният в Глава 1 подход на изкуствено смущаване на някои данни.

Главата съдържа шест раздела. В Раздел 3.1 е направено описание на модифицирания алгоритъм на цикличната редукция за решаване на тридиагонални системи линейни уравнения. В следващия раздел е представен стабилизираният вариант на същия алгоритъм. В Раздел 3.3 е изследвана неговата числена устойчивост. Полученото допълнително смутено решение се доуточнява посредством процедурата за итерационно уточняване, представена в Раздел 3.4. Специално внимание върху предимствата при прилагане на предложения стабилизиран алгоритъм за решаване на системи линейни уравнения с много десни части, е отделено в Раздел 3.5. В края на главата (Раздел 3.6) са представени множество числени експерименти, включително и със случайни матрици, които показват,че стабилизираният вариант на алгоритъма работи значително по-добре от оригиналния, а също и че за итерационното уточняване на решението са необходими само една-две итерации.

Нека да отбележим, че разглежданият в тази глава модифициран алгоритъм на цикличната редукция може да бъде обобщен в блочен вид (за решаване на лентови системи линейни уравнения) и това е направено в Глава 4. В същата глава е представено и пълно изследване на числената устойчовост на получения в този случай алгоритъм.

3.1 Описание на алгоритъма

Нека да разгледаме следната тридиагонална система линейни уравнения

$$Ax = d, \tag{3.1}$$

където

$$A = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix},$$

при което A е добре обусловена тридиагонална матрица. Понататък за простота в означенията ще предполагаме, че когато $i \leq 0$ или $i > n$ е изпълнено

$$a_i = c_i = d_i = 0, \ b_i = 1,$$

а също и, че $a_1 = c_n = 0$.

Нека да означим $m = \lceil \log_2 n \rceil$. Както вече отбелязахме, алгоритъмът се състои от $m+1$ стъпки, при което за m стъпки чрез подходящи умножения от ляво матрицата A се преобразува в диагонален вид и на $(m+1)$-вата стъпка се намира решението на системата. В аналитичен вид тези стъпки бихме могли да опишем по следния начин. Последователно за $k = 1, 2, \ldots, m$ се извършват следните действия:

1. Изчисляване на елементите

$$\alpha_i^{(k)} = -\frac{a_i^{(k-1)}}{b_{i-2^{k-1}}^{(k-1)}}, \tag{3.2}$$

$$\beta_i^{(k)} = -\frac{c_i^{(k-1)}}{b_{i+2^{k-1}}^{(k-1)}}, \tag{3.3}$$

$$i = 1, 2, \ldots, n,$$

на матрицата

$$L^{(k)} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \beta_1^{(k)} & & & & \\ 0 & \ddots & & & & \ddots & & & \\ \vdots & & \ddots & & & & \ddots & & \\ 0 & & & \ddots & & & & \ddots & \\ \alpha_{2^{k-1}+1}^{(k)} & & & & \ddots & & & & \beta_{n-2^{k-1}}^{(k)} \\ & \ddots & & & & \ddots & & & 0 \\ & & \ddots & & & & \ddots & & \vdots \\ & & & \ddots & & & & \ddots & 0 \\ & & & & \alpha_n^{(k)} & 0 & \cdots & 0 & 1 \end{pmatrix}, \ .$$

с която се умножава от ляво получените на предишната стъпка матрица $A^{(k-1)}$ и дясна част $d^{(k-1)}$ (за $k = 1$, $A^{(0)} = A$, $d^{(0)} = d$).

2. С помощта на така конструираната матрица $L^{(k)}$ получаваме

$$A^{(k)} = L^{(k)}A^{(k-1)}, \; d^{(k)} = L^{(k)}d^{(k-1)},$$

където елементите на $A^{(k)}$ и $d^{(k)}$ се намират по следните формули:

$$\begin{aligned} a_i^{(k)} &= \alpha_i^{(k)} a_{i-2^{k-1}}^{(k-1)}, \\ c_i^{(k)} &= \beta_i^{(k)} c_{i+2^{k-1}}^{(k-1)}, \\ b_i^{(k)} &= b_i^{(k-1)} + \alpha_i^{(k)} c_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} a_{i+2^{k-1}}^{(k-1)}, \\ d_i^{(k)} &= d_i^{(k-1)} + \alpha_i^{(k)} d_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} d_{i+2^{k-1}}^{(k-1)}, \\ i &= 1, 2, \ldots, n. \end{aligned} \tag{3.4}$$

В резултат ненулевите диагонали под и над главния диагонал на новата матрица $A^{(k)}$ са се преместили в посока съответно към долния ляв и горния десен ъгъл. При това броят на нулевите диагонали между тях и главния диагонал е $2^k - 1$. По този начин точно след m стъпки се стига до диагоналната матрица

$$A^{(m)} = \begin{pmatrix} b_1^{(m)} & & \\ & \ddots & \\ & & b_n^{(m)} \end{pmatrix}.$$

На $(m+1)$-вата намираме решението на системата (3.1)

$$x = L^{(m+1)}d^{(m)},$$

където

$$L^{(m+1)} = \begin{pmatrix} \frac{1}{b_1^{(m)}} & & \\ & \ddots & \\ & & \frac{1}{b_n^{(m)}} \end{pmatrix},$$

или по-точно

$$x_i = d_i^{(m)}/b_i^{(m)}, \quad i = 1, 2, \ldots, n.$$

Нека да отбележим, че при изчисляване на елементите на матрицата $L^{(k)}$ по формулите (3.2) и (3.3) е достатъчно някой от елементите $b_{i-2^{k-1}}^{(k-1)}$ или $b_{i+2^{k-1}}^{(k-1)}$ да стане равен на нула и алгоритъмът да прекъсне, или пък близък до нула и да се получи взрив на грешките от закръгляване. В следващия раздел ще дадем отговор на въпроса как да избегнем тази неприятна ситуация.

3.2 Стабилизиран вариант на алгоритъма

В този раздел, използвайки подхода със смущения, ще стабилизираме описания в предишния раздел алгоритъм. Нека да се спрем по-подробно върху случаите, когато се налага да се прилага този подход. Вече отбелязахме, че алгоритъмът може да прекъсне или да се получи взрив на грешките от закръгляване, когато някой от елементите $b_{i-2^{k-1}}^{(k-1)}$ или $b_{i+2^{k-1}}^{(k-1)}$ стане равен или близък до нула. Тогава, нека в такива случаи да смутим изкуствено тези елементи с достатъчно малко число δ_0, така че да ги отдалечим достатъчно от нулата. Стабилизираният вариант на алгоритъма е представен на Фигура 3.1.

Този подход гарантира, че елементите $b_{i-2^{k-1}}^{(k-1)}$ или $b_{i+2^{k-1}}^{(k-1)}$ ще бъдат достатъчно отдалечени от нулата, така че няма да се налага да се дели на числа близки до нулата. В резултат обаче, получаваме и допълнително смутено решение $\hat{x}$, т.е. влияние върху решението оказват, както грешките от закръгляване, така и направените от нас допълнителни смущения. Полученото решение $\hat{x}$ може да се направи по-близко до точното ако се използва процедура на итерационно уточняване (Раздел 3.4).

Нека, с цел да оценим направените в $b_j^{(k-1)}$ смущения, да ги изразим като смущения във входните данни. Щом смущаваме $b_j^{(k-1)}$ посредством $\mathrm{Sgn}(b_j^{(k-1)})\delta_0$, тогава, използвайки (3.4) можем да изразим това смущение като смущение в b_j. Нека да означим посредством Δ матрицата от всички такива смущения, които са всъщност смущения и в изходната матрица A. От факта, че на практика се смущават само някои от диагоналните елементи на A следва, че Δ е диагонална матрица, с някои ненулеви диагонални елементи. Тогава за Δ е валидна следната оценка:

$$|\Delta| \leq C\,\delta_0\,I, \tag{3.5}$$

където на практика константата $C = \mathcal{O}(1)$ (обикновено $C = 1$ или $C = 2$), а I е единичната матрица.

3.3 Изследване на числената устойчивост на стабилизирания алгоритъм

Както отбелязахме, числената устойчивост на оригиналния алгоритътм (без стабилизация), е изследвана подробно в [69], където за оценка на грешките от закръгляване, авторът прилага метода представен в [60, 70]. В настоящия раздел, използвайки получения в тази работа анализ, ще

for $k = 1, \ldots, m$
  for $i = 1, \ldots, n$

    if $(|b_{i-2^{k-1}}^{(k-1)}| < \delta_0)$

      $b_{i-2^{k-1}}^{(k-1)} = b_{i-2^{k-1}}^{(k-1)} + \mathrm{Sgn}(b_{i-2^{k-1}}^{(k-1)})\delta_0;$

    end

    if $(|b_{i+2^{k-1}}^{(k-1)}| < \delta_0)$

      $b_{i+2^{k-1}}^{(k-1)} = b_{i+2^{k-1}}^{(k-1)} + \mathrm{Sgn}(b_{i+2^{k-1}}^{(k-1)})\delta_0;$

    end

    $\alpha_i^{(k)} = -a_i^{(k-1)}/b_{i-2^{k-1}}^{(k-1)};$

    $\beta_i^{(k)} = -c_i^{(k-1)}/b_{i+2^{k-1}}^{(k-1)};$

    $a_i^{(k)} = \alpha_i^{(k)} a_{i-2^{k-1}}^{(k-1)};$

    $c_i^{(k)} = \beta_i^{(k)} c_{i+2^{k-1}}^{(k-1)};$

    $b_i^{(k)} = b_i^{(k-1)} + \alpha_i^{(k)} c_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} a_{i+2^{k-1}}^{(k-1)};$

    $d_i^{(k)} = d_i^{(k-1)} + \alpha_i^{(k)} d_{i-2^{k-1}}^{(k-1)} + \beta_i^{(k)} d_{i+2^{k-1}}^{(k-1)};$

  end
end
for $i = 1, \ldots, n$

  $x_i = d_i^{(m)}/b_i^{(m)};$

end

където $\mathrm{Sgn}(b_j) = \begin{cases} \mathrm{sign}(b_j), & \text{ако } b_j \neq 0, \\ 1, & \text{ако } b_j = 0, \quad j = 1, \ldots, n. \end{cases}$

Фигура 3.1: Стабилизиран вариант на модифицирания алгоритъм на циклличната редукция.

изследваме и числената устойчивост на стабилизирания вариант на алгоритъма.

В [69] е използван подход на обратен анализ. При това нас по-специално ни интересува оценката за еквивалентните смущения в елементите $b_i^{(k-1)}$, стоящи по главния диагонал на матрицата $A^{(k-1)}$, тъй като както отбелязахме, при необходимост ние смущаваме допълнително някои от тези елементи. При това нашата цел е да изразим тези допълнителни смущения, като такива във входните данни. И така, интересуващата ни оценка има вида

$$|\rho_{b_i^{(k-1)}}| \leq \sum_{j=k}^{m} K_1^{j-k}[5K_2(m-j)+K_1+10K_2]\rho_0,$$

където с $\rho_{b_i^{(k-1)}}$ е означено относителното еквивалентно смущение в $b_i^{(k-1)}$, а константите K_1, K_2 са такива, че

$$|b_i^{(k)}/b_i^{(k-1)}| \leq K_1, \quad (|t_1^{(ik)}|+|t_2^{(ik)}|)/|b_i^{(k-1)}| \leq K_2, \tag{3.6}$$

където

$$t_1^{(ik)} = \alpha_i^{(k)} c_{i-2^{k-1}}^{(k-1)}, \qquad t_2^{(ik)} = \beta_i^{(k)} a_{i+2^{k-1}}^{(k-1)}. \tag{3.7}$$

Понататък в [69], изхождайки от получените оценки за еквивалентните смущения, е изведена покомпонентна оценка за правата грешка в решението на системата. След това константите K_1, K_2 са ограничени от горе в случай на някои специални класове от матрици: матрици с диагонално преобладаване, симетрични и положително определени матрици, M-матрици, тотално неотрицателни матрици и е показано, че тези ограничения са относително малки (в смисъл $C(n)\rho_0$, където $C(n)$ е бавно растяща функция на n). Както вече стана ясно, в тази глава ние разглеждаме случая на произволна добре обусловена система. Следователно ще бъде интересно и какви са ограниченията за K_1, K_2 в този по-общ случай, при решаване на системата (3.1), използвайки стабилизирания алгоритъм.

Нека в резултат на числена реализация на този алгоритъм сме получили компютърно решение означено с $\hat{x}$. Целта, която си поставяме е да оценим грешката $|\hat{x}_l - x_l|$ в произволна l-та компонента на решението на системата. За осъществяване на тази цел, най-напред ще изразим грешката в тази l-та компонента, като еквивалентно смущение във входните данни, т.е. ще използваме покомпонентен обратен анализ. При това ще считаме, че изчисляваме само l-та компонента, използвайки стабилизирания алгоритъм (отчитат се грешките от закръгляване и допълнителните смущения), при което останалите компоненти на решението могат да бъдат различни от съответните им компоненти в $\hat{x}$. Ако така конструираното решение означим с $\hat{x}^{(l)}$, тогава $\hat{x}_l^{(l)} = \hat{x}_l$.

Забележка. Такъв подход за първи път е използван в [54] за изследване на числената устойчивост на метода на Гаус-Жордан при решаване на системи линейни уравнения.

На практика, в резултат на смущенията, които добавяме при стабилизацията на алгоритъма, всъщност системата, която решаваме има матрица $A+\Delta$. От тук, ако разгледаме системата, на която всъщност е решение $\hat{x}^{(l)}$, то тя има следния вид:

$$(A+\Delta+\varepsilon_A^{(l)})\hat{x}^{(l)} = d+\varepsilon_d^{(l)}, \tag{3.8}$$

където $\varepsilon_A^{(l)}$ и $\varepsilon_d^{(l)}$ са абсолютните еквивалентни смущения, получени от изразяването само на грешките при изчислението на $\hat{x}_l$.

