

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

МАСТЕР РАД

MULTIGRID МЕТОДА ЗА НУМЕРИЧКО
РЕШАВАЊЕ ПАРЦИЈАЛНИХ
ДИФЕРЕНЦИЈАЛНИХ ЈЕДНАЧИНА
ЕЛИПТИЧКОГ ТИПА

Студент: Јелена Томановић

Број индекса: 1013/2011

Ментор: Проф. др Бошко Јовановић

Београд, 2012.

Садржај

1	Увод	1
1.1	Мотивациони пример	2
1.1.1	Глобални и локални процес	2
1.1.2	Multiscale приступ	4
1.2	Класификација линеарних парцијалних диференцијалних једначина другог реда	6
1.3	Моделни проблем	8
2	Дискретизација	9
2.1	Једнодимензиона Poisson-ова једначина са хомогеним Dirichlet-овим граничним условом	9
2.2	Дводимензиона Poisson-ова једначина са хомогеним Dirichlet-овим граничним условом	17
3	Итеративне методе	23
3.1	Норме вектора и матрица	24
3.2	Стандардне итеративне методе	25
3.2.1	Jacobi-јева итеративна метода	29
3.2.2	Gauss-Seidel-ова итеративна метода	30
3.3	Ограничења стандардних итеративних метода	31
3.3.1	Локалност и брзина преноса информација	31
3.3.2	Сопствене вредности и редуковање грешке	32
4	Multigrid метода	36
4.1	Основне идеје multigrid-а	36
4.2	Multiscale приступ	37
4.3	Преклапање сопствених вектора	39
4.4	Грешка дискретизације и алгебарска грешка. Резидуална једначина	41
4.5	Multigrid алгоритам за једнодимензиони проблем	42
4.6	Multigrid алгоритам за дводимензиони проблем	45
4.7	FMG алгоритам	46
4.8	Анализа сложености (дводимензиони проблем)	48
4.8.1	Анализа сложености MG V алгоритма	48
4.8.2	Анализа сложености FMG алгоритма	49
4.9	Пример	50
4.10	Примене multigrid-а	62
5	Закључак	63

1 Увод

Multigrid¹ спада међу релативно новије методе у нумеричкој анализи. Први радови везани за ову тематику појавили су се '60-тих година двадесетог века. Њихови аутори били су Fedorenko² и Bakhvalov³, а односили су се на решавање елиптичких граничних проблема (видети [1], [2] и [3]). Нешто касније, '70-тих година, Brandt⁴ је показао практичну примену и ефикасност multigrid-а за уопштене проблеме, па се он сматра главним творцем ове методе (видети [4], [5] и [6]).

Иако ћемо у раду користити термин “multigrid метода”, напоменимо да то није само једна метода, нити фамилија метода - то је читав приступ решавању нумеричких проблема, колекција идеја и ставова, које је Brandt називао multi-level⁵ техникама.

До сада су multigrid, као и бројне multiscale⁶ методе настале из њих, развијене и примењене на многе проблеме у разним областима (о чему ће касније бити речи).

Multigrid је данас познат као најбржа метода за решавање елиптичких граничних проблема, што представља основну мотивацију за писање рада на ову тему.

¹Термин преузет из енглеског језика, а значи “вишемрежни”.

²Radii Petrovich Fedorenko (1930 - 2009), руски математичар.

³Nikolai Sergeevitch Bakhvalov (1934 - 2005), руски математичар.

⁴Achi Brandt (1938 -), израелски математичар.

⁵Термин преузет из енглеског језика, а значи “вишестепени”.

⁶Термин преузет из енглеског језика, а значи “вишескални”.

1.1 Мотивациони пример

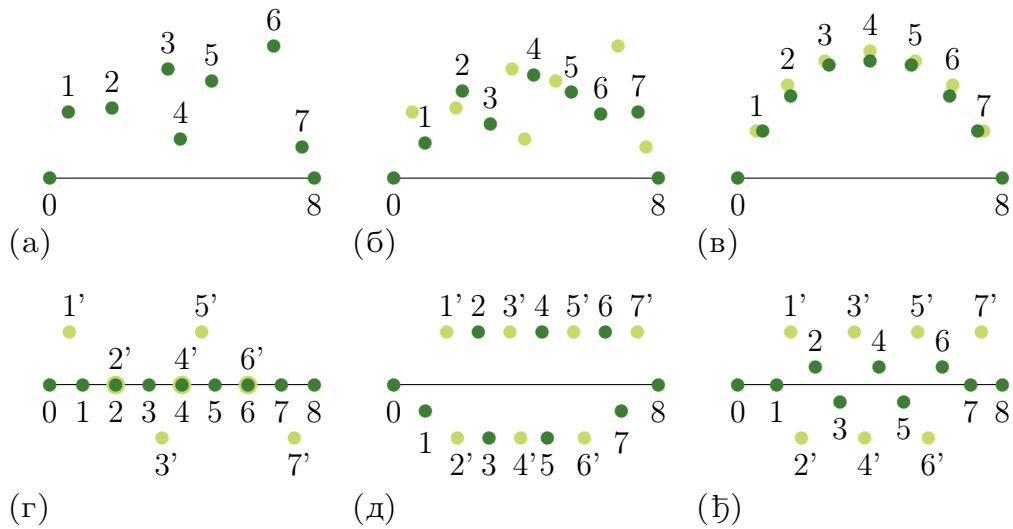
Ради уочавања веза и разлика између глобалног и локалног процеса, као и истицања предности multiscale приступа, размотримо следећу причу.

Како би установио какво стање влада у једној касарни, генерал је одлучио да је изненада посети. Капетан који руководи поменутом касарном приметио је како се из даљине приближавају генералова кола. Пошто већ дуже време жели унапређење, одлучио је покушати да задиви генерала тако што ће се сви војници поређати дуж једне праве линије поред пута, при чему ће размак између свака два суседна војника бити једнак, и тако га дочекати. С обзиром да су кола све ближе, капетан мора да смисли најбржи начин за реализацију своје идеје. Поменимо да се тај капетан зове Јакоби.

1.1.1 Глобални и локални процес

Пошто се у касарни налази $n + 1$ војник, Јакоби их је нумерисао бројевима од 0 до n . Затим је наредио 0-том и n -том да стану, респективно, на леви и десни крај линије дуж које ће се и остали распоредити. Дужина те линије (дакле, и растојање између 0-тог и n -тог војника) је l . Даље, Јакоби размишља да нареди i -том војнику, $i = 1, 2, \dots, n - 1$, да стане на тачку линије која се налази на растојању il/n од 0-тог војника. То би био *глобални процес* који би директно решио проблем. Међутим, Јакоби није сигуран да ли ће сваки војник умети тачно да одреди где се налази место на које треба да стане. Једино у шта је сигуран јесте то да ће i -ти војник, $i = 1, 2, \dots, n - 1$, умети довољно добро да процени где се налази средина дужи између $(i - 1)$ -вог и $(i + 1)$ -вог војника - што представља *локални процес*.

Наредна Јакобијева идеја је да примени следећи итеративни локални процес: на почетку сваке итерације он дуне у пиштаљку и



Слика 1: Тамно зелени кружићи представљају тренутне, а светло зелени претходне позиције, пре последњег звука пиштаљке: (а) почетне позиције, (б) позиције након прве итерације, (в) спора конвергенција, (г) брза конвергенција, (д) претрчавање на другу страну линије, (ђ) увођење релаксационог параметра.

i -ти војник, $i = 1, 2, \dots, n-1$, трчи у тачку која се налази на средини дужи на чијим крајевима су били $(i-1)$ -ви и $(i+1)$ -ви војник пре него што се чула пиштаљка. Ове итерације би се понављале све док грешка у позицијама војника не буде довољно мала да је генерал не примети - то је *критеријум конвергенције*.

Осим 0-тог и n -тог војника, који су заузели своје коначне позиције пре прве итерације, остали су насумично распоређени по касарни, што је приказано на слици 1а (узето је $n = 8$). Слика 1б приказује позиције након прве итерације. Испитајмо конвергенцију ове итеративне методе.

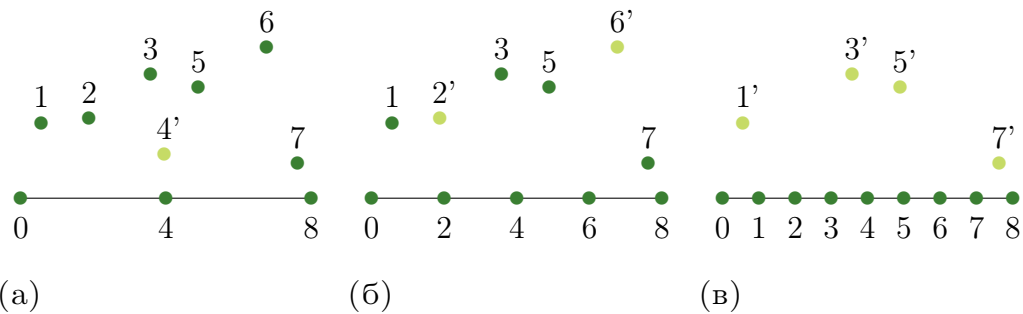
Може се десити да метода конвергира, али врло споро. Претпоставимо да се војници налазе у позицији приказаној на слици 1в. У овом случају прелазе мала растојања у свакој итерацији, али, нажалост, метода споро конвергира. Насупрот томе, конвергенција се постиже након само једне итерације, ако се војници налазе у позицији приказаној на слици 1г.

Зашто је конвергенција на слици 1в спора, а на слици 1г брза? На слици 1в имамо грешку велике скале. Али, процес је локалан и укључује само информације мале скале, тј. нова позиција војника

зависи само од његових првих суседа. Зато се сваком војнику са његове тачке гледишта чини да је грешка у позицији мала, док заправо није. Поента је да не можемо исправити оно што не можемо приметити, а грешка велике скале се не може приметити локално. Насупрот томе, слика 1г приказује грешке мале скале које могу бити ефикасно уочене и поправљене помоћу локалног процеса. Размотримо позицију на слици 1д. Очигледно постоји грешка мале скале, али је описани процес не поправља ефикасно; заправо, војници би само претрчали на другу страну линије. Видимо да локални процес неће у свим ситуацијама дати добре резултате. Међутим, овај проблем се лако решава увођењем константе зване *релаксациони параметар*. То значи да ће у свакој итерацији сваки војник прећи само део пута до места где би требало да стане. На пример, на слици 1ђ релаксациони параметар је $2/3$, што значи да сваки војник прелази $2/3$ пута до места на које би требало да стане.

1.1.2 Multiscale приступ

Главна идеја multigrid методе је да се користи локални процес, али да се то ради на свим скалама проблема. Ради једноставности, претпоставимо да је n степен броја 2. У нашој причи, капетан разматра и multiscale приступ. Почине од велике скале, у којој се проблем састоји од распоређивања само војника нумерисаних са 0, $n/2$ и n . Након звука пиштаљке, војник $n/2$ помера се на средину дужи између својих првих суседа на великој скали - војника 0 и n (који већ стоје на крајевима линије) (слика 2а). Затим прелази на средњу скалу, у којој се активирају и војници $n/4$ и $3n/4$, који се након звука пиштаљке померају на средину дужи између својих првих суседа на средњој скали - $n/4$ између 0 и $n/2$, а $3n/4$ између $n/2$ и n (напоменимо да војници, који су у претходној итерацији већ заузели своја права места, у наредној итерацији остају на њима) (слика 2б). Преласком на мању скалу активирају се и војници $n/8$, $3n/8$, $5n/8$ и $7n/8$, па је у случају $n = 8$ процес завршен (слика 2в). Дакле, након само $\log_2 n$ итерација и $n - 1$ индивидуалних померања, проблем је решен помоћу локалног процеса.