Нека да разгледаме частното $b_i^{(k)}/b_i^{(k-1)}$, да изразим числителя от (3.4) и да използваме (3.2), (3.3), (3.7) тогава получаваме

$$\begin{aligned}
|\frac{b_i^{(k)}}{b_i^{(k-1)}}| &= |\frac{b_i^{(k-1)}+\alpha_i^{(k)}c_{i-2^{k-1}}^{(k-1)}+\beta_i^{(k)}a_{i+2^{k-1}}^{(k-1)}}{b_i^{(k-1)}}| \\
&\leq 1+|\frac{t_1^{(ik)}}{b_i^{(k-1)}}|+|\frac{t_2^{(ik)}}{b_i^{(k-1)}}| \\
&= 1+|\frac{a_i^{(k-1)}c_{i-2^{k-1}}^{(k-1)}}{b_{i-2^{k-1}}^{(k-1)}b_i^{(k-1)}}|+|\frac{c_i^{(k-1)}a_{i+2^{k-1}}^{(k-1)}}{b_{i+2^{k-1}}^{(k-1)}b_i^{(k-1)}}|.
\end{aligned} \tag{3.9}$$

Нека сега да предположим, че $|a_j^{(k-1)}| \leq C_1$ и $|c_j^{(k-1)}| \leq C_2$, $j=1,2,\ldots,n$, където C_1, C_2 са някакви константи. Тогава в случая, когато е необходимо да смутим $b_j^{(k-1)}$ за някое j, т.е. $|b_j^{(k-1)}| < \delta_0$, получаваме следните оценки:

$$|\frac{a_i^{(k-1)}}{b_j^{(k-1)}+Sgn(b_j^{(k-1)})\delta_0}| \leq \frac{C_1}{\delta_0}, \qquad |\frac{c_i^{(k-1)}}{b_j^{(k-1)}+Sgn(b_j^{(k-1)})\delta_0}| \leq \frac{C_2}{\delta_0}. \tag{3.10}$$

Горните оценки са валидни и когато не е необходимо смущаване, тъй като тогава $|b_j^{(k-1)}| \geq \delta_0$. Следователно ако заместим (3.10) в (3.9), получаваме

$$|\frac{b_i^{(k)}}{b_i^{(k-1)}}| \leq 1+\frac{2C_1C_2}{\delta_0^2}.$$

Ако сега приложим рекурсивно последната оценка, то константите K_1 и K_2 можем да изберем по следния начин:

$$K_1 = \frac{1}{\delta_0^{s^*}}, \qquad K_2 = \frac{1}{\delta_0^{s^*}},$$

където s^* е неизвестно, при това е много трудно теоретично да го определим точно, но бихме могли приблизително да направим това, базирайки се на практически опит.

Понататък в нашите изследвания ще използваме следната теорема (доказана в [69]).

Теорема 3.1 *При компютърното пресмятане на l-тата компонента $\hat{x}_l$ на решението $\hat{x}$ са валидни следните оценки за относителните еквивалентни смущения във входните данни за тази част от алгоритъма*

$$\begin{aligned} |\rho_{a_j}| &\leq 4m\rho_0, \\ |\rho_{b_j}| &\leq \begin{cases} (K_3(K_1^m - 1) + K_4 m)\rho_0, & \text{за } K_1 \neq 1, \\ (2.5K_2 m^2 + (1 + 7.5K_2)m)\rho_0, & \text{за } K_1 = 1, \end{cases} \qquad (3.11) \\ |\rho_{c_j}| &\leq 4m\rho_0, \\ |\rho_{d_j}| &\leq (m+1)\rho_0, \\ j &= 1, 2, \ldots, n, \qquad (3.12) \end{aligned}$$

където

$$K_3 = 5K_2/(K_1 - 1)^2 + (K_1 + 10K_2)/(K_1 - 1),$$
$$K_4 = 5K_2/(1 - K_1),$$

K_1 и K_2 са такива, че удовлетворяват (3.6). Освен това при пресмятане на решението $\hat{x}^{(l)}$ на системата

$$(A + \varepsilon_A^{(l)})\hat{x}^{(l)} = d + \varepsilon_d^{(l)},$$

за абсолютните еквивалентни смущения $\varepsilon_A^{(l)}$ и $\varepsilon_d^{(l)}$ е в сила

$$|\varepsilon_A^{(l)}| \leq K_5|A|\rho_0, \qquad (3.13)$$
$$|\varepsilon_d^{(l)}| \leq (m+1)|d|\rho_0, \qquad (3.14)$$

където

$$K_5 = \begin{cases} \max\{4m, K_3(K_1^m - 1) + K_4 m\}, & \text{за } K_1 \neq 1, \\ \max\{4m, 2.5K_2 m^2 + (1 + 7.5K_2)m\}, & \text{за } K_1 = 1. \end{cases}$$

Накрая за правата грешка в решението на системата е валидна следната оценка

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq \frac{K\,\mathrm{cond}(A, x)\rho_0}{1 - K\,\mathrm{cond}(A, e)\rho_0}, \quad (\mathrm{K\ cond}(A, e) < 1),$$

където

$$\mathrm{cond}(A, x) = \frac{\| |A^{-1}|\,|A|\,|x| \|_\infty}{\|x\|_\infty}, \qquad e = (1, 1, \ldots, 1)^T,$$
$$K = 2K_5.$$

Нека да поясним, че константата K_1 се появява в (3.11) на степен m, защото в [69] е използвана една и съща константа K_1 на всяка стъпка k.

В този раздел нека да допуснем, че на всяка стъпка са определени различни константи $K_1^{(k)}$ и $K_2^{(k)}$. Тогава вместо K_1^m ще имаме $\prod_{k=1}^{m} K_1^{(k)}$. На практика, тъй като се смущават само някои елементи $b_j^{(k-1)}$, можем да считаме, че е изпълнено

$$\prod_{k=1}^{m} K_1^{(k)} \leq \frac{1}{\delta_0^s}, \tag{3.15}$$

където, както при избора на K_1 и K_2, теоретично s трудно може да бъде оценено. Базирайки се на нашия практически опит и числените експерименти, представени в последния раздел (включително и със случайни матрици) можем да очакваме, че s принадлежи на интервала $0.6 \leq s \leq 1$.

Нека сега да приложим Теорема 3.1 в нашия случай, като това ще направим за матрицата $A + \Delta$, вместо за A, както е в [69].

Да означим с $\delta x^{(l)} = \hat{x}^{(l)} - x$ правата грешка в решението $\hat{x}^{(l)}$. Тогава очевидно за l-тата компонента е изпълнено

$$(\delta x^{(l)})_l = \hat{x}_l^{(l)} - x_l = \hat{x}_l - x_l = (\delta x)_l, \tag{3.16}$$

където $\delta x = \hat{x} - x$. Замествайки $\hat{x}^{(l)} = x + \delta x^{(l)}$ в (3.8) получаваме

$$(A + \Delta + \varepsilon_A^{(l)})(x + \delta x^{(l)}) = d + \varepsilon_d^{(l)},$$

откъдето, ако вземем предвид, че $Ax = d$ и изразим $\delta x^{(l)}$, получаваме

$$\begin{aligned} \delta x^{(l)} &= (A + \Delta + \varepsilon_A^{(l)})^{-1}(-\Delta x - \varepsilon_A^{(l)} x + \varepsilon_d^{(l)}) \\ &= \left(I + A^{-1}(\Delta + \varepsilon_A^{(l)})\right)^{-1} \left(-A^{-1}\Delta x - A^{-1}\varepsilon_A^{(l)} x + A^{-1}\varepsilon_d^{(l)}\right), \end{aligned}$$

и сега, ако е изпълнено $\|A^{-1}(\Delta + \varepsilon_A^{(l)})\|_\infty < 1$, използвайки известното неравенство (виж [67]):

$$\|(I + W)^{-1}\| \leq \frac{1}{1 - \|W\|}, \tag{3.17}$$

където W е произволна матрица, такава че $\|W\| < 1$, стигаме до

$$\|\delta x^{(l)}\|_\infty \leq \frac{\| \, |A^{-1}| \, |x| \, \|_\infty \, \|\Delta\|_\infty + \| \, |A^{-1}| \, |\varepsilon_A^{(l)}| \, |x| \, \|_\infty + \| \, |A^{-1}| \, |\varepsilon_d^{(l)}| \, \|_\infty}{1 - (\| \, |A^{-1}| \, |\Delta| \, \|_\infty + \| \, |A^{-1}| \, |\varepsilon_A^{(l)}| \, \|_\infty)}. \tag{3.18}$$

Тогава ако предположим, че

$$C \, cond(A, e)(\delta_0 + \frac{\rho_0}{\delta_0^s}) < 1, \qquad \| \, |A^{-1}| \, |x| \, \|_\infty \leq const. \| \, |A^{-1}| \, |A| \, |x| \, \|_\infty,$$

и от (3.5), (3.13), (3.14), (3.15), (3.16) и (3.18) получаваме

$$\frac{\|\hat{x}^{(l)} - x\|_\infty}{\|x\|_\infty} \leq \frac{C\, cond(A,x)(\delta_0 + \frac{\rho_0}{\delta_0^s})}{1 - C\, cond(A,e)(\delta_0 + \frac{\rho_0}{\delta_0^s})}, \tag{3.19}$$

където C е константа от ред $\mathcal{O}(1)$.

Забележка. Тук умноженията на C по константи, независещи от n, също са означени с C.

Накрая за правата грешка (в относителен смисъл) от (3.19) получаваме

$$\begin{aligned} \frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} &= \frac{\max_l\{|\hat{x}_l - x_l|\}}{\|x\|_\infty} = \frac{\max_l\{|\hat{x}_l^{(l)} - x_l|\}}{\|x\|_\infty} \\ &\leq \max_l \frac{\|\hat{x}^{(l)} - x\|_\infty}{\|x\|_\infty} \leq \frac{C\, cond(A,x)(\delta_0 + \frac{\rho_0}{\delta_0^s})}{1 - C\, cond(A,e)(\delta_0 + \frac{\rho_0}{\delta_0^s})}. \end{aligned} \tag{3.20}$$

Ако сега минимизираме дясната част на (3.20) по отношение на δ_0, то получаваме $\delta_0 = s\rho_0^{\frac{1}{s+1}}$. Както отбелязахме точната стойност на s трудно може да бъде намерена. Но ако вземем предвид направеното предположение $s \in [0.6, 1]$, за δ_0 получаваме $\delta_0 \in [10^{-11}, 10^{-8}]$ в среда с двойна точност. Както можем да видим в Раздел 3.6, в качеството на оптимална стойност можем да вземем $\delta_0 \approx \rho_0^{5/9} \approx 10^{-9}$ в среда с двойна точност.

Нека да си спомним, че в Глава 1 (Раздел 1.4) с цел стабилизация на разглеждания там алгоритъм, в качеството на оптимална стойност за δ_0 беше препоръчана $\delta_0 = 10^{-8}$. В [32] при подобен подход пък се препоръчва $\delta_0 = 10^{-6}$. Следователно, оптималната стойност за δ_0 зависи от конкретния алгоритъм.

3.4 Итерационно уточняване на решението

Както отбелязахме, за да уточним полученото смутено решение, ще използваме стандартна процедура на итерационно уточнявяне (виж [30]), с малка модификация

$x^{(0)} = \hat{x}$;
for $k = 1, 2, \ldots$
$r^{(k-1)} = b - Ax^{(k-1)}$;
$(A + \Delta)y^{(k)} = r^{(k-1)}$;
$x^{(k)} = x^{(k-1)} + y^{(k)}$;
end

Разликата тук е, че вместо с A решаваме смутената система с матрицата $A+\Delta$. Както можем да видим в горе-описаната процедура на итерационно уточняване се налага да се решава няколко пъти смутената система

$$(A + \Delta)y^{(k)} = r^{(k-1)}. \tag{3.21}$$

За решаване на тази система можем да предложим следните подходи:

1. Решаване отново чрез прилагане на вече използвания стабилизиран алгоритъм.

2. Запазване в паметта на вече изчислените стойности на $\alpha_i^{(k)}$ и $\beta_i^{(k)}$, за всички i и k, с цел използването им за изчисляване на новата дясна част на всяка стъпка.

3. Изчисляване на първия и последния стълб на обратната матрица A^{-1}, прилагайки стабилизирания алгоритъм без итерационно уточняване. Тогава посредством представянето на обратната матрица дадено в Раздел 3.5 намираме решението на смутената система (3.21)

$$y^{(k)} = (A + \Delta)^{-1} r^{(k-1)}.$$

Нека да видим какви са разликите между тези три начина. Първият се нуждае от по-малко памет в сравнение с втория. При втория начин имаме нужда от $\mathcal{O}(n \log_2 n)$ допълнителни места за запомняне на коефициентите $\alpha_i^{(k)}$ и $\beta_i^{(k)}$, но за сметка на това съществено се намалява времето за изчисления. Третият начин е за предпочитане, когато матрицата A е симетрична, тъй като в този случай представянето на A^{-1} е много просто. В несиметричния случай също е възможно да използваме този подход, но ще бъдат направени повече изчисления.

Нека сега да направим кратък анализ на сходимостта на използваната и описана по-горе процедура на итерационно уточняване на решението.

Ако при решаване на (3.21) означим грешката в компютърно пресметнатото $\hat{y}^{(k)}$ с $\delta y^{(k)}$, то ще бъде изпълнено

$$\hat{y}^{(k)} = y^{(k)} + \delta y^{(k)}.$$

Тогава на k-тата итерация ще имаме

$$x^{(k)} = x^{(k-1)} + \hat{y}^{(k)} = x^{(k-1)} + y^{(k)} + \delta y^{(k)}.$$

На практика при числена реализация на описаната процедура на итерационно уточняване при пресмятането на резидуала $r^{(k-1)}$ и $x^{(k)}$ също се правят грешки от закръгляване. Ние обаче ще пренебрегваме тези грешки, тъй като те са от порядък $\mathcal{O}(\rho_0)$ и биха се отразили в крайна сметка в членовете съдържащи δ_0 във втори и по-висок порядък. Както стана ясно от предишния раздел и направеното там предположение $s \in [0.6, 1]$ то за δ_0 ще имаме поне

$$\delta_0 = 0.6\rho_0^{\frac{1}{0.6+1}} = 0.6\rho_0^{0.625}.$$

Така, понататък ще използваме точните стойности за $r^{(k-1)}$ и $x^{(k)}$ вместо изчислените.

Нека да заместим $\delta_0 = s\rho_0^{\frac{1}{s+1}}$ в (3.20), при което да вземем под внимание, че

$$s\rho_0^{\frac{1}{s+1}} + \frac{\rho_0}{s^s \rho_0^{\frac{s}{s+1}}} \leq 3\delta_0, \quad \text{при} \quad 0.6 \leq s \leq 1$$

тогава получаваме

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq \frac{3C\, cond(A, x)\delta_0}{1 - 3C\, cond(A, e)\delta_0}.$$

Тук няма да се спираме подробно върху въпроса за сходимостта на итерационното уточняване, тъй като този въпрос подробно е изследван в [72]. По същия начин, както в [72] можем да получим оценка за сходимостта от следния вид:

$$\|x^{(k)} - x\|_\infty \leq M\|x^{(k-1)} - x\|_\infty,$$

където за константата M е валидна следната оценка:

$$M \leq \frac{C^*\, cond(A, x)\delta_0}{1 - C^*\, cond(A, e)\delta_0}, \qquad C^* = \mathcal{O}(1).$$

Този резултат показва, че когато матрицата A не е лошо обусловена, тогава грешката в решението x се намалява съществено след една итерация на итерационно уточняване. Например, ако $cond(A, x) \approx 10^4$ и $\delta_0 = 10^{-9}$ тогава на всяка итерация грешката се намалява приблизително 10^5 пъти. Така след една или две итерации се намира решение, което е достатъчно близко до точното. Използваният на практика критерий за край е даден в Раздел 3.6.

3.5 Приложение на стабилизирания алгоритъм при решаване на системи с много десни части

В практиката понякога се налага да се решават системи линейни уравнения с две и повече десни части. Подобни задачи възникват не само при итеративно уточняване на решението на системата, но и в редица други случаи. В този раздел ще представим един възможен начин за прилагане на стабилизирания алгоритъм при решаване на тридиагонални системи линейни уравнения с много десни части. При това ще конкретизираме нещата за случая, когато матрицата A е симетрична. Тогава можем да използваме следния подход:

1. Намираме p и q посредством стабилизирания вариант на алгоритъма с итерационно уточняване, където $p = (p_1, \ldots, p_n)^T$ и $q = (q_1, \ldots, q_n)^T$ са съответно първият и последният стълб на обратната матрица A^{-1}.

2. Изчисляване на многото решения посредством прилагане на следната формула (която е дадена в [3] за несиметрични матрици):

$$A^{-1} = \gamma \left[\begin{pmatrix} p_1 & & & \\ & p_2 & & 0 \\ & & \ddots & \\ & 0 & & p_n \end{pmatrix} \begin{pmatrix} 1 & & & & \\ 1 & 1 & & & 0 \\ \vdots & & \ddots & & \\ & & & \ddots & \\ 1 & 1 & 1 & \ldots & 1 \end{pmatrix} \begin{pmatrix} q_1 & & & \\ & q_2 & & 0 \\ & & \ddots & \\ & 0 & & q_n \end{pmatrix} + \right.$$

$$\left. \begin{pmatrix} q_1 & & & \\ & q_2 & & 0 \\ & & \ddots & \\ & 0 & & q_n \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & \ldots & 1 \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ & 0 & & 0 & 1 \\ & & & & 0 \end{pmatrix} \begin{pmatrix} p_1 & & & \\ & p_2 & & 0 \\ & & \ddots & \\ & 0 & & p_n \end{pmatrix} \right],$$

където, отчитайки факта, че в нашия случай A е симетрична матрица, за константата γ имаме $\gamma = 1/p_n = 1/q_1$.

По този начин получаваме един бърз и устойчив метод за решаване на тридиагонални системи линейни уравнения от разглеждания тип с много десни части, при който е достатъчно посредством стабилизираният вариант на алгоритъма да намерим еднократно p и q, след което можем да намерим всички решения посредством матрично-векторни умножения.

Когато A не е симетрична същият подход може да бъде приложен, но тогава алгоритъмът значително се усложнява.

3.6 Числени експерименти

Числените експерименти в този раздел са направени, използвайки програмната среда MATLAB, където машинната точност е $\rho_0 \approx 2.22\text{E}{-16}$. Целта сега ще бъде да покажем какви резултати могат да се получат, когато е наложителна стабилизация на разглеждания алгоритъм и прилагаме подхода със смущения. При числената реализация измерваме правата грешка в решението на системата в относителен смисъл

$$FE = \frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty},$$

където $\hat{x}$ е компютърно пресметнатото решение, влияние върху което оказват грешките от закръгляване и направените от нас допълнителни смущения в някои данни. За да направим това решение по-близко до точното използваме описаната процедура на итерационно уточняване, при което итерациите спират, когато поне едно от следните две условия са изпълнят:

1. $\|Ax^{(k)} - d\|_\infty / \|d\|_\infty \leq 1000\rho_0$;
2. Броят на итерациите е > 10;

В примерите, които ще разгледаме (с изключение на Пример 3.5) в качеството на точно решение избираме $x = (1, 1, \ldots, 1)^T$. Във всички таблици с N_{it} сме означили броя на итерациите необходими за уточняване на решението.

Пример 3.1 Нека матрицата A има вида

$$A = \begin{pmatrix} 2 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & 1 & 0 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & 1 \\ & & & & & 1 & 0 \end{pmatrix}.$$

Фиксирайки $b_1 = 2$ матрицата A става добре обусловена. Получените резултати при различни стойности на n и δ_0 са представени в Таблица 3.1.

Пример 3.2 Нека матрицата A има вида

$$A = \begin{pmatrix} 1 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 1 & 1 & & & \\ & & 1 & 0 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & 1 \\ & & & & & 1 & 1 \end{pmatrix}.$$

За да е добре обусловена тази матрица е достатъчно тя да бъде от нечетен ред. Получените резултати при различни стойности на n и δ_0 са представени в Таблица 3.2.

Пример 3.3 Нека матрицата A има вида

$$A = \begin{pmatrix} 1 & 1 & & & & & & & & \\ 1 & 0 & 1 & & & & & & & \\ & 1 & 0 & 1 & & & & & & \\ & & 1 & 1 & 1 & & & & & \\ & & & 1 & 0 & 1 & & & & \\ & & & & 1 & 0 & 1 & & & \\ & & & & & 1 & 1 & 1 & & \\ & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & \ddots & \ddots & 1 \\ & & & & & & & & 1 & 1 \end{pmatrix}.$$

Тази матрица винаги е добре обусловена. Получените резултати при различни стойности на n и δ_0 са представени в Таблица 3.3.

И при трите примера оригиналният алгоритъм без стабилизация на практика прекъсва поради деление на нула. В същото врѐме при стабилизирания вариант на алгоритъма се получават значително по-добри резултати. При това от таблиците се вижда, че когато $\delta_0 = 10^{-9}$ правата грешка е минимална и са необходими само една-две стъпки на итерационно уточняване. За да получим друго потвърждение на тази хипотеза нека да направим и числени експерименти със случайни матрици.

Пример 3.4 Нека за различни стойности на δ_0 да генерираме по 1000 случайни матрици от ред 100, при което, за да направим оригиналния алгоритъм лошо обусловен да фиксираме $b_i = 10^{-13}$, където i е случайно цяло число от интервала $[1, 100]$. При това, когато $\delta_0 = 0$ получаваме оригиналния алгоритъм. В Таблица 3.4 са представени усредните (озн. с FEA) и максималните (озн. с FEM) стойности за правата грешка в решението на системата, както и усреднения брой на направените итерации, при

$n\backslash\delta_0$	δ_0 =1E–6	δ_0 =1E–7	δ_0 =1E–8	δ_0 =1E–9	δ_0 =1E–10
$n = 100$	5.11E–15	5.75E–13	8.88E–15	1.07E–14	1.33E–15
N_{it}	2	1	1	1	2
$n = 200$	1.86E–14	1.90E–12	1.89E–14	1.28E–14	1.53E–14
N_{it}	2	1	1	1	1
$n = 500$	1.19E–13	5.33E–15	4.84E–14	4.42E–14	4.12E–14
N_{it}	2	2	1	1	1
$n = 1000$	4.25E–13	6.88E–15	9.28E–14	1.01E–13	3.47E–14
N_{it}	2	2	1	1	1

Таблица 3.1: Права грешка и брой на итерациите в Пример 3.1 за различни n и δ_0.

$n\backslash\delta_0$	δ_0 =1E–6	δ_0 =1E–7	δ_0 =1E–8	δ_0 =1E–9	δ_0 =1E–10
$n = 101$	1.06E–11	1.05E–10	5.33E–15	5.55E–15	8.99E–15
N_{it}	1	1	1	1	1
$n = 201$	4.10E–11	4.14E–13	2.49E–14	1.22E–14	3.70E–14
N_{it}	1	1	1	1	1
$n = 501$	2.53E–10	2.52E–12	3.06E–14	4.04E–14	1.27E–13
N_{it}	1	1	1	1	1
$n = 1001$	1.02E–9	1.01E–11	2.78E–13	1.35E–13	1.09E–13
N_{it}	1	1	1	1	1

Таблица 3.2: Права грешка и брой на итерациите в Пример 3.2 за различни n и δ_0.

$n\backslash\delta_0$	δ_0 =1E–6	δ_0 =1E–7	δ_0 =1E–8	δ_0 =1E–9	δ_0 =1E–10
$n = 100$	2.16E–11	2.11E–10	2.13E–12	4.99E–15	2.22E–14
N_{it}	3	2	2	2	2
$n = 200$	6.86E–10	1.68E–9	1.67E–11	1.95E–13	1.05E–13
N_{it}	3	2	2	2	2
$n = 500$	1.84E–9	2.62E–8	2.61E–10	2.62E–12	9.07E–14
N_{it}	4	2	2	2	3
$n = 1000$	3.67E–8	2.15E–9	2.09E–9	2.09E–11	6.79E–13
N_{it}	5	2	2	2	3

Таблица 3.3: Права грешка и брой на итерациите в Пример 3.3 за различни n и δ_0.

		$\delta_0 = 0$	δ_0 =1E–8	N_{it}	δ_0 =1E–9	N_{it}	δ_0 =1E–10	N_{it}
Ориг.	FEA	0.02						
	FEM	4.77						
Стаб.	FEA		1.24E–12	1.00	2.32E–13	1.02	4.23E–13	1.21
	FEM		1.01E–9	2	2.51E–11	2	9.75E–12	3

Таблица 3.4: Права грешка и брой на итерациите в Пример 3.4 за различни δ_0.

	$d1$	$d2$	$d3$
FE	5.68E–17	1.14E–16	1.11E–16
N_{it}	2	2	2

Таблица 3.5: Права грешка и брой на итерациите в Пример 3.5 при избрани $n = 1000$ и $\delta_0 = 1E$–9.

различни δ_0 за двата варианта на алгоритъма. От получените резултати и при тези експерименти можем да направим извода, че стабилизираният вариант на алгоритъма работи значително по-добре от оригиналния, както и че в качеството на оптимална стойност за δ_0 можем да препоръчаме $\delta_0 = 10^{-9}$.

Пример 3.5 Нека да разгледаме и един пример за прилагане на стабилизирания вариант на алгоритъма за решаване на система линейни уравнения с различни (в примера с три) десни части, всяка от които получаваме в зависимост от избраното точно решение. Нека в качеството на точно решение да изберем последователно

$$x1 = (1, 2, \ldots, n)^T,\ x2 = (n, n-1, \ldots, 1)^T,\ x3 = (1, 1, 1, 1, 1, 0, 0, \ldots, 0)^T,$$

и съответните десни част да означим с $d1, d2, d3$. И нека отново изходната матрица A има вида от Пример 3.1. Получените резултати за правата грешка в решението на системата и броя на итерациите необходими за итерационното му уточняване, когато $n = 1000$ и $\delta_0 = 10^{-9}$ са представени в Таблица 3.5.

Глава 4

Изследване на числената устойчивост на метода на цикличната редукция за решаване на блочно тридиагонални системи линейни уравнения

Основните резултати, представени в тази глава, са публикувани в статията:
Yalamov, P., V. Pavlov. Stability of the Block Cyclic Reduction. *Linear Algebra and Its Applications*, 249 (1996), 341–358.

В тази глава е разгледано обобщение на метода на цикличната редукция [38] (модификация без обратен ход) за решаване на блочно тридиагонални системи линейни уравнения и е представен анализ на числената устойчивост на алгоритъма в този случай. За удобство в изложението ще наричаме изследвания метод още и метод на блочно-цикличната редукция (БЦР). При паралелна реализация този метод притежава определени предимства, изразяващи се най-вече в равномерно натоварване на отделните процесори, намалено време за комуникациии (имайки предвид блочния вид на алгоритъма), малък брой паралелни стъпки (поради липсата на обратна субституция).

Що се отнася до изследвания на метода на цикличната редукция за решаване на блочно тридиагонални системи линейни уравнения, то такива има представени единствено в [33], където оригиналният алгоритъм на цикличната редукция се третира като итерационен метод, при което е изследвана неговата сходимост. В настоящата глава е направено пълно изследване на разпространението на грешките от закръгляване от нача-

лото до края на изчислителния процес при реализацията на разглеждания алгоритъм на БЦР. При това нека да отбележим, че направените изследвания запазват своята валидност и за произволна лентова система (всяка лентова система може да бъде структурирана в блочно тридиагонален вид).

Нека да напомним, че описание на модифицирания алгоритъм на цикличната редукция в тридиагоналния случай беше направено в предходната глава, а също и че изследване на неговата числена устойчивост в този случай е представено в [69], където е използван подход на обратен анализ. За съжаление обаче такъв подход в блочния случай не дава резултат. Това налага изследването да се направи по друг начин, по-точно използван е прав анализ, като в крайна сметка е показано, че оценката за грешката в решението зависи от обусловеността на диагоналните блокове в изходната блочно тридиагонална матрица.