Слика 2: Распоређивање војника употребом multiscale приступа. Тамно зелени кружићи представљају тренутне, а светло зелени претходне позиције, пре последњег звука пиштаљке: (а) велика, (б) средња, (в) мала скала.

Пошто се у касарни налази 257 војника (дакле, $n = 257 - 1 = 256 = 2^8$), овим поступком их је могуће распоредити у само 8 итерација. Јакоби је тако учинио, а генерал, који је све то посматрао, био је задивљен и Јакоби је унапређен у мајора.

Напоменимо да решавање произвољног проблема multiscale приступом није увек једноставно (што би се могло помислити на основу претходне приче). Потребно је размотрити четири главна задатка:

- избор погодног локалног процеса,
- избор погодних променљивих велике скале (тзв. грубих променљивих),
- избор погодне методе за пренос информација кроз скале,
- развијање погодних једначина (или процеса) за грубе променљиве.

У зависности од проблема који решавамо, сваки од наведених задатака може бити различите тежине - од изузетно лак, до изузетно тежак.

1.2 Класификација линеарних парцијалних диференцијалних једначина другог реда

Најчешћи пример у математици где се користи локално правило за решавање глобалног проблема јесу диференцијалне једначине. Код њих тражимо опште решење (глобални проблем) помоћу локалног правила на изводима (плус гранични услови).

Једначину облика

$$\sum_{i,j=1}^n a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^n b_i \frac{\partial u}{\partial x_i} + cu = f, \quad (1)$$

где су a_{ij} , b_i , c и f познате функције од $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, називамо *линеарном парцијалном диференцијалном једначином другог реда*. Функције a_{ij} , b_i и c називамо *коэффицијентима*, а f *десном страном* једначине (1). Ако је $f(x) = 0$, једначину (1) називамо *хомогеном*. Израз

$$Lu \equiv \sum_{i,j=1}^n a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^n b_i \frac{\partial u}{\partial x_i} + cu$$

називамо *диференцијалним изразом* или *диференцијалним оператором*.

За $i \neq j$, Lu садржи члан

$$(a_{ij} + a_{ji}) \frac{\partial^2 u}{\partial x_i \partial x_j}.$$

Коефицијент уз $\frac{\partial^2 u}{\partial x_i \partial x_j}$ можемо поделити на два сабирка на произвољан начин, па без смањења општости можемо сматрати да је

$$a_{ij}(x) = a_{ji}(x).$$

Сматраћемо да су коэффицијенти $a_{ij}(x)$ реалне функције.

Нека је x фиксирана тачка из \mathbb{R}^n . Под претходним претпоставкама, матрица $A(x) = (a_{ij}(x))_{i,j=1}^n$ је реална и симетрична и има n сопствених вредности (узимајући у обзир вишеструкост). Нека је:

- n_+ - број позитивних сопствених вредности,
- n_- - број негативних сопствених вредности,

n_0 - број нултих сопствених вредности.

Тада кажемо да једначина (1) у тачки $x \in \mathbb{R}^n$ има тип (n_+, n_-, n_0) . Ако једначину (1) помножимо са -1 матрица $A(x)$ прелази у $-A(x)$, а бројеви n_+ и n_- мењају места. Зато ћемо сматрати да су типови (n_+, n_-, n_0) и (n_-, n_+, n_0) међусобно идентични. Специјално:

тип $(n, 0, 0)$ називамо *елиптичким*,

тип $(n - 1, 1, 0)$ називамо *хиперболичким*,

тип $(n - 1, 0, 1)$ називамо *параболичким*.

У овом раду разматрамо једначине елиптичког типа. Њима се описују бројне стационарне појаве, тј. појаве које не зависе од времена, као што је, на пример, дифузија, потенцијално електро-статичко поље, равнотежни положај затегнуте мембране, потенцијални ток инкомпресибилног флуида, стационарно провођење топлоте, итд.

Ако једначина (1) припада типу (n_+, n_-, n_0) у свакој тачки области $\Omega \subset \mathbb{R}^n$, тада кажемо да она припада том типу у области Ω .

Да би се издвојило јединствено решење једначине (1) у некој области $\Omega \subset \mathbb{R}^n$, једначини се додају гранични услови, на пример:

$$u(x) = g(x), \quad x \in \partial\Omega$$

- први гранични услов (Dirichlet⁷-ов),

$$\sum_{i,j=1}^n a_{ij} \frac{\partial u}{\partial x_j} \cos(n, x_i) = g(x), \quad x \in \partial\Omega$$

- други гранични услов (Neumann⁸-ов),

$$\sum_{i,j=1}^n a_{ij} \frac{\partial u}{\partial x_j} \cos(n, x_i) + \sigma(x)u(x) = g(x), \quad x \in \partial\Omega$$

- трећи гранични услов (Robin⁹-ов).

Граничне услове у којима је $g(x) = 0$ називамо *хомогеним*.

Једначину (1), с једним од набројаних граничних услова, називамо *граничним проблемом*.

⁷Johann Peter Gustav Lejeune Dirichlet (1805 - 1859), немачки математичар.

⁸Carl Gottfried Neumann (1832 - 1925), немачки математичар.

⁹Victor Gustave Robin (1855 - 1897), француски математичар.

1.3 Моделни проблем

Нека је $\Omega = (0, 1)^n$. Као моделни задатак разматрамо Poisson¹⁰-ову једначину

$$-\Delta u(x) = f(x), \quad x \in \Omega$$

са хомогеним Dirichlet-овим граничним условом

$$u(x) = 0, \quad x \in \partial\Omega.$$

Poisson-ова једначина је елиптичког типа, јер су јој све сопствене вредности једнаке -1 (дакле, све су негативне).

Бавићемо се једнодимензионим

$$\begin{aligned} -u''(x) &= f(x), \quad x \in \Omega, \\ u(0) &= u(1) = 0, \end{aligned}$$

и дводимензионим проблемом

$$\begin{aligned} -\Delta u &\equiv -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x, y), \quad (x, y)^T \in \Omega, \\ u(x, y) &= 0, \quad (x, y)^T \in \Gamma \equiv \partial\Omega. \end{aligned}$$

¹⁰Siméon Denis Poisson (1781 - 1840), француски математичар и физичар.

2 Дискретизација

2.1 Једнодимензиона Poisson-ова једначина са хомогеним Dirichlet-овим граничним условом

Пређимо на мање тривијалан задатак од мотивационог примера. Нека је $\Omega = (0, 1)$. Разматрамо проблем

$$\begin{aligned} -u''(x) &= f(x), & x \in \Omega, \\ u(0) &= u(1) = 0, \end{aligned} \tag{2}$$

који се може записати у операторском облику

$$Lu = f. \tag{3}$$

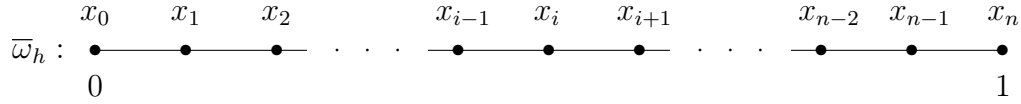
Како бисмо овај проблем свели на систем линеарних једначина, користимо *методе коначних разлика*, које се још називају и *методе мреже*. Ове методе се заснивају на замени извода количницима коначних разлика. Прво се изабере коначно много тачака сегмента $[0, 1]$ и оне чине *мрежу*. Изабране тачке називају се *чворови мреже*. Ако су чворови равномерно распоређени, кажемо да је мрежа равномерна и дефинисана је *кораком* h - растојањем између два суседна чвора (Слика 3),

$$\bar{\omega}_h = \{x_i \mid x_i = ih, i = 0, 1, \dots, n, h = 1/n\}.$$

У разматраном задатку потребна је апроксимација другог извода $u''(x)$. На равномерној мрежи могуће апроксимације у тачки x_i првог извода u' су:

$$u_{x,i} = \frac{u(x_{i+1}) - u(x_i)}{h}$$

- количници коначних разлика унапред,



Слика 3: Равномерна мрежа на сегменту $[0, 1]$.

$$u_{\bar{x},i} = \frac{u(x_i) - u(x_{i-1}))}{h}$$

- количници коначних разлика уназад,

$$u_{\dot{x},i} = \frac{u(x_{i+1}) - u(x_{i-1}))}{2h} = \frac{u_{x,i} + u_{\bar{x},i}}{2}$$

- централни количници коначних разлика,
док се други извод u'' може апроксимирати са:

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

- централни количници коначних разлика.

Под претпоставком да је функција $u(x)$ довољно глатка, развојем у Taylor¹¹-ов ред можемо оценити грешку ових апроксимација. На пример,

$$\begin{aligned} u'(x_i) - u_{x,i} &= u'(x_i) - \frac{1}{h} \left[u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i + \theta h) - u(x_i) \right] \\ &= -\frac{h}{2}u''(x_i + \theta h). \end{aligned}$$

Тако добијамо оцене

$$\begin{aligned} u'(x_i) &= u_{x,i} + O(h) = u_{\bar{x},i} + O(h) = u_{\dot{x},i} + O(h^2), \\ u''(x_i) &= u_{\bar{x}x,i} + O(h^2). \end{aligned} \tag{4}$$

Када $h \rightarrow 0$, тј. када се мрежа згушњава, апроксимације теже вредностима извода функције $u(x)$ у чворовима. При томе, конвергенција је бржа код апроксимације централним количницима коначних разлика $u_{\dot{x}}$ и $u_{\bar{x}x}$, јер је главни члан грешке $O(h^2)$.

Заменом функције $u(x)$ и њених извода у проблему (2) одговарајућим количницима коначних разлика у чворовима мреже $\bar{\omega}_h$, вршимо дискретизацију полазног проблема. Континуалну величину

¹¹Brook Taylor (1685 - 1731), енглески математичар.

$u(x)$ замењујемо вектором $\mathbf{v} = (v_0, v_1, \dots, v_n)^T$, при чему је $v_i \approx u(x_i)$, $i = 0, 1, \dots, n$, а проблем (2) системом линеарних једначина по \mathbf{v} ,

$$\begin{aligned} -v_{\bar{x}x,i} &= f_i, & i = 1, 2, \dots, n-1, \\ v_0 &= v_n = 0, \end{aligned}$$

односно, у развијеном облику

$$\begin{aligned} -\frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} &= f_i, & i = 1, 2, \dots, n-1, \\ v_0 &= v_n = 0, \end{aligned} \tag{5}$$

где је $f_i = f(x_i)$, $i = 1, 2, \dots, n-1$.

Дискретни проблем (5), који можемо записати у облику операторске једначине

$$L_h v = f_h, \tag{6}$$

назива се *диференцијском схемом*.