Настоящата глава се състои от четири раздела. В Раздел 4.1 е направено описание на изследвания модифициран алгоритъм на цикличната редукция обобщен в блочен вид. При това аналогично на тридиагоналния случай (Глава 3), алгоритъмът се състои от $\lceil \log_2 n \rceil$ стъпки, където n е блочната размерност на системата. След извършване на тези стъпки се стига до система, чиято матрица е блочно диагонална, откъдето директно се получава решението на системата. В следващия раздел, при предположение за обратимост на блоковете по главния диагонал на матрицата на изходната система и равномерно блочно диагонално преобладаване по стълбове, е показано, че това свойство се запазва на всяка стъпка от алгоритъма, а също и са разгледани още някои негови свойства. В Раздел 4.3 е изследвана числената устойчивост на вече описания алгоритъм на БЦР, при това съществено се използват свойствата на алгоритъма, доказани в предишния раздел, т.е. изследванията са направени при предположение, че матрицата на решаваната система притежава свойството на равномерно блочно диагонално преобладаване по стълбове. Изведени са оценки за правата грешка в решението на системата в случай на строго и нестрого блочно диагонално преобладаване по стълбове. Получените оценки дават линейно приближение за грешката, т.е. при тяхното извеждане са пренебрегвани членовете, съдържащи машинната точност във втори и по-висок порядък. В противен случай се получават сложни и дълги изрази, работата, с които значително би усложнила извеждането на оценките и би влошила тяхната обозримост. На базата на получените оценки е направен изводът, че в случай на строго преобладаване грешката е от порядък n^2 по-малка от тази в другия случай. В края на главата (Раздел 4.4) са представени числени експерименти, които потвърждават, че изведените теоретични оценки за правата грешка в решението на системата са почти

достижими.

4.1 Описание на алгоритъма

Нека да разгледаме следната блочно тридиагонална система линейни уравнения:

$$AX = D, \tag{4.1}$$

където

$$A = \begin{pmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & C_{n-1} \\ & & & A_n & B_n \end{pmatrix}, \quad D = \begin{pmatrix} D_1 \\ \vdots \\ D_n \end{pmatrix}.$$

Блоковете $A_i, B_i, C_i \in \mathcal{R}^{N\times N}$, и в общия случай са плътни, $D_i \in \mathcal{R}^N$ са вектори, а n блочната размерност системата (4.1).

Понататък за простота в означенията ще предполагаме, че когато $i \leq 0$ или $i > n$ е изпълнено

$$A_i = C_i = D_i = 0, \; B_i = I,$$

където $I \in \mathcal{R}^{N\times N}$ е единична матрица, а също и че $A_1 = C_n = 0$.

Нека да означим $m = \lceil \log_2 n \rceil$. Както вече отбелязахме, алгоритъмът се състои от $m+1$ стъпки, при което за m стъпки чрез подходящи умножения от ляво матрицата A се преобразува в блочно диагонален вид и на $(m+1)$-вата стъпка се намира решението на системата. В аналитичен вид тези стъпки бихме могли да опишем по следния начин. Последователно за $k = 1, 2, \ldots, m$ се извършват следните действия:

1. Изчисляване на блочните елементи

$$P_i^{(k)} = -A_i^{(k-1)}[B_{i-2^{k-1}}^{(k-1)}]^{-1}, \tag{4.2}$$

$$Q_i^{(k)} = -C_i^{(k-1)}[B_{i+2^{k-1}}^{(k-1)}]^{-1}, \tag{4.3}$$

$$i = 1, 2, \ldots, n,$$

на матрицата

$$L^{(k)} = \begin{pmatrix} I & 0 & \cdots & 0 & Q_1^{(k)} & & & & \\ 0 & \ddots & & & & \ddots & & & \\ \vdots & & \ddots & & & & \ddots & & \\ 0 & & & \ddots & & & & \ddots & \\ P_{2^{k-1}+1}^{(k)} & & & & \ddots & & & & Q_{n-2^{k-1}}^{(k)} \\ & \ddots & & & & \ddots & & & 0 \\ & & \ddots & & & & \ddots & & \vdots \\ & & & \ddots & & & & \ddots & 0 \\ & & & & P_n^{(k)} & 0 & \cdots & 0 & I \end{pmatrix},$$

с която се умножава от ляво получените на предишната стъпка матрица $A^{(k-1)}$ и дясна част $D^{(k-1)}$ (за $k = 1$, $A^{(0)} = A$, $D^{(0)} = D$). Съществено е да отбележим, че обратните матрици в (4.2) и (4.3) не е необходимо да бъдат изчислявани при компютърна реализация, тъй като блоковете $P_i^{(k)}$ и $Q_i^{(k)}$ могат да се получат като решение на системи линейни уравнения с много десни части.

2. С помощта на така конструираната матрица $L^{(k)}$ получаваме

$$A^{(k)} = L^{(k)} A^{(k-1)}, \; D^{(k)} = L^{(k)} D^{(k-1)}, \tag{4.4}$$

където блочните елементи на $A^{(k)}$ и $D^{(k)}$ се намират по следните формули:

$$\begin{aligned} A_i^{(k)} &= P_i^{(k)} A_{i-2^{k-1}}^{(k-1)}, & (4.5)\\ C_i^{(k)} &= Q_i^{(k)} C_{i+2^{k-1}}^{(k-1)}, & (4.6)\\ B_i^{(k)} &= B_i^{(k-1)} + P_i^{(k)} C_{i-2^{k-1}}^{(k-1)} + Q_i^{(k)} A_{i+2^{k-1}}^{(k-1)}, & (4.7)\\ D_i^{(k)} &= D_i^{(k-1)} + P_i^{(k)} D_{i-2^{k-1}}^{(k-1)} + Q_i^{(k)} D_{i+2^{k-1}}^{(k-1)}, & (4.8)\\ i &= 1, 2, \ldots, n. \end{aligned}$$

В резултат ненулевите диагонали под и над главния диагонал на новата матрица $A^{(k)}$ са се преместили в посока към долния ляв и горния десен ъгъл. При това броят на нулевите диагонали между тях и главния диагонал е $2^k - 1$. По този начин точно след m стъпки се стига до блочно диагоналната матрица

$$A^{(m)} = \begin{pmatrix} B_1^{(m)} & & \\ & \ddots & \\ & & B_n^{(m)} \end{pmatrix}.$$

На $(m+1)$-вата стъпка се намира решението на системата (4.1)

$$X = L^{(m+1)} D^{(m)},$$

където

$$L^{(m+1)} = \begin{pmatrix} [B_1^{(m)}]^{-1} & & \\ & \ddots & \\ & & [B_n^{(m)}]^{-1} \end{pmatrix},$$

или по-точно

$$X_i = [B_i^{(m)}]^{-1} D_i^{(m)}, \quad i = 1, 2, \ldots, n.$$

4.2 Основни свойства на алгоритъма

В този раздел изхождайки от свойствата на изследвания алгоритъм в скаларния случай (виж [69]) ще направим тяхно обобщение в блочния случай. Разглежданите свойства ще бъдат необходими понататък за извеждане на оценки за числената устойчивост на алгоритъма.

В нашите изследвания ще предполагаме обратимост на всички блокове B_i, при което матрицата A е с равномерно блочно диагонално пробладаване по стълбове, т.е. изпълнено е:

$$\|A_i[B_{i-1}]^{-1}\| \le p, \ \|C_i[B_{i+1}]^{-1}\| \le q, \ s = p + q \le 1, \ i = 1, 2, \ldots, n, \tag{4.9}$$

при което p и q не зависят от i. Тук и понататък, ако изрично не е означено, е взета такава норма, за която е изпълнено $\|I\| = 1$. Нека да означим $p^{(0)} = p$, $q^{(0)} = q$, $s^{(0)} = s$.

Ясно е, че целият алгоритъм зависи от обратимостта на блоковете $B_i^{(k)}$ на всяка негова стъпка. В следващата лема ще покажем, че при направеното предположение (4.9) тази обратимост е налице.

Лема 4.1 *Ако всички блокове B_i са обратими и (4.9) е изпълнено, тогава $B_i^{(k)}$ са обратими за всяко k и i, при което съществуват числа $p^{(k)}, q^{(k)}, s^{(k)} = p^{(k)} + q^{(k)}$, такива че*

$$\|A_i^{(k)}[B_{i-2^k}^{(k)}]^{-1}\| \le p^{(k)}, \ \|C_{i-2^k}^{(k)}[B_i^{(k)}]^{-1}\| \le q^{(k)}, \ i = 1, 2, \ldots, n, \tag{4.10}$$

и

$$s^{(k)} \le (s^{(k-1)})^2 \le s^{2^k}. \tag{4.11}$$

Доказателство. Доказателството на лемата ще направим индуктивно по k.

Нека $k = 0$. Тогава очевидно верността на (4.10) и (4.11) следва от условието на лемата.

Допускайки, че твърдението на лемата е изпълнено за всички стъпки на алгоритъма до $(k-1)$-вата включително, ще докажем, че съществува $[B_i^{(k)}]^{-1}$ и, че оценките (4.10) и (4.11) са валидни и на k-та стъпка.

Изхождайки от (4.7) за блочно диагоналните елементи имаме

$$B_i^{(k)} = (I + W)B_i^{(k-1)}, \tag{4.12}$$

където $W = P_i^{(k)} C_{i-2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1} + Q_i^{(k)} A_{i+2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1}$. Тогава ако $(I+W)^{-1}$ съществува, можем да изразим

$$[B_i^{(k)}]^{-1} = [B_i^{(k-1)}]^{-1}(I + W)^{-1}. \tag{4.13}$$

Следователно $[B_i^{(k)}]^{-1}$ съществува ако съществува $(I+W)^{-1}$, а това е вярно ако (виж [67]) $\|W\| < 1$. Наистина имайки предвид (4.2), (4.3) и индукционното предположение получаваме

$$\begin{aligned} \|W\| &= \|P_i^{(k)} C_{i-2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1} + Q_i^{(k)} A_{i+2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1}\| \\ &\leq \|P_i^{(k)}\| \, \|C_{i-2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1}\| + \|Q_i^{(k)}\| \, \|A_{i+2^{k-1}}^{(k-1)} [B_i^{(k-1)}]^{-1}\| \\ &\leq 2p^{(k-1)} q^{(k-1)} \\ &\leq 0.5(p^{(k-1)} + q^{(k-1)})^2 \leq 0.5. \end{aligned} \tag{4.14}$$

Следователно можем да направим извода, че $(I+W)^{-1}$ съществува, както и че $B_i^{(k)}$ са обратими за всяко i и k.

Нека сега да докажем първото неравенство в (4.10) на k-та стъпка, т.е. ще докажем, че

$$\|A_i^{(k)} [B_{i-2^k}^{(k)}]^{-1}\| \leq p^{(k)} = \frac{(p^{(k-1)})^2}{1 - 2p^{(k-1)} q^{(k-1)}}. \tag{4.15}$$

Изхождайки от факта, че $\|W\| < 1$, следва валидността на известното неравенство (виж [67])

$$\|(I + W)^{-1}\| \leq \frac{1}{1 - \|W\|}. \tag{4.16}$$

Сега имайки предвид индукционното допускане и от (4.5), (4.13), (4.14), (4.16) получаваме

$$\|A_i^{(k)} [B_{i-2^k}^{(k)}]^{-1}\| = \|P_i^{(k)} A_{i-2^{k-1}}^{(k-1)} [B_{i-2^k}^{(k-1)}]^{-1} (I + W)^{-1}\|$$

$$\begin{aligned} &\leq \|P_i^{(k)}\| \, \|A_{i-2^{k-1}}^{(k-1)} [B_{i-2^k}^{(k-1)}]^{-1}\| \, \|(I+W)^{-1}\| \\ &\leq \frac{(p^{(k-1)})^2}{1-\|W\|} \leq \frac{(p^{(k-1)})^2}{1-2p^{(k-1)}q^{(k-1)}} = p^{(k)}. \end{aligned}$$

По същия начин се доказва и че

$$\|C_{i-2^k}^{(k)} [B_i^{(k)}]^{-1}\| \leq \frac{(q^{(k-1)})^2}{1-2p^{(k-1)}q^{(k-1)}} = q^{(k)}.$$

Логичен е обаче въпросът, дали знаменателят на $p^{(k)}$ може да стане равен на нула? Нека да проверим това, като изходим от простото неравенство

$$pq \leq 0.25(p+q)^2, \tag{4.17}$$

и използваме индукционното допускане $p^{(k-1)} + q^{(k-1)} = s^{(k-1)} \leq 1$. Тогава получаваме

$$1 - 2p^{(k-1)}q^{(k-1)} \geq 1 - 2(p^{(k-1)} + q^{(k-1)})^2 = 1 - 0.5(s^{(k-1)})^2 \geq 0.5.$$

Така показахме, че знаменателя на $p^{(k)}$ наистина не може да бъде нула. Сега лесно можем да докажем и (4.11)

$$\begin{aligned} s^{(k)} &= p^{(k)} + q^{(k)} = \frac{(p^{(k-1)})^2 + (q^{(k-1)})^2}{1-2p^{(k-1)}q^{(k-1)}} \\ &\leq (p^{(k-1)} + q^{(k-1)})^2 = (s^{(k-1)})^2 \leq \left(s^{2^{k-1}}\right)^2 = s^{2^k}. \end{aligned}$$

Следователно твърдението на лемата е изпълнено за всяко i и k. $\Diamond$

Забележка. Разбира се, оценката (4.14) е валидна в този вид, при условие, че работим в точна аритметика. Ето защо ще предполагаме, че $\|\delta W\| < 0.5$, където $\hat{W} = W + \delta W$ и δW е грешката в W (това на практика не е силно ограничение). Следователно $\|\hat{W}\| < 1$ и матрицата $\hat{W}$ е също обратима.