Дискретизација је добро извршена уколико, пре свега, дискретни проблем (6) има јединствено решење. Даље, неопходно је да решење дискретног проблема конвергира ка решењу полазног проблема када $h \rightarrow 0$, тј. да грешка $\epsilon = u - v$ тежи нули када $h \rightarrow 0$. Ово ће бити испуњено уколико је схема конзистентна и стабилна. Конзистентност схеме (6) у односу на проблем (2) значи да

$$L_h u \rightarrow Lu, \quad f_h \rightarrow f, \quad \text{када } h \rightarrow 0,$$

што за схему (5) следи из (4). Стабилност схеме (6) своди се на стабилност система линеарних једначина. Укратко, схема (6) ће бити стабилна ако је L_h^{-1} униформно ограничен оператор. Дакле, ако је схема (6) конзистентна и стабилна, конвергенција непосредно следи, јер је

$$L_h(u - v) = L_h u - L_h v = L_h u - f_h + f - Lu = (L_h u - Lu) + (f - f_h),$$

те

$$u - v = L_h^{-1}(L_h u - Lu) + L_h^{-1}(f - f_h) \rightarrow 0, \quad \text{када } h \rightarrow 0.$$

Испитајмо још егзистенцију и јединственост решења проблема (6). У операторском запису (6) L_h је линеаран оператор

$$L_h v_i = \begin{cases} v_{\bar{x},i}, & \text{за } i = 1, 2, \dots, n-1, \\ 0, & \text{за } i = 0 \text{ или } i = n, \end{cases} \quad (7)$$

који пресликава у самог себе векторски простор дискретних функција $V = \{v : \bar{\omega}_h \rightarrow \mathbb{R} \mid v_0 = v_n = 0\}$, док је f_h рестрикција функције $f(x)$ на мрежу $\bar{\omega}_h$, додефинисана нулом у граничним тачкама,

$$f_{h,i} = \begin{cases} f_i, & \text{за } i = 1, 2, \dots, n-1, \\ 0, & \text{за } i = 0 \text{ или } i = n. \end{cases}$$

Свакој дискретној функцији $v \in V$ можемо придружити вектор $\mathbf{v} = (v_1, v_2, \dots, v_{n-1})^T \in \mathbb{R}^{n-1}$, и обрнуто. Зато су простори V и \mathbb{R}^{n-1} хомоморфни. У векторском простору V дефинишемо скаларни производ

$$(v, w)_h = h \sum_{i=1}^{n-1} v_i w_i$$

и норме

$$\|v\|_h = (v, v)_h^{1/2}, \quad \|v\|_{C,h} = \max_{1 \leq i \leq n-1} |v_i|.$$

У односу на овако дефинисан скаларни производ, линеарни оператор L_h је самоконјугован и позитивно дефинисан.

Самоконјугованост следи из симетричности билинеарне форме

$$(L_h v, w)_h = h \sum_{i=1}^{n-1} (-v_{\bar{x},i}) w_i = h \sum_{i=0}^{n-1} v_{x,i} w_{x,i}.$$

Слично,

$$(L_h v, v)_h = h \sum_{i=0}^{n-1} v_{x,i}^2. \quad (8)$$

Да бисмо доказали позитивну дефинисаност овог оператора, оценимо величину v_i . Можемо је, узимајући у обзир да је $v_0 = v_n = 0$, изразити на следећи начин:

$$v_i^2 = \left(\sum_{j=1}^i v_j - \sum_{j=0}^{i-1} v_j \right)^2 + ih \left[\left(\sum_{j=i+1}^n v_j - \sum_{j=i}^{n-1} v_j \right)^2 - \left(\sum_{j=1}^i v_j - \sum_{j=0}^{i-1} v_j \right)^2 \right]$$

$$= (1 - ih) \left(h \sum_{j=0}^{i-1} v_{x,j} \right)^2 + ih \left(h \sum_{j=i}^{n-1} v_{x,j} \right)^2 .$$

Применом неједнакости Cauchy¹²-Schwarz¹³-а на последње две суме, имамо да је

$$v_i^2 \leq (1 - ih)ih^2 \sum_{j=0}^{i-1} v_{x,j}^2 + ih(n - i)h^2 \sum_{j=i}^{n-1} v_{x,j}^2 = ih(1 - ih) \left(h \sum_{j=0}^{n-1} v_{x,j}^2 \right) ,$$

јер је $(n - i)h = (nh - ih) = 1 - ih$. Стога је, на основу (8),

$$\begin{aligned} \|v\|_{C,h}^2 &= \max_{1 \leq i \leq n-1} v_i^2 \leq \left(h \sum_{j=0}^{n-1} v_{x,j}^2 \right) \max_{1 \leq i \leq n-1} ih(1 - ih) \\ &\leq \frac{1}{4} \left(h \sum_{j=0}^{n-1} v_{x,j}^2 \right) = \frac{1}{4} (L_h v, v)_h . \end{aligned}$$

Како је још

$$\|v\|_h^2 = h \sum_{i=1}^{n-1} v_i^2 \leq h(n - 1) \max_{1 \leq i \leq n-1} v_i^2 \leq \|v\|_{C,h}^2 ,$$

коначно добијамо да је

$$\|v\|_h^2 \leq \|v\|_{C,h}^2 \leq \frac{1}{4} \left(h \sum_{j=0}^{n-1} v_{x,j}^2 \right) = \frac{1}{4} (L_h v, v)_h , \quad (9)$$

што доказује позитивну дефинисаност оператора L_h .

Стога једначина (6), тј. схема (5), има решење и оно је јединствено.

Непосредна последица неједнакости (9)

$$4 \|v\|_{C,h}^2 \leq (L_h v, v)_h \leq \|L_h v\|_h^2 \|v\|_h^2 \leq \|L_h v\|_h^2 \|v\|_{C,h}^2 ,$$

су оцене

$$\|v\|_h \leq \|v\|_{C,h} \leq \frac{1}{4} \|L_h v\|_h . \quad (10)$$

¹²Augustin Louis Cauchy (1789 - 1857), француски математичар.

¹³Karl Hermann Amandus Schwarz (1843 - 1921), немачки математичар.

Да бисмо доказали конвергенцију схеме (5), уочимо да грешка $\epsilon = u - v$, која је као и v дефинисана само у чворовима мреже, задовољава диференцијску схему са истим оператором L_h , датим изразом (7), само са промењеном десном страном. Наиме, како је

$$-\epsilon_{\bar{x}x,i} = -(u_{\bar{x}x,i} - v_{\bar{x}x,i}) = -u_{\bar{x}x,i} - f_i = -u_{\bar{x}x,i} - (-u''(x_i)) = u''(x_i) - u_{\bar{x}x,i},$$

то је ϵ решење диференцијске схеме

$$\begin{aligned} -\epsilon_{\bar{x}x,i} &= u''(x_i) - u_{\bar{x}x,i}, \quad i = 1, 2, \dots, n-1, \\ \epsilon_0 &= \epsilon_n = 0, \end{aligned}$$

па је, према (3) и (10),

$$\|\epsilon\|_h \leq \|\epsilon\|_{C,h} \leq \frac{1}{4} \|u'' - u_{xx}\|_h = O(h^2).$$

Према томе схема (5) је конвергентна, јер је

$$\|\epsilon\|_h \leq \max_{1 \leq i \leq n-1} |u(x_i) - v_i| \rightarrow 0, \quad \text{када } h \rightarrow 0.$$

С обзиром да ће нам неки од наредних резултата бити потребни за дискретизацију дводимензионе Poisson-ове једначине, размотримо у овом поглављу и одговарајући проблем сопствених вредности или Sturm¹⁴-Liouville¹⁵-ов проблем. То је специјални случај проблема (2):

$$\begin{aligned} -u''(x) &= \lambda u(x), \quad x \in \Omega, \\ u(0) &= u(1) = 0, \end{aligned} \tag{11}$$

који има нетривијално решење само за неке вредности параметра λ . Вредност параметра λ за коју проблем (11) има нетривијално решење назива се *сопствена вредност*, а одговарајуће решење *сопствена функција* граничног проблема (11). Задатак (11) има пребројиво много сопствених вредности λ_k , које су све позитивне и $\lambda_k \rightarrow \infty$ кад $k \rightarrow \infty$. Одговарајуће сопствене функције u_k су узajамно ортогоналне у смислу стандардног скаларног производа у $L_2(0,1)$.

Диференцијска схема која апроксимира проблем (11) с грешком $O(h^2)$ има облик

¹⁴Jacques Charles-Francois Sturm (1803 - 1855), француски математичар.

¹⁵Joseph Liouville (1809 - 1882), француски математичар.

$$\begin{aligned} -v_{\bar{x},i} &= \lambda_h v_i, \quad i = 1, 2, \dots, n-1, \\ v_0 &= v_n = 0, \end{aligned} \tag{12}$$

где је са λ_h означена апроксимација сопствене вредности. Ова схема, уствари, представља хомогени систем линеарних једначина с тродијагоналном матрицом у којој фигурише параметар λ_h . Стога се гранични проблем (11), узимајући у обзир граничне услове, методом коначних разлика своди на проблем сопствених вредности

$$A\mathbf{v} = \lambda_h \mathbf{v},$$

$(n-1)$ -димензионе квадратне матрице

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & \\ 0 & & & -1 & 2 \end{pmatrix}.$$

Матрица A је симетрична и позитивно дефинисана, те су јој све сопствене вредности позитивне. Сопствене вредности представљају апроксимације првих $n-1$ сопствених функција проблема (11) у чворовима мреже $\bar{\omega}_h$.

Сопствене вредности и сопствене функције за проблем (11) су добро познате

$$\lambda = \lambda_k = k^2 \pi^2, \quad u(x) \equiv u_k(x) = \sin k\pi x, \quad k = 1, 2, 3, \dots$$

Непосредно се проверава да функције $u_k(x) = \sin k\pi x$ (тачније, њихове рестрикције на мрежу $\bar{\omega}_h$) задовољавају и диференцијску схему (12):

$$\begin{aligned} -(\sin k\pi x)_{\bar{x},i} &= \frac{1}{h^2} (-\sin k\pi(x_i + h) + 2\sin k\pi x_i - \sin k\pi(x_i - h)) \\ &= \frac{1}{h^2} (-2\sin k\pi x_i \cos k\pi h + 2\sin k\pi x_i) = \frac{4}{h^2} \sin^2 \frac{k\pi h}{2} \sin k\pi x_i. \end{aligned}$$

Приметимо такође да је

$$\sin(2n \mp k)\pi x_i = \mp \sin k\pi x_i.$$

На тај начин, индекси виших фреквенција (код којих је $k > n$), на мрежи се виде у фреквенцијском опсегу $k \leq n$. Специјално, $\sin n\pi x_i = 0$. Одатле закључујемо да су

$$v = v^k(x) = \sin k\pi x, \quad x = x_i \in \bar{\omega}_h, \quad k = 1, 2, \dots, n-1,$$

сопствене функције, а

$$\lambda_{h,k} = \frac{4}{h^2} \sin^2 \frac{k\pi h}{2}, \quad k = 1, 2, \dots, n-1,$$

сопствене вредности диференцијске схеме (12).