Нека да въведем следната норма:

$$\begin{aligned} \|G\|_{B\infty} &= \max_{1 \leq i \leq n} \sum_{j=1}^{n} \|G_{ij}\|, \\ \|g\|_{B\infty} &= \max_{1 \leq j \leq n} \|g_j\|, \end{aligned}$$

където G е една произволна блочна матрица с блочна размерност $n \times n$, и g е произволен блочен вектор (всеки елемент, на който е също вектор) с блочна размерност n.

Лема 4.2 *За матриците $L^{(k)}$ относно така въведената $B\infty$-норма е изпълнено*

$$\|L^{(k)}\|_{B\infty} \leq 1 + s^{2^{k-1}}.$$

Доказателство. Изхождайки от вида на матрицата $L^{(k)}$, формулите за дефиниране на нейните елементи (4.2), (4.3) и Лема 4.1 получаваме:

$$\|L^{(k)}\|_{B\infty} \leq 1 + p^{(k-1)} + q^{(k-1)} = 1 + s^{(k-1)} \leq 1 + s^{2^{k-1}}. \quad \Diamond$$

Лема 4.3 *На всяка стъпка k за блоковете $B_i^{(k)}$, $i = 1, 2, \ldots, n$, са валидни оценките*

$$\|B_i^{(k)}\| \leq (1 + 0.5s^{2^k})\|B_i^{(k-1)}\|, \tag{4.18}$$

$$\|[B_i^{(k)}]^{-1}\| \leq \frac{1}{1 - 0.5s^{2^k}}\|[B_i^{(k-1)}]^{-1}\|. \tag{4.19}$$

Доказателство. Нека най-напред да докажем (4.18). За целта изхождайки от (4.12) и (4.14) получаваме

$$\begin{aligned}\|B_i^{(k)}\| &= \|(I + W)\|\,\|B_i^{(k-1)}\| \\ &\leq (1 + 2p^{(k-1)}q^{(k-1)})\,\|B_i^{(k-1)}\| \\ &\leq (1 + 0.5(s^{(k-1)})^2)\,\|B_i^{(k-1)}\| \leq (1 + 0.5s^{2^k})\,\|B_i^{(k-1)}\|,\end{aligned}$$

което е точно (4.18).

За да докажем (4.19) нека да изходим от (4.12), откъдето

$$\|[B_i^{(k)}]^{-1}\| \leq \|(I + W)^{-1}\|\|[B_i^{(k-1)}]^{-1}\|. \tag{4.20}$$

Търсената оценка отгоре на (4.20) намираме използвайки (4.11), (4.14) и (4.16)

$$\begin{aligned}\|[B_i^{(k)}]^{-1}\| &\leq \frac{1}{1 - \|W\|}\|[B_i^{(k-1)}]^{-1}\| \\ &\leq \frac{1}{1 - 2p^{(k-1)}q^{(k-1)}}\|[B_i^{(k-1)}]^{-1}\| \\ &\leq \frac{1}{1 - 0.5(s^{(k-1)})^2}\|[B_i^{(k-1)}]^{-1}\| \\ &\leq \frac{1}{1 - 0.5s^{2^k}}\|[B_i^{(k-1)}]^{-1}\|,\end{aligned}$$

което е точно (4.19). $\Diamond$

4.3 Анализ на грешките от закръгляване

В този раздел, използвайки прав анализ ще проследим разпространението на грешките от закръгляване от началото до края на изчислителния процес при компютърната реализация на изследвания алгоритъм. Използването на такъв подход се налага и поради факта, че обратният анализ не дава добър резултат. При това за по-голяма яснота, навсякъде ще пренебрегваме членовете съдържащи машинната точност във втори и по-висок порядък. В противен случай се получават сложни и дълги изрази, работата с които значително би усложнила извеждането на оценките и би влошила тяхната обозримост.

В резултат ще получим линейно приближение за правата грешка в решението на системата (4.1). Поради това в нашия анализ можем да използваме точните стойности на елементите $P_i^{(k)}$, $Q_i^{(k)}$, $A_i^{(k)}$, $B_i^{(k)}$, $C_i^{(k)}$, $D_i^{(k)}$ вместо изчислените, защото в изведените оценки те се умножават с малки локални грешки, като по този начин грешките в тези елементи се отразяват върху членовете съдържащи машинната точност във втори и по-висок ред, които както отбелязахме ще пренебрегваме.

Нека в нашия анализ да изходим от матричните умножения (4.4), които се извършват на всяка стъпка от алгоритъма. Да изследваме найнапред грешката в матрицата $A^{(k)}$. На практика за нейното пресмятане се използват не точните $L^{(k)}$ и $A^{(k-1)}$, а изчислените с грешки от закръгляване $\hat{L}^{(k)}$ и $\hat{A}^{(k-1)}$. Освен това резултатът от умножението на двете матрици също се закръглява. В такъв случай получаваме

$$\hat{A}^{(k)} = \hat{L}^{(k)}\hat{A}^{(k-1)} + \eta^{(k)}, \tag{4.21}$$

където $\eta^{(k)}$ е матрица от локални грешки получени от матричното умножение $\hat{L}^{(k)}\hat{A}^{(k-1)}$.

Нека да припомним, че елементите $P_i^{(k)}$ и $Q_i^{(k)}$ на матрицата $L^{(k)}$ пресметнати по (4.2) и (4.3) всъщност са решения на матричните уравнения

$$P_i^{(k)} B_{i-2^{k-1}}^{(k-1)} + A_i^{(k-1)} = 0, \tag{4.22}$$

$$Q_i^{(k)} B_{i+2^{k-1}}^{(k-1)} + C_i^{(k-1)} = 0, \tag{4.23}$$

т.е. $P_i^{(k)}$ и $Q_i^{(k)}$ се намират по такъв начин, че след умножението (4.4), на мястото на старите ненулеви диагонали под и над главния диагонал на матрицата $A^{(k-1)}$ да се получат нули. В точна аритметика това наистина би било така, но на практика вече изчислените $\hat{P}_i^{(k)}$ и $\hat{Q}_i^{(k)}$ ще доведат до грешки при компютърното пресмятане на блочните елементи $A_{i,i-2^{k-1}}^{(k)}$,

$A^{(k)}_{i,i+2^{k-1}}$ на матрицата $A^{(k)}$, или по-общо ще имаме $\eta^{(k)}_{i,i-2^{k-1}}, \eta^{(k)}_{i,i+2^{k-1}} \neq 0$. Освен това новите блочни елементи $A^{(k)}_i$, $B^{(k)}_i$, $C^{(k)}_i$ на матрицата $A^{(k)}$ са също изчислени с грешки. По този начин във всеки ред на матрицата $\eta^{(k)}$ (с изключение на първите и последните няколко реда) има 5 ненулеви блочни елемента.

Прилагайки (4.21) рекурсивно стигаме до

$$\hat{A}^{(k)} = \hat{L}^{(k)} \dots \hat{L}^{(1)} A + \hat{L}^{(k)} \dots \hat{L}^{(2)} \eta^{(1)} + \dots + \hat{L}^{(k)} \eta^{(k-1)} + \eta^{(k)},$$

тогава за $k = m + 1$ ще имаме

$$\begin{aligned} \hat{A}^{(m+1)} &= \hat{L}^{(m+1)} \dots \hat{L}^{(1)} A + \hat{L}^{(m+1)} \dots \hat{L}^{(2)} \eta^{(1)} + \dots \\ &\quad + \hat{L}^{(m+1)} \hat{L}^{(m)} \eta^{(m-1)} + \hat{L}^{(m+1)} \eta^{(m)}, \end{aligned}$$

или

$$\hat{A}^{(m+1)} = \hat{I} + \delta A^{(m+1)}, \tag{4.24}$$

където

$$\hat{I} = \hat{L}^{(m+1)} \dots \hat{L}^{(1)} A, \tag{4.25}$$

$$\delta A^{(m+1)} = \hat{L}^{(m+1)} \dots \hat{L}^{(2)} \eta^{(1)} + \dots + \hat{L}^{(m+1)} \hat{L}^{(m)} \eta^{(m-1)} + \hat{L}^{(m+1)} \eta^{(m)}. \tag{4.26}$$

Аналогично за дясната част получаваме

$$\begin{aligned} \hat{D}^{(m+1)} &= \hat{L}^{(m+1)} \dots \hat{L}^{(1)} D + \hat{L}^{(m+1)} \dots \hat{L}^{(2)} \sigma^{(1)} + \dots \\ &\quad + \hat{L}^{(m+1)} \hat{L}^{(m)} \sigma^{(m-1)} + \hat{L}^{(m+1)} \sigma^{(m)} + \sigma^{(m+1)}, \end{aligned}$$

или

$$\hat{X} = \bar{X} + \delta D^{(m+1)}, \tag{4.27}$$

където сме означили

$$\begin{aligned} \hat{X} &= \hat{D}^{(m+1)}, \\ \bar{X} &= \hat{L}^{(m+1)} \dots \hat{L}^{(1)} D, \\ \delta D^{(m+1)} &= \hat{L}^{(m+1)} \dots \hat{L}^{(2)} \sigma^{(1)} + \dots + \hat{L}^{(m+1)} \hat{L}^{(m)} \sigma^{(m-1)} \\ &\quad + \hat{L}^{(m+1)} \sigma^{(m)} + \sigma^{(m+1)}, \end{aligned} \tag{4.28}$$

а $\sigma^{(k)}, k = 1, 2, \dots, m + 1$, са вектори от локални грешки получени при изчисляването на $D^{(k)}$.

Изразът (4.25) за $\hat{I}$ можем да запишем и така

$$\hat{I} = \left(L^{(m+1)} \dots L^{(1)} + \delta L\right) A = (L + \delta L) A = I + \delta L A, \tag{4.29}$$

където δL е общата грешка от изчислението на матричното произведение $\hat{L}^{(m+1)}\hat{L}^{(m)}\dots\hat{L}^{(1)}$, т.е. при въведените означения е изпълнено

$$\hat{L}^{(m+1)}\hat{L}^{(m)}\dots\hat{L}^{(1)} = L^{(m+1)}L^{(m)}\dots L^{(1)} + \delta L = L + \delta L. \tag{4.30}$$

Тогава замествайки (4.29) в (4.24) получаваме

$$\hat{A}^{(m+1)} = I + \delta LA + \delta A^{(m+1)},$$

но в крайна сметка на $(m+1)$-вата стъпка трябва $\hat{A}^{(m+1)} = I$, откъдето

$$\delta L = -\delta A^{(m+1)}A^{-1}. \tag{4.31}$$

Тогава от (4.27), (4.30) и (4.31) намираме

$$\begin{aligned}\hat{X} &= (L+\delta L)D + \delta D^{(m+1)} = X + \delta LD + \delta D^{(m+1)}\\ &= X - \delta A^{(m+1)}X + \delta D^{(m+1)},\end{aligned}$$

следователно

$$\hat{X} - X = -\delta A^{(m+1)}X + \delta D^{(m+1)}.$$

Сега за правата грешка $\delta X = \hat{X} - X$, използвайки въведената блочна норма, получаваме

$$\|\delta X\|_{B\infty} \le \|\delta A^{(m+1)}\|_{B\infty}\|X\|_{B\infty} + \|\delta D^{(m+1)}\|_{B\infty}, \tag{4.32}$$

и за да намерим оценка за нея е необходимо да оценим дясната част на (4.32). За тази цел ще ни бъде необходима оценка за $\|L^{(m+1)}\|_{B\infty}$. Изхождайки от вида на $L^{(m+1)}$ от направената дефиниция за блочна норма имаме

$$\|L^{(m+1)}\|_{B\infty} = \max_{1\le i\le n}\|[B_i^{(m)}]^{-1}\|. \tag{4.33}$$

Нека да оценим рекурсивно $[B_i^{(m)}]^{-1}$ от (4.19), да го заместим в (4.33) и да използваме очевидното неравенство

$$\frac{1}{1-0.5s^{2^j}} \le 1 + s^{2^j}.$$

Тогава получаваме

$$\begin{aligned}\|L^{(m+1)}\|_{B\infty} &\le \max_{1\le i\le n}\prod_{j=1}^{m}\frac{1}{1-0.5s^{2^j}}\|[B_i]^{-1}\|\\ &\le b\prod_{j=1}^{m}(1+s^{2^j}),\end{aligned} \tag{4.34}$$

където $b = \max_i \|B_i^{-1}\|$. Нека сега да оценим $\|\delta A^{(m+1)}\|_{B\infty}$ изхождайки от (4.26) и използвайки (4.34), при което щом пренебрегваме членовете съдържащи ρ_0 във втори и по-висок порядък можем да използваме $L^{(k)}, k = 1, \ldots, m+1$, вместо $\hat{L}^{(k)}$. Тогава получаваме

$$\begin{aligned}
\|\delta A^{(m+1)}\|_{B\infty} &\leq \|L^{(m+1)}\|_{B\infty} \sum_{k=1}^{m} \prod_{j=k+1}^{m} \|L^{(j)}\|_{B\infty} \|\eta^{(k)}\|_{B\infty} \\
&\leq b \prod_{j=1}^{m} (1 + s^{2^j}) \sum_{k=1}^{m} \prod_{j=k}^{m-1} (1 + p^{(j)} + q^{(j)}) \|\eta^{(k)}\|_{B\infty} \\
&\leq b \prod_{j=1}^{m} (1 + s^{2^j}) \sum_{k=1}^{m} \prod_{j=k}^{m-1} (1 + s^{2^j}) \|\eta^{(k)}\|_{B\infty}. \qquad (4.35)
\end{aligned}$$

Забележка. Тук и понататък ще считаме, че ако $t_1 > t_2$, то

$$\prod_{t=t_1}^{t_2} u_t = 1.$$

По подобен начин изхождайки от (4.28) получаваме и оценка за $\|\delta D^{(m+1)}\|_{B\infty}$:

$$\|\delta D^{(m+1)}\|_{B\infty} \leq b \prod_{j=1}^{m} (1 + s^{2^j}) \sum_{k=1}^{m} \prod_{j=k}^{m-1} (1 + s^{2^j}) \|\sigma^{(k)}\|_{B\infty} + \|\sigma^{(m+1)}\|_{B\infty}. \qquad (4.36)$$

Нека сега да оценим локалните грешки $\eta^{(k)}$ и $\sigma^{(k)}$. Както е известно (виж [36]) в резултат на закръгляването при компютърно пресмятане на сума и произведение на две матрици $X, Y \in \mathcal{R}^{N\times N}$ се появяват грешки от закръгляване, при което са валидни следните равенства:

$$\begin{aligned}
\mathrm{fl}(X + Y) &= X + Y + \eta_{x+y}, \\
\mathrm{fl}(XY) &= XY + \eta_{xy},
\end{aligned}$$

където за матриците от грешки са изпълнени оценките

$$\begin{aligned}
\|\eta_{x+y}\| &\leq \|X + Y\| \rho_0, \qquad (4.37) \\
\|\eta_{xy}\| &\leq N \|X\| \, \|Y\| \rho_0. \qquad (4.38)
\end{aligned}$$

В конкретния случай за грешките от закръгляване в (4.5) и (4.6), използвайки горните оценки и Лема 4.1 получаваме

$$\|\eta_{i+2^k,i}^{(k)}\| \leq N \|P_{i+2^{k-1}}^{(k)}\| \|A_i^{(k-1)}\| \rho_0$$

$$\begin{aligned} &= N\|P^{(k)}_{i+2^{k-1}}\|\,\|A^{(k-1)}_i[B^{(k-1)}_i]^{-1}B^{(k-1)}_i\|\rho_0 \\ &\le N(p^{(k-1)})^2\|B^{(k-1)}_i\|\rho_0 \\ &\le N(p^{(k-1)})^2\bar{B}_{k-1}\rho_0,\ \bar{B}_{k-1}=\max_i\|B^{(k-1)}_i\|, \end{aligned} \tag{4.39}$$

аналогично

$$\begin{aligned} \|\eta^{(k)}_{i-2^k,i}\| &\le N\|Q^{(k)}_{i-2^{k-1}}\|\|C^{(k-1)}_i\|\rho_0 \\ &\le N(q^{(k-1)})^2\bar{B}_{k-1}\rho_0. \end{aligned} \tag{4.40}$$

Нека сега да оценим и грешката в (4.7) $\eta^{(k)}_{ii}$, при което ще предполагаме, че сумиранията са направени в следния ред:

$$B^{(k)}_i = B^{(k-1)}_i + \left(P^{(k)}_i C^{(k-1)}_{i-2^{k-1}} + Q^{(k)}_i A^{(k-1)}_{i+2^{k-1}}\right). \tag{4.41}$$

Тогава тази грешка ще бъде резултат от сумата на грешките за всяка матрична операция в (4.41), където има две умножения и две събирания. Пренебрегвайки отново членовете съдържащи ρ_0 във втори и по-висок порядък получаваме

$$\eta^{(k)}_{ii} = \mu^{(ki)}_1 + \mu^{(ki)}_2 + \mu^{(ki)}_3 + \mu^{(ki)}_4, \tag{4.42}$$

където за отделните грешки използвайки (4.37), (4.38) и Лема 4.1 са валидни оценките

$$\begin{aligned} \|\mu^{(ki)}_1\| &\le N\|P^{(k)}_i\|\|C^{(k-1)}_{i-2^{k-1}}\|\rho_0 \le Np^{(k-1)}q^{(k-1)}\bar{B}_{k-1}\rho_0, \\ \|\mu^{(ki)}_2\| &\le N\|Q^{(k)}_i\|\|A^{(k-1)}_{i+2^{k-1}}\|\rho_0 \le Np^{(k-1)}q^{(k-1)}\bar{B}_{k-1}\rho_0, \\ \|\mu^{(ki)}_3\| &\le \left(\|P^{(k)}_i C^{(k-1)}_{i-2^{k-1}}\| + \|Q^{(k)}_i A^{(k-1)}_{i+2^{k-1}}\|\right)\rho_0 \le 2\bar{B}_{k-1}p^{(k-1)}q^{(k-1)}\rho_0, \\ \|\mu^{(ki)}_4\| &\le \left(\|B^{(k-1)}_i\| + \|P^{(k)}_i C^{(k-1)}_{i-2^{k-1}}\| + \|Q^{(k)}_i A^{(k-1)}_{i+2^{k-1}}\|\right)\rho_0 \\ &\le \bar{B}_{k-1}(1+2p^{(k-1)}q^{(k-1)})\rho_0. \end{aligned}$$

Тогава, като използваме тези оценки, (4.17), (4.42) и Лема 4.1 стигаме до

$$\begin{aligned} \|\eta^{(k)}_{ii}\| &\le \|\mu^{(ki)}_1\| + \|\mu^{(ki)}_2\| + \|\mu^{(ki)}_3\| + \|\mu^{(ki)}_4\| \\ &\le \bar{B}_{k-1}\left(2Np^{(k-1)}q^{(k-1)} + 2p^{(k-1)}q^{(k-1)} + 1 + 2p^{(k-1)}q^{(k-1)}\right)\rho_0 \\ &= \bar{B}_{k-1}\left((2N+4)p^{(k-1)}q^{(k-1)} + 1\right)\rho_0 \\ &= \bar{B}_{k-1}\left(4(\frac{N}{2}+1)p^{(k-1)}q^{(k-1)} + 1\right)\rho_0 \\ &\le \bar{B}_{k-1}\left(s^{2^k}(0.5N+1)+1\right)\rho_0 \le \bar{B}_{k-1}(0.5N+2)\rho_0. \end{aligned} \tag{4.43}$$

Нека да отбележим, че оценката няма да се промени съществено ако сумиранията са направени в друг ред.

Както вече отбелязахме елементите $P_i^{(k)}$ и $Q_i^{(k)}$ на матрицата $L^{(k)}$ могат да се намерят като решения на системи линейни уравнения с много десни части. Нека да предположим, че това е направено използвайки алгоритъм, който е обратно устойчив.

Решаването на произволна система линейни уравнения с много десни части всъщност е еквивалентно на решаване на произволно матрично уравнение от вида $AX = F$. Нека за решаване на това уравнение е използван обратно устойчив алгоритъм. Тогава ще бъде изпълнено (виж [67])

$$\begin{aligned} (A+\varepsilon_A)\hat{X} &= F+\varepsilon_F, && (4.44)\\ \|\varepsilon_A\| &\le C_N\|A\|\rho_0, && (4.45)\\ \|\varepsilon_F\| &\le C_N\|F\|\rho_0, && (4.46) \end{aligned}$$

където ε_A и ε_B са матрици от еквивалентни смущения, $\hat{X} = X + \delta X$ е изчисленото решение, а C_N е константа, зависеща линейно от N. Нека изхождайки от (4.44) да оценим по норма δX

$$\begin{aligned} \|\delta X\| &\le \|(A+\varepsilon_A)^{-1}\|\,\|\varepsilon_F-\varepsilon_A X\| \\ &\le \|\left[A(I+A^{-1}\varepsilon_A)\right]^{-1}\|\,(\|\varepsilon_F\|+\|\varepsilon_A\|\,\|X\|, \end{aligned}$$

и ако вземем предвид оценките (4.45) и (4.46) получаваме

$$\|\delta X\| \le \|(I+A^{-1}\varepsilon_A)^{-1}\|\,\|A^{-1}\|C_N(\|A\|\,\|X\|+\|F\|). \tag{4.47}$$

Освен това за резидуала $R = F - A\hat{X}$ изхождайки от (4.44), (4.45) и (4.46) получаваме $R = -\varepsilon_A\hat{X} + \varepsilon_F$ и

$$\|R\| \le C_N(\|A\|\,\|\hat{X}\|+\|F\|)\rho_0. \tag{4.48}$$

В конкретния случай, както отбелязахме $P_i^{(k)}$ и $Q_i^{(k)}$ са съответно решения на системите (4.22) и (4.23). При числено решаване на тези системи, обаче вместо $B_{i-2^{k-1}}^{(k-1)}, A_i^{(k-1)}, B_{i+2^{k-1}}^{(k-1)}, C_i^{(k-1)}$ на практика участват $\hat{B}_{i-2^{k-1}}^{(k-1)}, \hat{A}_i^{(k-1)}, \hat{B}_{i+2^{k-1}}^{(k-1)}, \hat{C}_i^{(k-1)}$, откъдето в резултат получаваме и приближени решения $\hat{P}_i^{(k)}$ и $\hat{Q}_i^{(k)}$. Следователно ако заместим тези решения съответно в (4.22) и (4.23) получаваме

$$\begin{aligned} \hat{P}_i^{(k)}\hat{B}_{i-2^{k-1}}^{(k-1)} + \hat{A}_i^{(k-1)} &= \eta_{i,i-2^{k-1}}^{(k)}, \\ \hat{Q}_i^{(k)}\hat{B}_{i+2^{k-1}}^{(k-1)} + \hat{C}_i^{(k-1)} &= \eta_{i,i+2^{k-1}}^{(k)}, \end{aligned}$$

където $\eta^{(k)}_{i,i-2^{k-1}}$ и $\eta^{(k)}_{i,i+2^{k-1}}$ са грешките, които допускаме и очевидно те съвпадат с резидуалите, а тях можем да оценим използвайки (4.48)

$$\begin{aligned}
\|\eta^{(k)}_{i,i-2^{k-1}}\| &\leq C_N \left(\|B^{(k-1)}_{i-2^{k-1}}\|\|P^{(k)}_i\| + \|A^{(k-1)}_i\|\right)\rho_0 \\
&\leq C_N \left(\|B^{(k-1)}_{i-2^{k-1}}\|\|P^{(k)}_i\| + \|A^{(k-1)}_i[B^{(k-1)}_{i-2^k}]^{-1}\|\,\|B^{(k-1)}_{i-2^k}\|\right)\rho_0 \\
&\leq C_N \left(\bar{B}_{k-1}p^{(k-1)} + p^{(k-1)}\bar{B}_{k-1}\right)\rho_0 \\
&\leq C_N p^{(k-1)}\bar{B}_{k-1}\rho_0, \qquad (4.49)
\end{aligned}$$

и аналогично

$$\|\eta^{(k)}_{i,i+2^{k-1}}\| \leq C_N \left(\|B^{(k-1)}_{i+2^{k-1}}\|\|Q^{(k)}_i\| + \|C^{(k-1)}_i\|\right)\rho_0 \leq C_N q^{(k-1)}\bar{B}_{k-1}\rho_0, \qquad (4.50)$$

Забележка. Тук и понататък умноженията на C_N по константи независещи от N, също ще означаваме с C_N.

Сега за матрицата от грешки $\eta^{(k)}$ от (4.39), (4.40), (4.43), (4.49), (4.50), Лема 4.1 и Лема 4.3 стигаме до

$$\begin{aligned}
\|\eta^{(k)}\|_{B\infty} &\leq \bar{B}_{k-1}\left(N(s^{(k-1)})^2 + 0.5N + 2 + C_N s^{(k-1)}\right)\rho_0 \\
&\leq C_N\bar{B}_{k-1}\rho_0 \\
&\leq C_N\bar{B}\prod_{j=1}^{k-1}(1+0.5s^{2^j})\rho_0, \qquad (4.51)
\end{aligned}$$

където $\bar{B} = \max_i \|B_i\|$. Вече можем да получим окончателната оценка за $\|\delta A^{(m+1)}\|_{B\infty}$, като заместим (4.51) в (4.34)

$$\|\delta A^{(m+1)}\|_{B\infty} \leq C_N\bar{B}b\prod_{j=1}^{m}(1+s^{2^j})\sum_{k=1}^{m}\prod_{j=1}^{k-1}(1+0.5s^{2^j})\prod_{j=k}^{m}(1+s^{2^j})\rho_0. \qquad (4.52)$$

За локалните грешки $\sigma^{(k)}$ при пресмятане на дясната част $\delta D^{(m+1)}$ по подобен начин считайки, че сумиранията в (4.8) са направени в следния ред:

$$D^{(k)}_i = D^{(k-1)}_i + \left(P^{(k)}_i D^{(k-1)}_{i-2^{k-1}} + Q^{(k)}_i D^{(k-1)}_{i+2^{k-1}}\right),$$

получаваме

$$\sigma^{(k)}_i = \mu^{(ki)}_5 + \mu^{(ki)}_6 + \mu^{(ki)}_7 + \mu^{(ki)}_8, \qquad (4.53)$$

където за отделните грешки използвайки (4.37), (4.38) и Лема 4.1 имаме

$$\begin{aligned}
\|\mu^{(ki)}_5\| &\leq N\|P^{(k)}_i\|\|D^{(k-1)}_{i-2^{k-1}}\|\rho_0 \leq Np^{(k-1)}\|D^{(k-1)}_{i-2^{k-1}}\|\rho_0, \\
\|\mu^{(ki)}_6\| &\leq N\|Q^{(k)}_i\|\|D^{(k-1)}_{i+2^{k-1}}\|\rho_0 \leq Nq^{(k-1)}\|D^{(k-1)}_{i+2^{k-1}}\|\rho_0,
\end{aligned}$$

$$\begin{aligned}
\|\mu_7^{(ki)}\| &\leq \left(\|P_i^{(k)}\|\|D_{i-2^{k-1}}^{(k-1)}\| + \|Q_i^{(k)}\|\,\|D_{i+2^{k-1}}^{(k-1)}\|\right)\rho_0 \\
&\leq \left(p^{(k-1)}\|D_{i-2^{k-1}}^{(k-1)}\| + q^{(k-1)}\|D_{i+2^{k-1}}^{(k-1)}\|\right)\rho_0, \\
\|\mu_8^{(ki)}\| &\leq \left(\|D_i^{(k-1)}\| + p^{(k-1)}\|D_{i-2^{k-1}}^{(k-1)}\| + q^{(k-1)}\|D_{i+2^{k-1}}^{(k-1)}\|\right)\rho_0.
\end{aligned}$$