Због монотоности синусне функције на сегменту $[0, \pi/2]$ је

$$\frac{4}{h^2} \sin^2 \frac{\pi h}{2} = \lambda_{h,1} < \lambda_{h,2} < \dots < \lambda_{h,n-1} = \frac{4}{h^2} \cos^2 \frac{\pi h}{2}.$$

Одатле, користећи неједнакост

$$\sin x \geq \frac{2\sqrt{2}}{\pi} x, \quad x \in (0, \pi/4),$$

добијамо следећу оцену сопствених вредности (за $h \leq 1/2$, односно $n \geq 2$):

$$8 \leq \frac{4}{h^2} \sin^2 \frac{\pi h}{2} \leq \lambda_{h,k} \leq \frac{4}{h^2} \cos^2 \frac{\pi h}{2} < \frac{4}{h^2}. \quad (13)$$

Из асимптотског развоја синусне функције, за фиксирано k , следи

$$\lambda_{h,k} = \frac{4}{h^2} \left(\frac{k\pi h}{2} + O(k^3 h^3) \right)^2 = k^2 \pi^2 + O(k^4 h^2) = \lambda_k + O(k^4 h^2),$$

што значи да $\lambda_{h,k} \rightarrow \lambda_k$ када $h \rightarrow 0$. Ако k није ограничено кад $h \rightarrow 0$ претходни закључак не важи. На пример, за $k = n-1$ је

$$\lambda_{h,n-1} = \frac{4}{h^2} \sin^2 \frac{(n-1)\pi h}{2} = \frac{4}{h^2} \sin^2 \left(\frac{\pi}{2} - \frac{\pi h}{2} \right) = \frac{4}{h^2} \cos^2 \frac{\pi h}{2},$$

па је

$$\frac{\lambda_{h,n-1}}{\lambda_{n-1}} = \frac{\frac{4}{h^2} \cos^2 \frac{\pi h}{2}}{(n-1)^2 \pi^2} = \frac{4}{\pi^2} \frac{\cos^2 \frac{\pi h}{2}}{(1-h)^2} = \frac{4}{\pi^2} (1 + 2h + O(h^2)) \rightarrow \frac{4}{\pi^2},$$

када $h \rightarrow 0$.

Можемо закључити да континуални проблем (11) има пребројиво много сопствених вредности и сопствених функција. Његова дискретна апроксимација коначним разликама (12) има их коначно много, тачније $n - 1$, где је $n + 1$ број чворова мреже диференцијске схеме. Стога диференцијском схемом (12) одређујемо првих $n - 1$ сопствених вредности и сопствених вектора, при чему је апроксимација сопствене вредности утолико лошија уколико је k веће. Згушњавањем мреже, тј. смањивањем корака и повећањем броја чворова добијају се апроксимације већег броја сопствених вредности и већа тачност апроксимације оних са нижим индексом k .

2.2 Дводимензиона Poisson-ова једначина са хомогеним Dirichlet-овим граничним условом

Пређимо на дводимензиони задатак. Нека је $\Omega = (0, 1)^2$. Разматрамо проблем

$$\begin{aligned} -\Delta u &\equiv -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x, y), & (x, y)^T \in \Omega, \\ u(x, y) &= 0, & (x, y)^T \in \Gamma \equiv \partial\Omega. \end{aligned} \quad (14)$$

И овде ћемо користити методу коначних разлика, која се, као и у једнодимензионом случају, заснива на замени извода количницима коначних разлика. Нека је n природан број и

$$\bar{\omega}_h = \{x_i \mid x_i = ih, i = 0, 1, \dots, n, h = 1/n\}$$

равномерна мрежа на сегменту $[0, 1]$. Мрежу у области Ω дефинишимо као директан производ $\bar{\Omega}_h = \bar{\omega}_h \times \bar{\omega}_h$. Уведимо још и скупове унутрашњих $\Omega_h = \bar{\Omega}_h \cap \Omega$ и граничних $\Gamma_h = \bar{\Omega}_h \cap \Gamma = \bar{\Omega}_h \setminus \Omega_h$ чворова. Вредности функције v у чворовима мреже означаваћемо са $v_{ij} = v(x_i, x_j)$. Са $v_x, v_{\bar{x}}, v_y, v_{\bar{y}}$ означаваћемо количнике коначних разлика функције v по променљивим x и y :

$$v_{x,ij} = \frac{1}{h}(v_{i+1,j} - v_{ij}) = v_{\bar{x},i+1,j},$$

$$v_{y,ij} = \frac{1}{h}(v_{i,j+1} - v_{ij}) = v_{\bar{y},i,j+1}.$$

Гранични проблем (14) апроксимирајмо следећом диференцијском схемом:

$$\begin{aligned} -\Delta_h v &\equiv -(v_{\bar{x}\bar{x}} + v_{\bar{y}\bar{y}}) = f, & (x, y)^T &\in \Omega_h, \\ v &= 0, & (x, y)^T &\in \Gamma_h, \end{aligned} \quad (15)$$

односно, у развијеном облику

$$\begin{aligned} -\frac{v_{i+1,j} - 2v_{ij} + v_{i-1,j}}{h^2} - \frac{v_{i,j+1} - 2v_{ij} + v_{i,j-1}}{h^2} &= f_{ij}, \\ i, j &= 1, 2, \dots, n-1, \\ v_{i0} = v_{in} = 0, & \quad i = 0, 1, \dots, n, \\ v_{0j} = v_{nj} = 0, & \quad j = 1, 2, \dots, n-1, \end{aligned} \quad (16)$$

где је $f_{ij} = f(x_i, x_j)$, $i, j = 1, 2, \dots, n-1$.

Задатак (16) представља систем од $(n+1)^2$ линеарних алгебарских једначина с исто толико непознатих v_{ij} . (После елиминације непознатих које одговарају граничним чворовима и једнаке су нули, остаје $(n-1)^2$ једначина и исто толико непознатих). Општа једначина система (16) садржи пет непознатих које одговарају чворовима $(x_i, x_j)^T$, $(x_{i+1}, x_j)^T$, $(x_{i-1}, x_j)^T$, $(x_i, x_{j+1})^T$, $(x_i, x_{j-1})^T$ (Слика 4). Зато се ова диференцијска схема често назива и *схема "крст"*.

Пре него што докажемо егзистенцију решења диференцијске схеме (15), доказаћемо неколико помоћних резултата.

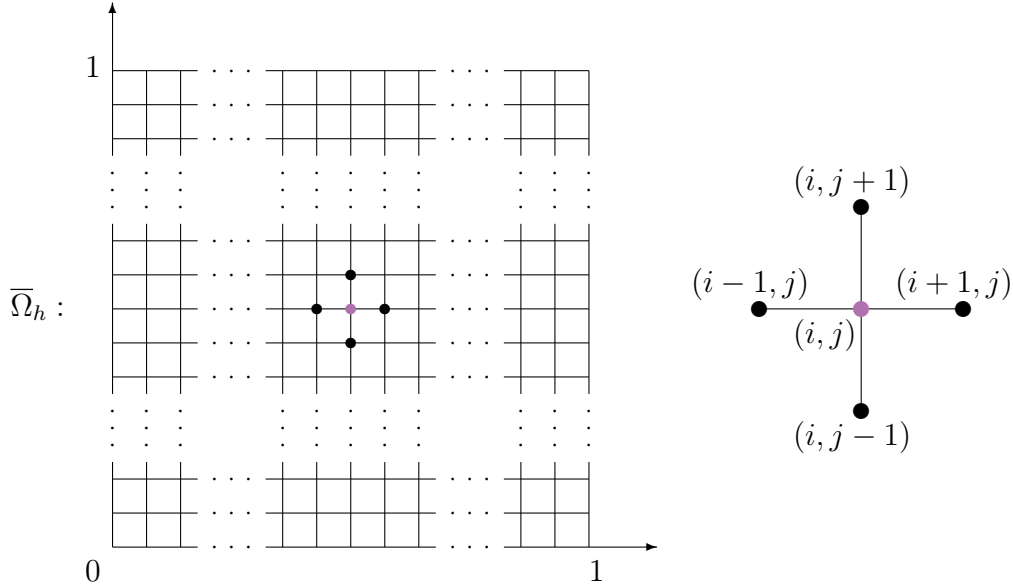
Нека је $V = \{v : \bar{\Omega}_h \rightarrow \mathbb{R} \mid v_{i0} = v_{in} = v_{0j} = v_{nj} = 0, i, j = 0, 1, \dots, n\}$ векторски простор функција дефинисаних на мрежи $\bar{\Omega}_h$. Простор V је хомоморфан са $\mathbb{R}^{(n-1)^2}$. Уведемо у V скаларни производ

$$(v, w)_h = h^2 \sum_{i,j=1}^{n-1} v_{ij} w_{ij}$$

и норме

$$\|v\|_h = (v, v)_h^{1/2} \quad \text{и} \quad \|v\|_{C,h} = \max_{1 \leq i,j \leq n-1} |v_{ij}|.$$

Нека је даље $\hat{\Delta}_h$ линеарни оператор који пресликава V у V , дефинисан са



Слика 4: Равномерна мрежа на сегменту $[0, 1]^2$.

$$\dot{\Delta}_h v = \begin{cases} \Delta_h v, & \text{на } \Omega_h, \\ 0, & \text{на } \Gamma_h. \end{cases}$$

Диференцијска схема (15) може се тада компактно записати на следећи начин:

$$-\dot{\Delta}_h v = \dot{f}, \quad v, \dot{f} \in V, \quad (17)$$

где је означено

$$\dot{f} = \begin{cases} f, & \text{на } \Omega_h, \\ 0, & \text{на } \Gamma_h. \end{cases}$$

Уведени оператор $\dot{\Delta}_h$ је самокоњугован:

$$\begin{aligned} (\dot{\Delta}_h v, w)_h &= h^2 \sum_{i,j=1}^{n-1} v_{x\bar{x},ij} w_{ij} + h^2 \sum_{i,j=1}^{n-1} v_{y\bar{y},ij} w_{ij} \\ &= -h^2 \sum_{i=0}^{n-1} \sum_{j=1}^{n-1} v_{x,ij} w_{x,ij} - h^2 \sum_{i=1}^{n-1} \sum_{j=0}^{n-1} v_{y,ij} w_{y,ij} = (v, \dot{\Delta}_h w)_h. \end{aligned}$$

Размотримо такође проблем сопствених вредности

$$-\dot{\Delta}_h v = \lambda_h v ,$$

односно, у развијеном облику

$$\begin{aligned} -\Delta_h v &= \lambda_h v , & (x, y)^T &\in \Omega_h , \\ v &= 0 , & (x, y)^T &\in \Gamma_h . \end{aligned}$$

Непосредно се проверава да су сопствене функције овог оператора

$$v = v^{kl} = \sin k\pi x \sin l\pi y , \quad (x, y)^T \in \bar{\Omega}_h , \quad k, l = 1, 2, \dots, n-1 ,$$

док су сопствене вредности

$$\lambda_h = \lambda_{h,kl} = \frac{4}{h^2} \left(\sin^2 \frac{k\pi h}{2} + \sin^2 \frac{l\pi h}{2} \right) .$$

Користећи оцену (13), добијамо

$$16 \leq \lambda_{h,kl} \leq \frac{8}{h^2} .$$

Према томе, сопствене вредности $\lambda_{h,kl}$ су строго позитивне, па је оператор $-\dot{\Delta}_h$ позитивно дефинисан. Зато једначина (17), а значи и диференцијска схема (15), има јединствено решење за свако $\dot{f} \in V$.