Заместваики тези оценки в (4.53) и отчитайки, че $\|D_j^{(k-1)}\| \leq \|D^{(k-1)}\|_{B\infty}$ за всяко j, получаваме

$$\begin{aligned}
\|\sigma^{(k)}\|_{B\infty} &\leq \|D^{(k-1)}\|_{B\infty}(1 + p^{(k-1)} + p^{(k-1)} + q^{(k-1)} + q^{(k-1)} \\
&\quad + Np^{(k-1)} + Nq^{(k-1)})\rho_0 \\
&\leq \|D^{(k-1)}\|_{B\infty}(1 + 2s^{(k-1)} + Ns^{(k-1)})\rho_0 \\
&\leq \|D^{(k-1)}\|_{B\infty}(1 + 2s^{2^k} + Ns^{2^k})\rho_0. \qquad (4.54)
\end{aligned}$$

Като използваме Лема 4.2 можем да оценим $\|D^{(k-1)}\|_{B\infty}$, изхождайки от

$$D^{(k-1)} = L^{(k-1)} \ldots L^{(1)} D,$$

откъдето получаваме

$$\|D^{(k-1)}\|_{B\infty} \leq \prod_{j=0}^{k-2}(1 + s^{2^j})\|D\|_{B\infty}, \qquad (4.55)$$

и заместваики (4.55) в (4.54) стигаме до

$$\|\sigma^{(k)}\|_{B\infty} \leq (1 + (N+2)s^{2^k}) \prod_{j=0}^{k-2}(1 + s^{2^j})\|D\|_{B\infty}\rho_0. \qquad (4.56)$$

Нека да разгледаме системата

$$B_i^{(m)} X_i = D_i^{(m)}, \qquad (4.57)$$

и да предположим, че тя се решава използвайки алгоритъм, който е обратно устойчив. Тогава да оценим $\|\sigma_i^{(m+1)}\|$ е все едно да оценим $\|\delta X_i\|$, тъй като това е еквивалентно да оценим грешката в решението X_i, означена с $\sigma_i^{(m+1)}$. За целта ще намерим оценки отгоре, като използваме (4.47), при което ще предполагаме, че $\|[B_i^{(m)}]^{-1}\varepsilon_{B_i^{(m)}}\| < 1$. При тези предположения получаваме

$$\begin{aligned}
\|\sigma_i^{(m+1)}\| &= \|\delta X_i\| \\
&\leq \|\left(I + [B_i^{(m)}]^{-1}\varepsilon_{B_i^{(m)}}\right)^{-1}\|\,\|[B_i^{(m)}]^{-1}\|C_N\left(\|B_i^{(m)}\|\,\|X_i\| + \|D_i^{(m)}\|\right) \\
&\leq \frac{1}{1 - C_N\|[B_i^{(m)}]^{-1}\|\,\|B_i^{(m)}\|\rho_0}\|[B_i^{(m)}]^{-1}\|C_N\left(\|B_i^{(m)}\|\,\|X_i\| + \|D_i^{(m)}\|\right) \\
&= C_N\|[B_i^{(m)}]^{-1}\|\left(\|B_i^{(m)}\|\,\|X_i\| + \|D_i^{(m)}\|\right)\rho_0 \\
&= C_N\left(\|[B_i^{(m)}]^{-1}\|\,\|B_i^{(m)}\|\,\|X_i\| + \|[B_i^{(m)}]^{-1}\|\,\|D_i^{(m)}\|\right)\rho_0. \qquad (4.58)
\end{aligned}$$

Сега за вектора от грешки $\sigma^{(m+1)}$, като използваме Лема 4.3 и (4.58) стигаме до

$$\|\sigma^{(m+1)}\|_{B\infty} \leq C_N \left(\bar{B}b \prod_{j=1}^{m} (1+0.5s^{2^j})(1+s^{2^j}) \|X\|_{B\infty} + \prod_{j=0}^{m-1} (1+s^{2^j})^2 \|D\|_{B\infty} \right) \rho_0. \quad (4.59)$$

Окончателната оценка за $\|\delta D^{(m+1)}\|_{B\infty}$ получаваме, като заместим (4.56) и (4.59) в (4.36)

$$\|\delta D^{(m+1)}\|_{B\infty} \leq C_N \|D\|_{B\infty} b \prod_{j=1}^{m} (1+s^{2^j}) \sum_{k=1}^{m} \prod_{j=k}^{m-1} (1+s^{2^j}) \prod_{j=0}^{k-2} (1+s^{2^j}) \rho_0 + C_N \bar{B}b \prod_{j=1}^{m} (1+s^{2^j})(1+0.5s^{2^j}) \|X\|_{B\infty} \rho_0 + C_N b \prod_{j=1}^{m} (1+s^{2^j})^2 \|D\|_{B\infty} \rho_0. \quad (4.60)$$

Накрая ако заместим намерените оценки (4.52) и (4.60) в (4.32) получаваме и окончателната оценка за грешката в решението на системата

$$\|\delta X\|_{B\infty} \leq C_N b \left(\bar{B} \prod_{j=1}^{m} (1+s^{2^j}) \sum_{k=1}^{m+1} \prod_{j=1}^{k-1} (1+0.5s^{2^j}) \prod_{j=k}^{m} (1+s^{2^j}) \|X\|_{B\infty} + (m+1) \prod_{j=0}^{m} (1+s^{2^j})^2 \|D\|_{B\infty} \right) \rho_0. \quad (4.61)$$

Нека сега изхождайки от (4.61) да намерим оценки за грешката в решението на системата, но в относителен смисъл. За целта ще разгледаме два случая. Първо, нека предположим, че $s = 1$, т.е. изходната матрица е с нестрого блочно диагонално преобладаване по стълбове. Тогава от (4.61) получаваме

$$\frac{\|\delta X\|_{B\infty}}{\|X\|_{B\infty}} \leq \frac{C_N b n^2}{\|X\|_{B\infty}} \left(\bar{B} \|X\|_{B\infty} + (\log_2 n + 1) \|D\|_{B\infty} \right) \rho_0 \leq f(n,N) \kappa \rho_0, \quad (4.62)$$

където $f(n,N) = C_N n^2 \log_2 n$ и $\kappa = b(\bar{B}\|X\|_{B\infty} + \|D\|_{B\infty})/\|X\|_{B\infty}$ е число на обусловеност зависещо от входните данни.

Вторият случай е, когато $0 < s < 1$, т.е. изходната матрица е със строго блочно диагонално преобладаване по стълбове. В този случай е ясно, че s^{2^j} намалява много бързо, когато j нараства и тогава произведенията в (4.61) могат да бъдат ограничени от малки константи. По този

s	1/3	1/2	2/3	3/4	5/6	9/10
$g(s)$	2.25	4.00	9.01	16.00	36.00	100.00

Таблица 4.1: Стойности на функцията $g(s)$.

начин влиянието на последните членове в тези произведения е пренебрежимо. Следователно за $j \geq j_0$, където j_0 определяме от условието

$$2s^{2^{j_0}} \leq \rho_0, \tag{4.63}$$

можем да пренебрегнем съответните членове в произведенията на (4.61), т.е. пренебрегваме членовете съдържащи ρ_0 във втори и по-висок порядък. Изхождайки от (4.63) можем да определим точно j_0:

$$j_0 = \lceil \log_2 \log_s (\frac{1}{2}\rho_0) \rceil.$$

Така окончателната оценка в този втори случай добива вида

$$\frac{\|\delta X\|_{B\infty}}{\|X\|_{B\infty}} \leq \frac{C_N b g(s)}{\|X\|_{B\infty}} \left(\bar{B}\|X\|_{B\infty} + (\log_2 n + 1)\|D\|_{B\infty}\right) \rho_0 \leq h(n,N)\kappa\rho_0, \tag{4.64}$$

където $h(n,N) = C_N g(s) \log_2 n$, κ е вече дефинираното по-горе число на обусловеност, а

$$g(s) = \prod_{j=0}^{j_0} (1 + s^{2^j})^2.$$

е лесно изчислима функция. Някои стойности на тази функция за различни s са дадени в Таблица 4.3. И така, в случая когато $s < 1$ (и разбира се s не е близко до 1) получаваме, че относителната грешка в решението на системата има порядък $\mathcal{O}(C_N g(s)(\log_2 n)\kappa\rho_0)$, който намалява, когато s намалява. В другия случай, когато $s = 1$ или е близко до 1 можем да използваме оценката (4.61), откъдето се вижда, че относителната грешка в решението на системата има порядък $\mathcal{O}(C_N n^2 (\log_2 n)\kappa\rho_0)$. Накрая ако сравним порядъка на грешките в двата случая, то очевидно в случая на строго блочно диагонално преобладаване по стълбове грешката е от порядък n^2 по-малка от тази в другия случай.

4.4 Числени експерименти

Числените експерименти в този раздел са направени на Фортран, с двойна точност $\rho_0 \approx 2.22\text{E}{-16}$. При числената реализация ще измерваме правата

грешка в решението в относителен смисъл

$$FE = \frac{\|\hat{X} - X\|_\infty}{\|X\|_\infty}, \tag{4.65}$$

и стойностите при различни n на функцията $TERR(n)$, която дава финна горна граница за тази грешка. Нека да отбележим, че тъй като $b, \bar{B}, \|D\|_{B\infty}$ не се променят съществено при различни стойности на n разглеждаме $TERR$ само като функция на n. При това изхождайки от разгледаните два случая на нестрого и строго блочно диагонално преобладаване по стълбове на матрицата A (виж оценките(4.62) и (4.64)) $TERR(n)$ има съответно следния вид:

$$TERR(n) = \frac{bn^2\left(\bar{B}\|X\|_{B\infty} + (\log_2 n + 1)\|D\|_{B\infty}\right)\rho_0}{\|X\|_{B\infty}}, \tag{4.66}$$

$$TERR(n) = \frac{bg(s)\left(\bar{B}\|X\|_{B\infty} + (\log_2 n + 1)\|D\|_{B\infty}\right)\rho_0}{\|X\|_{B\infty}}. \tag{4.67}$$

Целта при направените числени експерименти е да покажем, че изведените теоретични оценки са почти достижими. Нека да отбележим, че навсякъде в направените числени експерименти в качеството на точно решение е избран вектор от единици.

За илюстрация, най-напред ще използваме блочно тридиагонални системи, които се пораждат при дискретизация по т. нар. в литературата Boundary Value Method [12] на следната диференциална задача от типа на Коши:

$$\begin{cases} y'(t) = M(t)y(t) + v(t), & t \in [t_0, t_f], \\ y(t_0) = y_0, \end{cases}$$

където матрицата $M(t) \in \mathcal{R}^{N\times N}$. Нека да отбележим, че в [12] е дискутирана подробно получената дискретна задача и са анализирани вариантите за използване на различни подходи за числено интегриране. За нашите цели е достатъчно да изберем $M(t) = M$, матрица не зависеща от t и като правило за интегриране да използваме метода на правоъгълниците. В този случай блочните елементи на матрицата A се определят по следния начин (виж [12]):

$$A_i = -I_N, \quad B_i = -2hM, \quad C_i = I_N,$$

където h е стъпка по мрежата за дискретизация, която считаме за константа (в направените експерименти е избрано $h = 0.1$). Така дефинираната матрица A не притежава свойството на блочно диагонално преобладаване по стълбове, но след една стъпка по изследвания метод на цикличната

n	100	200	500	1000
FE	1.16E–13	4.04E–13	7.26E–12	1.67E–11
$TERR(n)$	6.34E–12	2.53E–11	1.58E–10	6.33E–10

Таблица 4.2: Права грешка и финна горна граница за тази грешка при различни стойности на n в Пример 4.1

n	100	200	500	1000
FE	1.22E–15	1.24E–15	7.77E–16	1.55E–15
$TERR(n)$	3.93E–15	4.34E–15	4.88E–15	5.29E–15

Таблица 4.3: Права грешка и финна горна граница за тази грешка при различни стойности на n в Пример 4.2

редукция, новата матрица $A^{(1)}$ притежава желаното свойство, при което нейните блочни елементи се определят по следния начин:

$$A_i^{(1)} = \frac{1}{2h}M^{-1}, \quad B_i^{(1)} = -2hM - \frac{1}{h}M^{-1}, \quad C_i^{(1)} = \frac{1}{2h}M^{-1}.$$

Грешката не се променя съществено ако започнем изчисленията от оригиналната матрица A, тъй като изчисленията на блочните елементи $A_i^{(1)}, B_i^{(1)}$, $C_i^{(1)}$ зависят главно от обусловеността на матрицата M. В нашите примери в качеството на M използваме матрици, за които е изпълнено

$$\|M^{-1}\|_2\|M\|_2 \leq 2.24,$$

т.е. обусловеността е много добра, откъдето можем да очакваме, че $A_i^{(1)}, B_i^{(1)}, C_i^{(1)}$ са пресметнати с минимални грешки.

В нашите тестове сме избрали $N = 4$, откъдето $C_N = O(1)$ и в качеството на точно решение $X = (1, 1, \ldots, 1)^T$.

Пример 4.1 Нека матрицата M има вида

$$M = \begin{pmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{pmatrix}^{1/2}.$$

Тогава матрицата $A^{(1)}$, която се поражда, е с нестрого блочно диагонално преобладаване по стълбове. Получените резултати за правата грешка,

дефинирана чрез (4.65) и стойностите на функцията $TERR(n)$ определена чрез (4.66) за различни n са представени в Таблица 4.2. При това s е в граници $0.9999 \leq s < 1$, а грешката е от порядък $O(n^2 \log_2 n)\rho_0$. За да установим верността на това можем да намерим приближение по метода на най-малките квадрати с функция от вида $const\ TERR(n)$ на данните за грешката от Таблица 4.2. Резултатът е $const$ = 2.75E–2, а грешката от метода на най-малките квадрати е 4.03E–19. Откъдето можем да направим извода, че макар и да има някакво надценяване реалната грешка може да расте като $O(n^2 \log_2 n)\rho_0$. Следователно теоретично получената оценка (4.61) е почти достижима.