Испитајмо конвергенцију диференцијске схеме. Нека је u решење граничног проблема (14), v решење диференцијске схеме (15) и $z = u - v$. Функција z дефинисана је у чворовима мреже $\bar{\Omega}_h$ и задовољава услов

$$z(x, y) = 0 , \quad (x, y)^T \in \Gamma_h , \quad \text{односно} \quad z \in V .$$

Даље је

$$-\Delta_h z = -\Delta_h u + \Delta_h v = -\Delta_h u + \Delta u = \left(\frac{\partial^2 u}{\partial x^2} - u_{x\bar{x}} \right) + \left(\frac{\partial^2 u}{\partial y^2} - u_{y\bar{y}} \right) \equiv \psi .$$

Према томе, z је решење задатка

$$-\dot{\Delta}_h z = \dot{\psi} ,$$

где је

$$\dot{\psi} = \begin{cases} \psi, & \text{на } \Omega_h, \\ 0, & \text{на } \Gamma_h. \end{cases}$$

Ако $u \in C^4(\overline{\Omega})$ тада помоћу Тајлор-овог развоја добијамо $\psi = O(h^2)$.
Одатле непосредно следи

$$\left\| -\dot{\Delta}_h z \right\|_h = \left\| \dot{\psi} \right\|_h = O(h^2). \quad (18)$$

Користећи неједнакост (9) добијамо

$$z_{ij}^2 \leq \frac{1}{4} h \sum_{k=0}^{n-1} z_{x,kj}^2.$$

На исти начин је

$$z_{x,kj}^2 \leq \frac{1}{4} h \sum_{l=0}^{n-1} z_{xy,kl}^2,$$

што са претходним даје

$$z_{ij}^2 \leq \frac{1}{16} h^2 \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} z_{xy,kl}^2.$$

Непосредно се проверава идентитет

$$h^2 \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} z_{xy,kl}^2 = h^2 \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} z_{x\bar{x},kl} z_{y\bar{y},kl}.$$

Следи

$$\begin{aligned} z_{ij}^2 &\leq \frac{1}{16} h^2 \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} z_{x\bar{x},kl} z_{y\bar{y},kl} \\ &\leq \frac{1}{32} h^2 \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} (z_{x\bar{x},kl}^2 + 2z_{x\bar{x},kl} z_{y\bar{y},kl} + z_{y\bar{y},kl}^2) = \frac{1}{32} h^2 \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} (\Delta_h z)_{kl}^2, \end{aligned}$$

односно

$$\|z\|_{C,h} \leq \frac{1}{4\sqrt{2}} \left\| \dot{\Delta}_h z \right\|_h. \quad (19)$$

Из релација (18) и (19) непосредно добијамо тражену оцену брзине конвергенције диференцијске схеме (15):

$$\|z\|_{C,h} = O(h^2) .$$

Напоменимо да оцена (19) представља дводимензиони аналогон оцене (10). У дводимензионом случају не важи оцена аналогна (9). На исти начин се изводи континуални налогон оцене (19)

$$\|u\|_{C(\bar{\Omega})} \leq \frac{1}{4\sqrt{2}} \|\Delta u\|_{L_2(\Omega)} .$$

3 Итеративне методе

У претходном одељку смо проблем решавања диференцијалне једначине са граничним условом свели на проблем решавања система линеарних једначина. Нумеричке методе за решавање система линеарних једначина могу се поделити на две основне групе - директне и итеративне. Директним методама одређује се тачно решење с коначно много рачунских операција, под претпоставком да су сви параметри дати тачно и да се све рачунске операције реализују тачно. (Ако су коефицијенти система приближни бројеви или ако се при рачунању врше заокруживања, уместо тачног решења добиће се приближно). Итеративним методама решење је одређено граничном вредношћу низа узастопних апроксимација, које се рачунају неким једнообразном алгоритмом.

Теоријски, директне методе дају тачно решење система помоћу коначно много аритметичких операција, док итеративне методе то не могу. У пракси је, међутим, ситуација другачија: осим у тривијалним случајевима, када се не врши заокруживање, директне методе такође дају приближно решење. Зато за критеријум упоређивања ефикасности метода можемо изабрати број аритметичких операција потребних за добијање (приближног) решења са задатом тачношћу. У том случају има смисла упоређивати ефикасност итеративних са ефикасношћу директних метода.

3.1 Норме вектора и матрица

Једна фамилија норми у векторском простору \mathbb{C}^n дата је изразом

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}.$$

Специјални случајеви претходног израза су:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$$

- апсолутна норма, дефинисана за $p = 1$,

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}$$

- еуклидска¹⁶ норма, дефинисана за $p = 2$,

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

- униформна норма, дефинисана за $p \rightarrow \infty$.

Матрица $A^* = (a_{ij}^*)$ је конјугована матрици $A = (a_{ij})$ ако је $a_{ij}^* = \bar{a}_{ij}$. У векторски простор \mathbb{C}^n скаларни производ се може увести на следећи начин:

$$(\mathbf{x}, \mathbf{y}) = \mathbf{y}^* \mathbf{x} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^n x_i \bar{y}_i.$$

За норму матрице $\|A\|$ се каже да је индукована нормом вектора $\|\mathbf{x}\|$ ако је

$$\|A\| = \sup_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}.$$

¹⁶Еуклид (око 330 п.н.е. - око 275 п.н.е.), грчки математичар.

Норме матрица индуковане, редом, апсолутном, униформном и еуклидском нормом вектора су:

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| ,$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| ,$$

$$\|A\|_2 = \sqrt{\max_{1 \leq i \leq n} \lambda_i(A^*A)} .$$

Овде је $\lambda_i(A^*A)$ i -та сопствена вредност матрице A^*A . Норме матрице $\|A\|$ и вектора $\|\mathbf{x}\|$ су сагласне, ако је

$$\|A\mathbf{x}\| \leq \|A\| \|\mathbf{x}\|$$

за сваку квадратну матрицу A и вектор \mathbf{x} димензије n .

На основу дефиниције сагласних норми матрице и вектора и дефиниције сопствених вредности матрице имамо да је

$$|\lambda| \|\mathbf{x}\| = \|\lambda\mathbf{x}\| = \|A\mathbf{x}\| \leq \|A\| \|\mathbf{x}\| ,$$

одакле следи

$$|\lambda| \leq \|A\| . \tag{20}$$

3.2 Стандардне итеративне методе

Multigrid је настао са циљем да се уклоне недостаци стандардних итеративних метода, па зато у овом поглављу разматрамо те методе, које су важан део multigrid алгорита.

Решавамо систем линеарних једначина

$$A\mathbf{x} = \mathbf{b}, \quad (21)$$

где је A регуларна квадратна матрица димензије $n \times n$ и \mathbf{b} n -димензиони вектор,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

Код стандардних итеративних метода се, полазећи од произвољно изабраног вектора \mathbf{x}_0 , рекурентном формулом

$$\mathbf{x}_{k+1} = G_k(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k), \quad k = 0, 1, 2, \dots \quad (22)$$

одређује низ вектора, који под одређеним условима конвергира ка решењу система (21). Ако G_k зависи од $\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-l+1}$, а не и од $\mathbf{x}_{k-l}, \mathbf{x}_{k-l-1}, \dots, \mathbf{x}_0$, каже се да је итеративна метода дефинисана формулом (22) $(l + 1)$ -слојна. Ако G_k не зависи од k , метода је стационарна, а ако је G_k линеарна функција својих аргумената, метода је линеарна.

Двослојне стационарне линеарне итеративне методе за решавање система (21) могу се представити у облику

$$\mathbf{x}_{k+1} = B\mathbf{x}_k + \mathbf{c}, \quad k = 0, 1, 2, \dots \quad (23)$$

Нека систем (21) има јединствено решење и нека низ вектора одређен формулом (23) конвергира ка њему. Тада је $\mathbf{x} = A^{-1}\mathbf{b}$ решење система

$$\mathbf{x} = B\mathbf{x} + \mathbf{c}, \quad (24)$$

одакле добијамо следећу везу између A , B , \mathbf{b} и \mathbf{c} :

$$\mathbf{c} = (I - B)A^{-1}\mathbf{b}.$$

Теорема 1. *Нека систем (24) има јединствено решење. Тада итеративни процес (23) конвергира ка том решењу ако и само ако су све сопствене вредности матрице B по модулу мање од један.*

Доказ. Услов је потребан:

Претпоставимо да услов теореме није испуњен, тј. нека постоји сопствена вредност λ_0 матрице B таква да је $|\lambda_0| \geq 1$. Нека је

\mathbf{v}_0 одговарајући сопствени вектор, а \mathbf{x} решење система (24). За почетни вектор низа (23) узмимо

$$\mathbf{x}_0 = \mathbf{x} + \mathbf{v}_0 .$$

Тада је

$$\begin{aligned} \mathbf{x}_k - \mathbf{x} &= B\mathbf{x}_{k-1} - B\mathbf{x} = B(\mathbf{x}_{k-1} - \mathbf{x}) = \dots = B^k(\mathbf{x}_0 - \mathbf{x}) \\ &= B^k \mathbf{v}_0 = \lambda_0^k \mathbf{v}_0 \not\rightarrow 0 , \end{aligned}$$

одакле следи да низ \mathbf{x}_k не конвергира ка \mathbf{x} .

Услов је довољан:

Нека је услов теореме испуњен, тј. нека су све сопствене вредности λ_i матрице B по модулу мање од један. Пошто је

$$\mathbf{x}_k - \mathbf{x} = B^k(\mathbf{x}_0 - \mathbf{x}) ,$$

довољно је показати да $B^k \rightarrow O$ кад $k \rightarrow \infty$.

Нека је

$$J = \begin{pmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_m \end{pmatrix} ,$$

Jordan¹⁷-ова нормална форма матрице B , и

$$J_i = \begin{pmatrix} \lambda_i & 1 & \dots & 0 \\ & \lambda_i & 1 & \\ & & \lambda_i & \ddots \\ 0 & & & \ddots & \\ & & & & \lambda_i \end{pmatrix}$$

квадратна матрица реда $n_i \geq 1$ ($n_1 + n_2 + \dots + n_m = n$). Следи

$$B = C^{-1} J C ,$$

где је C регуларна матрица, и

$$B^k = C^{-1} J^k C .$$

Даље је

¹⁷Marie Ennemond Camille Jordan (1838 - 1922), француски математичар.

$$J^k = \begin{pmatrix} J_1^k & & 0 \\ & \ddots & \\ 0 & & J_m^k \end{pmatrix}.$$

Индукцијом се показује да је за $k > n_i$

$$J_i^k = \begin{pmatrix} \lambda_i^k & \binom{k}{1} \lambda_i^{k-1} & \cdots & \binom{k}{n_i-1} \lambda_i^{k-n_i+1} \\ 0 & \lambda_i^k & \cdots & \binom{k}{n_i-2} \lambda_i^{k-n_i+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_i^k \end{pmatrix}$$

Пошто је $|\lambda_i| < 1$ следи: $J_i^k \rightarrow O$, $J^k \rightarrow O$ и $B^k \rightarrow O$ кад $k \rightarrow \infty$. □

Ако је испуњен услов теореме, тада систем (24) има јединствено решење (обрнуто не важи). Заиста, сопствене вредности матрице система $I - B$ једнаке су $1 - \lambda_i$ и због постављеног услова су различите од нуле, што значи да је матрица $I - B$ регуларна.

Величина $\max_i |\lambda_i|$ назива се спектрални радијус матрице B , те итеративна метода конвергира ако је спектрални радијус матрице B мањи од један.

Доказана теорема даје потребан и довољан услов конвергенције итеративног процеса (23). Али, њена практична вредност није велика, јер је одређивање сопствених вредности матрице тежи задатак од решавања система линеарних једначина. Међутим, ако је нека норма матрице B , сагласна с неком векторском нормом, мања од један, тада су према неједнакости (20) и све сопствене вредности матрице B по модулу мање од један, те је услов теореме аутоматски испуњен.

Представимо матрицу A линеарног система у облику суме

$$A = L + D + U,$$

где је L доња, а U горња троугаона матрица с нулама на дијагонали, док је D дијагонална матрица.

Размотримо сада неке од стандардних итеративних метода.

3.2.1 Јасоби-јева итеративна метода

Јасоби¹⁸-јева итеративна метода дефинисана је формулом

$$D\mathbf{x}_{k+1} + (L + U)\mathbf{x}_k = \mathbf{b} ,$$

која се своди на канонски облик (23) за

$$B = -D^{-1}(L + U) \quad \text{и} \quad \mathbf{c} = D^{-1}\mathbf{b} .$$

На основу теореме 1, Јасоби-јева метода конвергира ако и само ако су сви корени једначине

$$\det [-D^{-1}(L + U) - \lambda I] = 0 ,$$

односно

$$\det (L + U + \lambda D) = 0 ,$$

по модулу мањи од један.

Из услова

$$\|B\|_{\infty} < 1$$

добивамо следећи довољан услов конвергенције Јасоби-јеве методе:

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}| , \quad i = 1, 2, \dots, n , \quad (25)$$

где су a_{ij} елементи матрице A .

Матрице B и $\hat{B} = -(L+U)D^{-1}$ су сличне, те имају једнаке сопствене вредности. Зато је услов

$$\|\hat{B}\|_1 < 1$$

такође довољан за конвергенцију Јасоби-јеве методе. Одатле непосредно добијамо:

$$\sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| < |a_{jj}| , \quad j = 1, 2, \dots, n . \quad (26)$$

¹⁸Carl Gustav Jacob Jacobi (1804 - 1851), немачки математичар.

Приметимо да се код Јасоби-јеве методе компоненте вектора \mathbf{x}_{k+1} не користе чим се нека од њих израчуна, већ тек кад су све израчунате, тј. након завршетка итерације. При том се до краја итерације користе све компоненте вектора \mathbf{x}_k . То при имплементацији захтева одвајање меморије и за све компоненте вектора \mathbf{x}_k и за све компоненте вектора \mathbf{x}_{k+1} . Следећа метода решава поменути проблем заузимања меморије.

3.2.2 Gauss-Seidel-ова итеративна метода

Gauss¹⁹-Seidel²⁰-ова итеративна метода је дата формулом

$$(L + D)\mathbf{x}_{k+1} + U\mathbf{x}_k = \mathbf{b} ,$$

што значи да је

$$B = -(L + D)^{-1}U \quad \text{и} \quad \mathbf{c} = (L + D)^{-1}\mathbf{b} .$$

На основу теореме 1, Gauss-Seidel-ова метода конвергира ако и само ако су корени једначине

$$\det [\lambda(L + D) + U] = 0$$

по модулу мањи од један. Услови (25) и (26) су довољни за конвергенцију и ове методе.

¹⁹Johann Carl Friedrich Gauss (1777 - 1855), немачки математичар и физичар.

²⁰Philipp Ludwig von Seidel (1821 - 1896), немачки математичар.

3.3 Ограничења стандардних итеративних метода

3.3.1 Локалност и брзина преноса информација

Илуструјмо овај проблем на примеру дводимензионе Poisson-ове једначине. За корак мреже $h = 1/n$, једначине одговарајућег линеарног система добијеног дискретизацијом имају облик

$$4v_{ij} - v_{i+1,j} - v_{i-1,j} - v_{i,j+1} - v_{i,j-1} = h^2 f_{ij}, \quad i, j = 1, 2, \dots, n-1.$$

Свака једначина је локалног карактера и повезује решење и десну страну у тачки (i, j) са решењем у најближим суседним тачкама мреже. То важи без обзира на редослед једначина у систему, тј. независно је од нумерације чворова. Један од проблема код стандардних итеративних метода је што матрица итерације B одражава исту локалност.

На пример, у Јасоби-јевој методи, итерације имају облик

$$x_{ij}^{(k+1)} = \frac{1}{4} \left(x_{i+1,j}^{(k)} + x_{i-1,j}^{(k)} + x_{i,j+1}^{(k)} + x_{i,j-1}^{(k)} + h^2 f_{ij} \right), \\ i, j = 1, 2, \dots, n-1, \quad k \geq 0.$$

Уз ознаку за десну страну $b_{ij} := h^2 f_{ij}$, очигледно је да се нова апроксимација добија локално, усредњавањем вредности из претходне апроксимације $x^{(k)}$ и десне стране, преко најближих суседних чворова мреже, укључујучи и дати чвор.

Претпоставимо да десна страна \mathbf{b} има један једини елемент различит од нуле (нпр. једнак један) и то у центру квадрата - у тачки $(i, j) = (n/2, n/2)$. Број непознатих је $N = (n-1)^2$. Егзактно решење v_{ij} је различито од нуле у свим унутрашњим чворовима мреже и опада према рубовима квадрата (хомогени гранични услови).

За почетну итерацију узмимо $x^{(0)} = 0$. Погледајмо шта се дешава у Јасоби-јевој методи након m итерација. Занима нас у којим чворовима је $x^{(m)}$ различито од нуле и колика је грешка. У m -тој итерацији, због усредњавања преко најближих суседа, само вредности удаљене за највише m од центра могу бити различите од

нуле. До осталих чворова информација из центра још није стигла, јер се шири брзином од једног чвора по итерацији.

Уз такво ширење информације, најбоља могућа апроксимација решења након m итерација је рестрикција тачног решења на одговарајућу околину центра, јер апроксимација мора бити нула у свим тачкама удаљеним за више од m од центра. Грешка те апроксимације једнака је правом решењу у чвору удаљеном за $m+1$ од центра. Како опада та грешка са порастом m ? Потребно је бар $O(\log N)$ итерација, односно $O(N \log N)$ операција да грешка падне за константни фактор мањи од један.

Исто понашање важи и за остале стандардне итеративне методе. Дакле, ниједна итеративна метода базирана на локалном усредњавању не може бити оптимална. Информације кроз мрежу чворова треба преносити брже од једне тачке по итерацији, што је основна мотивација за multigrid методу.

3.3.2 Сопствене вредности и редуковање грешке

Вратимо се на мотивациони пример. Напоменимо да је итеративна метода без релаксационог параметра која се тамо користи уствари Јасоби-јева итеративна метода. Методу коју добијамо увођењем релаксационог параметра називаћемо (Јасоби-јева) итеративна метода са релаксационим параметром.

Уочимо прво да уместо вектора грешке можемо посматрати његову компоненту која је паралелна (или ону која је управна) на линију дуж које се војници распоређују. Вектор који садржи грешке у почетним позицијама означимо са $e^{(0)} = (e_1^{(0)}, e_2^{(0)}, \dots, e_{n-1}^{(0)})^T$. (Имајмо у виду да за војнике 0 и n важи $e_0^{(0)} = e_n^{(0)} = 0$). За $k = 1, 2, 3, \dots$, нова грешка у позицији i -тог војника након k итерација је дата као средња вредност грешака његових првих суседа непосредно пре k -те итерације, односно:

$$e_i^{(k)} = \frac{1}{2} \left(e_{i+1}^{(k-1)} + e_{i-1}^{(k-1)} \right).$$

То можемо записати у матричном облику

$$e^k = B e^{k-1}, \quad (27)$$

где је B тродијагонална матрица са $1/2$ изнад и испод главне дијагонале и нулама на главној дијагонали,

$$B = \begin{pmatrix} 0 & 1/2 & & & 0 \\ 1/2 & 0 & 1/2 & & \\ & 1/2 & \ddots & \ddots & \\ & & \ddots & \ddots & 1/2 \\ 0 & & & 1/2 & 0 \end{pmatrix} = 1/2 \begin{pmatrix} 0 & 1 & & & 0 \\ 1 & 0 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & 0 \end{pmatrix}.$$

Приметимо да је записом (27) дат специјалан случај Јасови-јеве итеративне методе.

Понашање итеративне методе и његова повезаност са величином скале може се довести у везу са грешком \mathbf{e} , која је сопствени вектор матрице B , што значи да задовољава $B\mathbf{e} = \lambda\mathbf{e}$, за неку сопствену вредност λ . Матрица B има $n - 1$ сопствених вредности, које су дате са

$$\lambda^{(k)} = \cos \frac{k\pi}{n}, \quad k = 1, 2, \dots, n - 1,$$

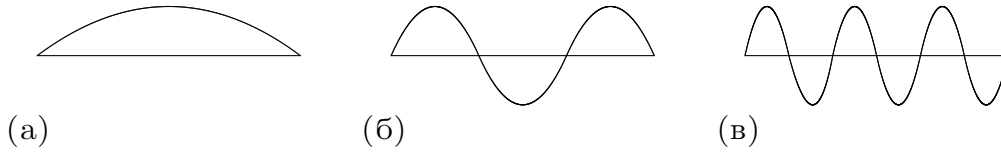
док је $n - 1$ одговарајућих сопствених вектора дато са

$$\mathbf{w}^{(k)} = \begin{pmatrix} \sin \frac{k\pi}{n} \\ \sin \frac{2k\pi}{n} \\ \vdots \\ \sin \frac{(n-1)k\pi}{n} \end{pmatrix}, \quad k = 1, 2, \dots, n - 1. \quad (28)$$

(Како бисмо проверили да важи $B\mathbf{w}^{(k)} = \lambda^{(k)}\mathbf{w}^{(k)}$ користимо тригонометријски идентитет $\sin \frac{k(i+1)\pi}{n} + \sin \frac{k(i-1)\pi}{n} = 2 \cos \frac{k\pi}{n} \sin \frac{ki\pi}{n}$). Слика 5 приказује $\mathbf{w}^{(1)}$, $\mathbf{w}^{(3)}$ и $\mathbf{w}^{(6)}$. Величину скале придружујемо дужини таласа, која је обрнуто пропорционална k , тј. мале вредности k одговарају сопственим векторима велике скале, и обрнуто.

Сопствени вектори $\mathbf{w}^{(k)}$, $k = 1, 2, \dots, n - 1$, су линеарно независни и чине базу простора \mathbb{R}^{n-1} , тако да грешку у почетним позицијама можемо написати као линеарну комбинацију сопствених вектора,

$$\mathbf{e}^{(0)} = \sum_{k=1}^{n-1} c_k \mathbf{w}^{(k)},$$



Слика 5: Сопствени вектори: (а) $\mathbf{w}^{(1)}$, (б) $\mathbf{w}^{(3)}$, (в) $\mathbf{w}^{(6)}$.

где $c_k \in \mathbb{R}$, $k = 1, 2, \dots, n-1$. Грешка у позицијама војника након m итерација може се изразити са

$$\begin{aligned} \mathbf{e}^{(m)} &= B^m \mathbf{e}^{(0)} = B^m \sum_{k=1}^{n-1} c_k \mathbf{w}^{(k)} = \sum_{k=1}^{n-1} c_k B^m \mathbf{w}^{(k)} = \sum_{k=1}^{n-1} c_k (\lambda^{(k)})^m \mathbf{w}^{(k)} \\ &= \sum_{k=1}^{n-1} c_k \left(\cos \frac{k\pi}{n} \right)^m \mathbf{w}^{(k)}. \end{aligned}$$

Како је $|\lambda^{(k)}| = \left| \cos \frac{k\pi}{n} \right| < 1$ за $k = 1, 2, \dots, n-1$, закључујемо да итеративна метода конвергира, јер $|\lambda^{(k)}|^m \rightarrow 0$ када $m \rightarrow \infty$. Испитајмо брзину конвергенције за различите вредности k .