Пример 4.2 Нека матрицата M има вида

$$M = \begin{pmatrix} 1/(4h^2) & -1 & -1 & -1 \\ -1 & 1/(4h^2) & -1 & -1 \\ -1 & -1 & 1/(4h^2) & -1 \\ -1 & -1 & -1 & 1/(4h^2) \end{pmatrix}^{1/2}.$$

Тогава матрицата $A^{(1)}$, която се поражда е със строго блочно диагонално преобладаване по стълбове. Получените резултати за правата грешка, дефинирана чрез (4.65) и стойностите на функцията $TERR(n)$ определена чрез (4.67) за различни n са представени в Таблица 4.3. При това $s \approx 0.5, g(s) \approx 4$, а грешката е от порядък $O(\log_2 n)\rho_0$. Отново по метода на най-малките квадрати можем да намерим приближение с функция от вида $const\ TERR(n)$ на данните за грешката от Таблица 4.3. Резултатите са за $const$ = 2.56, и за грешката от метода на най-малките квадрати 4.79E–29. Така можем да направим извода и че теоретично получената оценка (4.64) е почти достижима.

Пример 4.3 Нека накрая да разгледаме и един пример, който се появява при числено решаване на частни диференциални уравнения. При дискретизация по метода на диференчните схеми на следната диференциална задача

$$\left| \begin{array}{l} au_{xx} + bu_{yy} = f(x,y), \\ u|_{\partial D} = 0, \quad D = \{0 \leq x \leq 1,\ 0 \leq y \leq 1\} \end{array} \right. ,$$

където са избрани $a = b = -1$, се получава блочно тридиагонална система линейни уравнения, чиито блокове се определят по следния начин:

$$A_i = -I_N, \quad B_i = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \in \mathcal{R}^{N \times N}, \quad C_i = -I_N.$$

n	10	12	15	20
FE	4.38E–15	4.96E–15	5.66E–15	6.58E–15
$TERR(n)$	6.23E–14	6.50E–14	6.82E–14	7.24E–14

Таблица 4.4: Права грешка и финна горна граница за тази грешка при различни стойности на n в Пример 4.3

Така породената матрица A е със строго блочно диагонално преобладаване по стълбове. Получените резултати за правата грешка (4.65) и стойностите на функцията $TERR(n)$ определена чрез (4.67) за различни n, където размерността N на блоковете е избрана така че $N = n$, са представени в Таблица 4.4. В този случай $s \approx 0.83, g(s) \approx 36$, а грешката е от порядък $O(\log_2 n)\rho_0$. Отново по метода на най-малките квадрати можем да намерим приближение с функция от вида $const\ TERR(n)$ на данните за грешката от Таблица 4.3. Резултатите са за $const = 8.10\text{E–}2$, и за грешката от метода на най-малките квадрати 1.52E–26. Така и в този пример се вижда че теоретично получената оценка (4.64) е почти достижима.

Библиография

[1] Андреев, А., Хр. Джиджев, Б. Сендов, Н. Янев. *Обзор в паралелното смятане.* Единен център по математика и механика, БАН, София, 1985.

[2] Воеводин, В. *Вычислительные основы линейной алгебры.* Наука, Москва, 1977.

[3] Воеводин, В., Е. Тыртышников. *Вычислительные процессы с теплицевыми матрицами.* Наука, Москва, 1987.

[4] Вълчанов, Н., М. Константинов. *Съвременни компютърни методи за компютърни пресмятания, Част 1.* Студии на БИАП, Математически лекции, София, 1996.

[5] Сендов, Б., В. Попов. *Числени методи. Част 2.* София, "Наука и изкуство", 1976.

[6] Ялымов, П. *Новый метод исследования ошибок округления и его приложения.* Диссертация, МГУ, Москва, 1990.

[7] Amodio, P. Optimized cyclic reduction for the solution of linear tridiagonal systems on parallel computers. *Comput. math. Appl.*, 26 (1993), 45–53.

[8] Amodio, P., L. Brugnano. Parallel factorizations and parallel solvers for tridiagonal linear systems. *Linear Algebra and Its Applications*, 172 (1992), 347–364.

[9] Amodio, P., L. Brugnano. The parallel QR factorization algorithm for tridiagonal linear systems. *Parallel Computing*, 21 (1995), 1097–1110.

[10] Amodio, P., L. Brugnano, T. Politi. Parallel factorizations for tridiagonal matrices. *SIAM J. Numer. Anal.*, 30 (1993), 813–823.

[11] Amodio, P., F. Mazzia. Backward error analysis of cyclic reduction for the solution of tridiagonal systems. *Math. Comp.*, 62 (1994), 601–617.

[12] Amodio, P., F. Mazzia, D. Trigiante. Stability of some boundary value methods for the solution of initial value problems. *BIT*, 33 (1993), 434–451.

[13] Ashcraft, C. Parallel reduction methods for the solution of banded systems of equations. *Research Rept. GMR-5094*, General Motors Research Labs., Computer Science Department (1985).

[14] Axelsson, O. *Iterative solution methods.* Cambridge University Press, New York, 1994.

[15] Babuska, I. Numerical stability in problems of linear algebra. *SIAM J. Numer. Anal.*, 9 (1972), 53–77.

[16] Balle, S., P. Hansen, N. Higham. A Strassen-type matrix inversion algorithm for the connection machine. APPARC PaA2 Deliverable, Esprit BRA Contract # 6634; *Report UNIC-93-11, UNI•C*, October 1993.

[17] Bauer, F. Computational graphs and rounding errors. *SIAM J. Numer. Anal.*, 11 (1974), 87–96.

[18] Berman, A., R. Plemmons. *Nonnegative matrices in the mathematical sciences.* Academic Press, New York, 1979.

[19] Bondeli, S. *Divide and Conquer: a new parallel algorithm for the solution of a tridiagonal linear system of equations.* Tech. Report 130, ETH Zürich, Department Informatik, Institut für Wissenschaftliches Rechnen, Zürich, Switzerland, 1990.

[20] Brugnano, L. A parallel solver for tridiagonal linear systems for distributed memory parallel computers. *Parallel Computing*, 17 (1991), 1017–1023.

[21] Buneman, O. A compact non-iterative Poisson solver. *Report 294*, Stanford University Institute for Plasma Research, Stanford CA, 1969.

[22] Buzbee, B., G. Golub, C Nielson. On direct methods for solving Poisson's equations. *SIAM J. Numerical Analysis*, 7 (1970), 627–656.

[23] Concus, P., P. Saylor. A modified direct preconditioner for indefinite symmetric Toeplitz systems. *Numerical Linear Algebra and Applications*, 2 (1995), 497–514.

[24] Conroy, J. Parallel Algorithms for the solution of narrow banded systems. *Appl. Numer. Math.*, 5 (1989), 409–421.

[25] Dongarra, J., A. Sameh. One some parallel banded system solvers. *Parallel Comput.*, 1 (1984), 223–235.

[26] Dorr, F. An example of ill-conditioning in the numerical solution of singular perturbed problems. *Math. Comp.*, 25 (1971), 271–283.

[27] Forsythe, G., M. Malcolm, C. Moler. *Computer methods for mathematical computations.* Englewood Cliffs, Prentice Hall, 1977.

[28] Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchenk, V. Sunderam. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing.* The MIT Press, England, 1994.

[29] Goldstine, H., J. Von Neumann. Numerical inverting of matrices of higher order. *Bull. Amer. Math. Soc.*, 53 (1947), 1021–1099.

[30] Golub, G., C. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, 1989.

[31] Hajj, I., S. Skelboe. A multilevel parallel solver for block tridiagonal and banded linear systems. *Parallel Computing*, 15 (1990), 21–45.

[32] Hansen, P., P. Yalamov. Stabilization by perturbation of a $4n^2$ Toeplitz solver, *Preprint N25*, Technical University of Russe, January 1995, (изпратена за разглеждане в SIMAX).

[33] Heller, D. Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems. *SIAM J. Numer. Anal.*, 13 (1976), 484–496.

[34] Higham, D., N. Higham. Backward error and condition of structured linear systems. *SIAM J. Matrix Anal. Appl.*, 13 (1992), 162–175.

[35] Higham, N. Bounding the error in Gaussian elimination for tridiagonal systems. *SIAM J. Matrix Anal. Appl.*, 11 (1990), 521–530.

[36] Higham, N. *Accuracy and Stability of Numerical Algorithms.* SIAM, 1996.

[37] Hockney, R. A fast direct solution of Poisson's equation using Fourier analysis. *J. ACM*, 12 (1965), 95–112.

[38] Hockney, R., Jesshope, C. *Parallel Computers 2.* Adam Hilger, Bristol, 1988.

[39] Johnsson, S. Solving tridiagonal systems on ensemble architectures. *SIAM Journal on Scientific and Statistical Computing*, 8 (1987), 354–392.

[40] Johnsson, S. Solving narrow banded systems on ensemble architectures. *ACM Trans. Math. Software*, 11 (1985), 271–288.

[41] Kershaw, D. Solution of single tridiagonal linear systems and vectorizationof the ICCG algorithm on the CRAY 1. *Parallel Computations*, Academic Press, New York, 1982, 85–99.

[42] Krenchel, A., H. Plum, K. Stüben. Parallelizzation and vectorization aspects of the solution of tridiagonal linear systems. *Parallel Computing*, 14 (1990), 31–49.

[43] Larson, J., M. Pasternak, J. Wisniewski. Algorithm 594: Software for relative error analysis. *ACM Trans. Math. Software*, 9 (1983), 125–130.

[44] Larson, J., A. Sameh. Efficient calculation of the effects of roundoff errors. *ACM Trans. Math. Software*, 4 (1978), 228–236.

[45] Larson, J., A. Sameh. Algorithms for roundoff error analysis - a relative error approach. *Computing*, 24 (1980), 275–297.

[46] Meier, U. A parallel partition method for solving banded linear systems. *Parallel Comput.*, 2 (1985), 33–43.

[47] Miller, W. Remarks on the complexity of roundoff errors. *Computing*, 12 (1974), 149–161.

[48] Miller, W. Software for roundoff error analysis. *ACM Trans. Math. Software*, 1 (1975), 108–128.

[49] Miller, W. Graph transformations for roundoff analysis. *SIAM J. Comput.*, 5 (1976), 204–216.

[50] Miller, W., D. Spooner. Software for roundoff error analysis II. *ACM Trans. Math. Software*, 4 (1978), 369–387.

[51] Pavlov, V. Iterative refinement for ill-conditioned cyclic reduction. *Proc. Fifth International Conference on Differential Equations and Applications*, (Eds. S. Bilchev and S. Tersian), Rousse, August 24–29, 1995, 84–95.

[52] Pavlov, V., D. Todorova. Stabilization and experience with the partitioning method for tridiagonal systems. *Lecture Notes in Computer Science*, (Eds. L. Vulkov, J. Wasniewski and P. Yalamov), Springer, 1196 (1997), 424–431.

[53] Pavlov, V., P. Yalamov. Stabilization by perturbation of ill-conditioned cyclic reduction. *International Journal of Computer Mathematics*, 68 (1998), 273–283. (приета за печат).

[54] Peters, G., J. Wilkinson. On the stability of Gauss-Jordan elimination with pivoting. *Commun. ACM*, 18 (1975), 20–34.

[55] Saad, Y., H. Shultz. Parallel direct methods for solving banded linear systems. *Linear Algebra Appl.*, 88/89 (1987), 623–650.

[56] Skeel, R. Scaling for numerical stability in Gaussian elimination. *J. Assoc. Comput. Mach.*, 26 (1979), 494–526.

[57] Stone, H. An efficient parallel algorithm for the solution of a tridiagonal linear system of equations. *J. Assoc. Comput. Mach.*, 20 (1973), 27–38.

[58] Stone, H. Parallel tridiagonal solvers. *ACM Trans. Math. Software*, 1 (1975), 289–307.

[59] Stummel, F. Perturbation theory for evaluation algorithms of arithmetic expressions. *Math. Comp.*, 37 (1981), 435–473.

[60] Voevodin, V., P. Yalamov. A new method of roundoff error estimation. *Proc. Workshop Parallel and Distributed Processing*, (Eds. K. Boyanov), Elsevier, Amsterdam, 1990, 315–333.

[61] Vorst, H. Large tridiagonal and block tridiagonal linear systems on vector and parallel computers. *Parallel Computing*, 5 (1987), 45–54.

[62] Vorst, H. Analysis of a parallel solution method for tridiagonal linear systems. *Parallel Computing*, 5 (1987), 303–311.

[63] Walshaw, C. Diagonal dominance in the parallel partition method for tridiagonal systems. *SIAM Journal Matrix Anal. Appl.*, 16 (1995), 1086–1099.

[64] Wang, H. A parallel method for tridiagonal linear systems. *ACM Transactions on Mathematical Software*, 7 (1981), 170–183.

[65] Wilkinson, J. Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, 8 (1961), 281–330.

[66] Wilkinson, J. *Rounding errors in algebraic processes.* Prentice Hall, Englewood Cliffs, 1963.

[67] Wilkinson, J. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, 1965.

[68] Wright, S. Parallel algorithms for banded linear systems. *SIAM J. Sci. Stat. Comput.*, 12 (1991), 824–842.

[69] Yalamov, P. On the stability of the cyclic reduction without back substitution for tridiagonal systems. *BIT*, 34 (1994), 428–447.

[70] Yalamov, P. Graphs and stability of algorithms. *Numer. Math.* (изпратена за разглеждане).

[71] Yalamov, P. Stability of a partitioning algorithm for bidiagonal systems. *Parallel Computing* (приета за печат).

[72] Yalamov, P. Convergence of the iterative refinement procedure applied to stabilization of a fast Toeplitz solver. *Proc. Second IMACS Symposium on Iterative Methods in Linear Algebra*, Eds. P. Vassilevski and S. Margenov, IMACS, 1996, 354–363.

[73] Yalamov, P., V. Pavlov. Stability of the block cyclic reduction. *Linear Algebra and Its Applications*, 249 (1996), 341–358.

[74] Yalamov, P., V. Pavlov. On the stability of a partitioning algorithm for tridiagonal systems. *SIAM J. Matrix Anal. Appl.* (приета за печат).

[75] Yalamov, P., V. Pavlov. Backward stabilty of a parallel partitioning algorithm for banded linear systems. *Proc. 4th International Conference on Numerical Methods and Applications*, Sofia, August 19–23, 1998 (приета за печат).