Када је $k/n \ll 1$, тада је $\lambda^{(k)} = \cos \frac{k\pi}{n} \approx 1$, па се коефицијенти уз сопствене векторе који одговарају малим k незнатно мењају у свакој итерацији (слика 1в). За $k = n/2$ је $\lambda^{(k)} = \cos \frac{k\pi}{n} = 0$ и процес конвергира већ након прве итерације (слика 1г). Како се k приближава n , $\cos \frac{k\pi}{n}$ се приближава -1 , па коефицијенти уз сопствене векторе који одговарају оваквим k скоро да само мењају знак при свакој итерацији (слика 1д). Како би се решио проблем из последње ситуације уводи се релаксациони параметар. Ако је тај параметар $2/3$, имамо

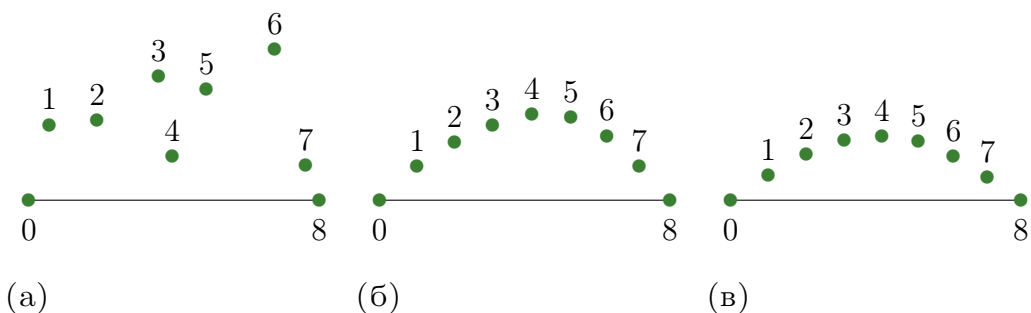
$$e_i^{(m)} = \frac{1}{3} e_i^{(m-1)} + \frac{2}{3} \left[\frac{1}{2} \left(e_{i+1}^{(m-1)} + e_{i-1}^{(m-1)} \right) \right] \quad (29)$$

и матрични запис

$$\mathbf{e}^{(m)} = B_r \mathbf{e}^{(m-1)} = \left(\frac{1}{3} I + \frac{2}{3} B \right) \mathbf{e}^{(m-1)}, \quad (30)$$

где је B_r матрица итеративне методе са релаксационим параметром, а I јединична матрица. Сопствени вектори матрице B_r су исти као и код матрице B , док су сопствене вредности дате са

$$\lambda_r^{(k)} = \frac{1}{3} + \frac{2}{3} \lambda^{(k)} = \frac{1}{3} + \frac{2}{3} \cos \frac{k\pi}{n}. \quad (31)$$



Слика 6: Редуковање грешке код итеративне методе са релаксационим параметром: (а) почетна позиција, (б) након 4 итерације, (в) након 8 итерација.

Зашто итеративна метода са релаксационим параметром ефикасно редукује грешке мале скале? За $1 \leq k < n/2$ одговарајуће сопствене векторе називамо *глатким* (или векторима велике скале), а за $n/2 \leq k \leq n$ одговарајуће сопствене векторе називамо *грубим* (или векторима мале скале). Приметимо да, када се k креће од $n/2$ до $n - 1$, $\cos \frac{k\pi}{n}$ се креће од 0 до (приближно) -1 , па се $\lambda_r^{(k)}$ креће од $1/3$ до (приближно) $-1/3$. Чињеница да је $|\lambda_r^{(k)}| \leq 1/3$ за све грубе сопствене векторе повлачи да се сваки коефицијент уз груби сопствени вектор у свакој итерацији редукује до највише $1/3$ своје вредности пре итерације, независно од n . Тада кажемо да је *релаксациони фактор* итерације $1/3$. Ова итеративна метода је ефикасна у редуковању грешке, али након њеног редуковања постаје неефикасна за даљу употребу, што је илустровано сликом 6.

4 Multigrid метода

4.1 Основне идеје multigrid-a

Multigrid је рекурзивни алгоритам базиран на *divide et impera*²¹ приступу, који користи више различитих мрежа за генерисање апроксимације решења.

Multigrid спада у итеративне методе, али се итерације одвијају на разним мрежама - од грубљих, према све финијим (а, делом, и обрнуто). Све итерације су не на само једној мрежи, већ на више њих, па отуда и назив ове методе.

Стандардне итеративне методе, које су локалног карактера, користе се само за (итеративно) побољшање резултата на свакој појединој мрежи.

Код multigrid-a локалност нема лоших последица, јер се почетно решење на свакој мрежи добија глобално, на бази *divide et impera* - из двоструко грубље мреже. Управо то обезбеђује брже ширење информација кроз мрежу (или мреже) чворова.

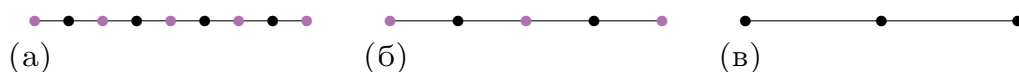
Divide et impera приступ се у multigrid-у користи на два повезана начина:

- по густини мреже (просторни домен),
- по индексу сопствених вредности (фреквенцијски домен).

За илустрацију *divide et impera* приступа користимо једнодимензиону (дводимензиону) Poisson-ову једначину. Како изгледа просторна рекурзија по густини мреже?

Почетно решење на n ($n \times n$) мрежи добија се коришћењем грубље $n/2$ ($(n/2) \times (n/2)$) мреже као апроксимације, узимањем сваке друге

²¹Термин преузет из латинског језика, а значи “подели, па владај”.



Слика 7: Divide et impera у просторном домену - једнодимензиони случај: (а) 9 чворова, 7 непознатих. Љубичастом бојом су означени чворови који припадају следећој, грубљој мрежи; (б) 5 чворова, 3 непознате. Љубичастом бојом су означени чворови који припадају следећој, грубљој мрежи; (в) 3 чвора, 1 непозната.

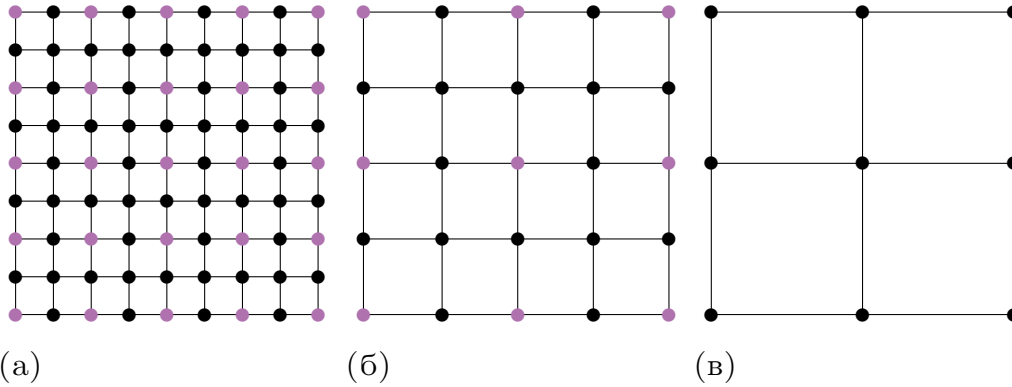
тачке из n ($n \times n$) мреже, половљењем у свакој димензији. Ова грубља $n/2$ ($(n/2) \times (n/2)$) мрежа опет се апроксимира још грубљом $n/4$ ($(n/4) \times (n/4)$) мрежом и тако редом, рекурзивно (Слике 7 и 8). Циљ је очување глобалности у свим фазама решавања проблема, тј. брже ширење информација.

Размотримо divide et impera у фреквенцијском домену. Посао који радимо на одређеној мрежи има за циљ да смањи (или елиминисхе) грешку у оној половини фреквенцијских компоненти чија грешка још није смањена (или елиминисана) на грубљим мрежама. То значи да треба редуковати грешку у горњој или вишој половини фреквенција, јер се оне појављују тек на финијој мрежи.

Реализација овог побољшања решења на појединој мрежи је усредњавање решења у свакој тачки мреже, с обзиром на њене суседе, нпр. варијацијом Јасоби-јеве итеративне методе.

4.2 Multiscale приступ

Проширимо multiscale приступ, који смо примењивали у мотивационом примеру, на проблем једнодимензионе Poisson-ове једначине са хомогеним Dirichlet-овим граничним условом. Главни циљ је показати да, када знамо шта да радимо на великим скалама, прво морамо пренети адекватну информацију са малих на велике скале. Уочимо сличност између ова два проблема: ако ставимо $f = 0$,



Слика 8: Divide et impera у просторном домену - дводимензиони случај: (а) 9×9 чворова, 7×7 непознатих. Љубичастом бојом су означени чворови који припадају следећој, грубљој мрежи; (б) 5×5 чворова, 3×3 непознате. Љубичастом бојом су означени чворови који припадају следећој, грубљој мрежи; (в) 3×3 чвора, 1×1 непозната.

очигледно је да је решење u права линија која повезује тачке $(0, 0)$ и $(1, 0)$. Тада је дискретно решење скуп вредности дуж линије, скака једнака средини својих првих суседа.

Multiscale приступ можемо прилагодити решавању једнодимензионе Poisson-ове једначине са хомогеним Dirichlet-овим граничним условом. Прво претпоставимо да је негативан предзнак оператора Δ утопљен у десну страну једначине, тј. да је полазна једначина $\Delta u(x) = f(x)$, а затим искористимо добијене резултате, водећи рачуна о промени знака оператора Δ . Нека је $v^{(0)}$ почетна апроксимација тачног решења v која задовољава гранични услов. У m -тој итерацији рачунамо $v_i^{(m)}$, $i = 1, 2, \dots, n - 1$, које задовољава i -ту једначину система (5):

$$v_i^{(m)} = \frac{1}{2} \left(v_{i+1}^{(m-1)} + v_{i-1}^{(m-1)} - h^2 f_i \right), \quad i = 1, 2, \dots, n - 1. \quad (32)$$

Понашање конвергенције је идентично као и у мотивационом примеру. Означимо грешку у m -тој итерацији са

$$\mathbf{e}_h^{(m)} = v - v^{(m)}, \quad (33)$$

а $e_{h,0}^{(m)} = e_{h,n}^{(m)} = 0$, јер све апроксимације задовољавају граничне услове. Једначина (32) произилази из једначине

$$v_i = \frac{1}{2} \left(v_{i+1} + v_{i-1} - h^2 f_i \right),$$

коју добијамо издвојивши v_i у једначини (5). Одавде следи

$$e_{h,i}^{(m)} = \frac{1}{2} \left(e_{h,i+1}^{(m-1)} + e_{h,i-1}^{(m-1)} \right) ,$$

што је исто као и у случају распоређивања војника. Слично, итеративна метода са релаксационим параметром $2/3$ је дефинисана са

$$v_i^{(m)} = \frac{1}{3} v_i^{(m-1)} + \frac{2}{3} \left(v_{i+1}^{(m-1)} + v_{i-1}^{(m-1)} - h^2 f_i \right) ,$$

а важе и једначине попут (29), (30) и (31).

Ради једноставности и даље претпостављамо да је n степен двојке. Слично као и у мотивационом примеру, почињемо применом итеративне методе на проблем дефинисан на највећој скали, који се састоји од променљивих v_0 , $v_{n/2}$ и v_n . Корак мреже на овој скали је $1/2$, а v_0 и v_n су одређени граничним условом. Помоћу Јасоби-јеве итеративне методе $v_{n/2}$ рачунамо као

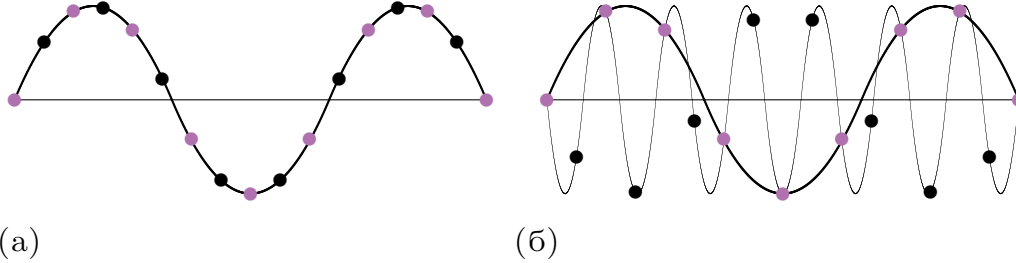
$$v_{n/2} = \frac{1}{2} \left(v_0 + v_n - \frac{1}{4} f_{n/2} \right) .$$

Даље, на основу вредности највеће скале, примењујемо Јасоби-јеву итеративну на другу по величини скалу, укључујући $v_{n/4}$ и $v_{3n/4}$. Настављамо док не одредимо све променљиве, као и у мотивационом примеру.

Овај алгоритам је рачунски очигледно једноставан. Нажалост, не даје коректно решење (осим ако је f тривијално), јер не решава праву једначину на грубљим мрежама. Тачније, десне стране једначине нису довољно прецизне.

4.3 Преклапање сопствених вектора

Слика 9а приказује глатки сопствени вектор ($k = 3$) на мрежи са $n = 16$ чворова, као и на двоструко грубљој мрежи, са $n =$



Слика 9: Преклапање: Љубичасти кружићи одговарају грубљој мрежи (а) $\mathbf{w}^{(3)}$ и (б) $-\mathbf{w}^{(13)}$.

8 чворова, која се састоји од равномерно распоређених чворова fine мреже. Слична је ситуација и на слици 9б, само за грубљи сопствени вектор ($k = 13$), с коефицијентом -1. Очигледно је да на грубој мрежи долази до преклапања. На њој је могуће приказати отприлике половину сопствених вектора од оних приказаних на fine мрежи, тако да се неки парови вектора fine мреже морају преклопити. Специјално, у једнакости (28) комплементни сопствени вектори $\mathbf{w}^{(k)}$ и $\mathbf{w}^{(n-k)}$ се разликује само по знаку. Ако f_h запишемо као линеарну комбинацију сопствених вектора, појавиће се груби сопствени вектори са ненула коефицијентима - то неће бити коректно приказано на грубој мрежи. У специјалном случају, ако је f_h линеарна комбинација $\mathbf{w}^{(k)}$ и $\mathbf{w}^{(n-k)}$ за неко k , где је $\mathbf{w}^{(k)} = -\mathbf{w}^{(n-k)}$, оно ће нестати када се прикаже на грубљој мрежи.

Увођење релаксационог параметра, базирано на подели спектра на гладак и груб део, наговештавало је употребу двоструко грубље мреже. Глатки сопствени вектори су ти који се могу приказати тачно на два пута грубљој мрежи, док се груби сопствени вектори поклапају са глатким.

Можемо закључити да конструкција коректне једначине за грубе мреже захтева информације са fine мреже. У случају једнодимензионог моделног проблема можемо превазићи проблем преклапања sukcesивним локалним усредњавањем. Међутим, у општем случају, проблем не можемо решити једноставним рачуном, што указује да multiscale приступ није довољан. Како су грубе мреже корисне само за приказивање глатких сопствених вектора, морамо водити рачуна при избору променљивих грубе мреже.

4.4 Грешка дискретизације и алгебарска грешка. Резидуална једначина

Важно је направити разлику између два типа грешке. Прва од њих је *грешка дискретизације*, дата у чвору мреже i са $u(x_i) - v_i$, $i = 1, 2, \dots, n - 1$, која представља разлику између решења диференцијалне једначине и (тачног) решења система добијеног дискретизацијом (на одговарајућој мрежи); одређена је проблемом, дискретизацијом и величином мреже. Итеративна метода нема утицаја на њу, јер утиче само на апроксимацију v . Други тип је *алгебарска грешка*, дата разликом између решења система добијеног дискретизацијом и тренутне апроксимације тог решења (једнакост (33)), која се редукује помоћу итеративне методе. Наша идеја је да прво применимо Јасови-јеву итеративну методу са релаксационим параметром, која редукује алгебарску грешку, а онда да конструишемо линеарни систем за вектор грешке и апроксимирамо га на грубљој мрежи. Ово може рекурзивно да се понавља, све док на грубљој мрежи не решимо приближну једначину за алгебарску грешку следеће - финаје мреже.

Мотивација за употребу алгебарске грешке као променљиве грубе мреже је двострука. Пошто нам преклапање сопствених вектора ограничава могућност рачунања прецизне апроксимације v на грубој мрежи, морамо потражити итеративно побољшање. Боље је користити грубу мрежу да коригујемо алгебарску грешку fine мреже, него тражити решење на финој мрежи. Сем тога, итеративна метода редукује алгебарску грешку и можемо је довољно прецизно одредити на грубој мрежи.

Означимо са \tilde{v} тренутну апроксимацију v , добијену након неколико итерација, и нека је $e_h = v - \tilde{v}$ ознака за одговарајућу алгебарску грешку, за коју знамо да је глатка (слика 6). Из (6) и тривијалне релације

$$(L_h \tilde{v})_i = (L_h \tilde{v})_i, \quad i = 1, 2, \dots, n - 1,$$

добијамо систем за алгебарску грешку:

$$(L_h e_h)_i = f_{h,i} - (L_h \tilde{v})_i \equiv r_{h,i}, \quad i = 1, 2, \dots, n - 1,$$

$$e_{h,0} = e_{h,n} = 0,$$

где је $r_{h,i}$ i -ти резидуал. Претходни систем можемо записати као

$$r_h = f_h - L_h \tilde{v}.$$

Због јединствености решења важи:

$$r_h = 0 \iff e_h = 0.$$

Узевши у обзир да је $L_h v = f_h$ и искористивши дефиниције алгебарске грешке и резидуала добијамо изузетно важну везу између те две величине:

$$L_h e_h = r_h,$$

која се назива резидуална једначина. Она показује да грешка задовољава исти систем једначина као и непозната v , с тим што се f_h замени резидуалом r_h .

Зашто је резидуална једначина значајна? Претпоставимо да је \tilde{v} израчунато неком (итеративном) методом. Помоћу $r_h = f_h - L_h \tilde{v}$ лако израчунамо резидуал. Затим, ради побољшања апроксимације v , решавамо резидуалну једначину по e_h и рачунамо нову апроксимацију као $\tilde{v} + e_h$.

4.5 Multigrid алгоритам за једнодимензиони проблем

За пренос информација између мрежа потребни су нам адекватни оператори. Размотримо само две мреже: фину мрежу с кораком h и са $n - 1$ чворова (искључујући граничне чворове) и грубу мрежу са кораком $2h$ и са $n/2 - 1$ чворова. За пренос података са фине на грубу мрежу користимо оператор рестрикције I_h^{2h} . Ако је g^h

вектор дефинисан на финој мрежи, одговарајући вектор на грубој мрежи је

$$(I_h^{2h} g^h)_i = g_{2i}^h, \quad i = 1, 2, \dots, n/2 - 1,$$

тј. вредности грубе мреже су дате вредностима фино мреже, које се налазе на одговарајућим локацијама. Одговарајући вектор грубе мреже се може добити и локалним усредњавањем:

$$(I_h^{2h} g^h)_i = \frac{1}{4} (g_{2i-1}^h + 2g_{2i}^h + g_{2i+1}^h), \quad i = 1, 2, \dots, n/2 - 1.$$

За пренос информација са грубе на фину мрежу користимо оператор продужења I_{2h}^h . Ако је g^{2h} вектор дефинисан на грубој мрежи, одговарајући вектор на финој мрежи се добија обичном линеарном интерполацијом:

$$(I_{2h}^h g^{2h})_i = \begin{cases} g_{i/2}^{2h}, & \text{ако је } i \text{ паран,} \\ \frac{1}{2} (g_{(i+1)/2}^{2h} + g_{(i-1)/2}^{2h}), & \text{ако је } i \text{ непаран,} \end{cases}$$

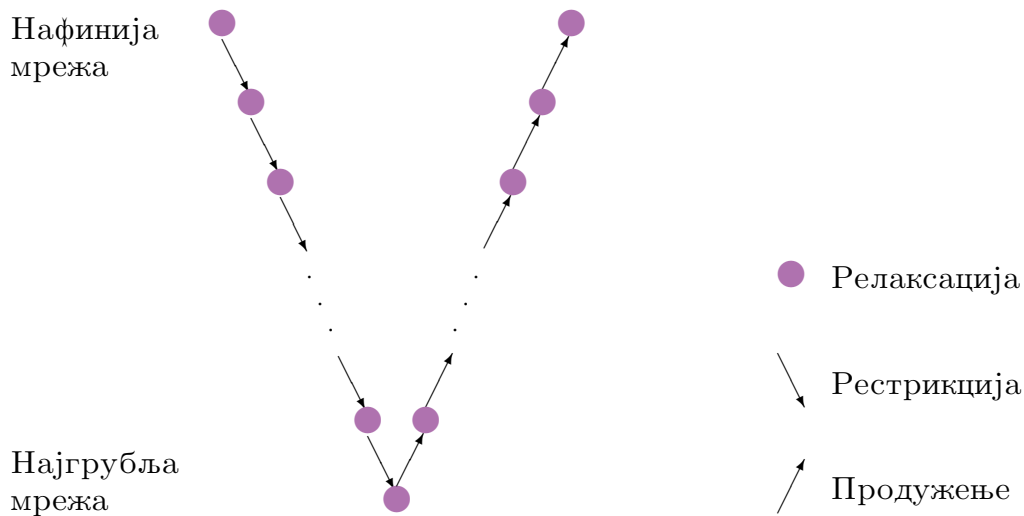
$i = 1, 2, \dots, n - 1$.

Даље, морамо дефинисати оператор грубе мреже L_{2h} , који апроксимира оператор фино мреже L_h . L_{2h} можемо увести дискретизацијом диференцијалне једначине на грубој мрежи користећи схему сличну оној коју смо користили за фину мрежу. Помоћу ових оператора и употребом рекурзије за приближно решавање проблема на грубој мрежи, можемо дефинисати тзв. multigrid V-циклус (слика 10).

Уведимо два параметра, p_1 и p_2 , који представљају, редом, број релаксација (у Јасоби-јевој итеративној методи са релаксационим параметром) пре и након корекције на грубој мрежи. Како бисмо нагласили на којој мрежи се тренутно налазимо, уместо v користимо ознаку v_h . Multigrid алгоритам, који се још назива и multigrid V-циклус (скраћено MGV), дефинише се са:

Алгоритам V-циклуса: $v_h = \text{Vciklus}(v_h, n, h, f_h, p_1, p_2)$

1. Ако је $n = 2$, решити $L_h v_h = f_h$ и вратити v_h ;
2. $v_h = \text{relaksacija}(v_h, n, h, f_h, p_1)$;
релаксирати p_1 пута.
3. $r_h = f_h - L_h v_h$;
израчунати резидуал.



Слика 10: Шематски приказ V-циклуса. Алгоритам напредује с лева на десно и с врха (најфинија мрежа) ка дну (најгрубља мрежа), а затим с дна ка врху. На свакој мрежи, осим на најгрубљој, релаксирамо p_1 пута пре преноса (рестрикције) на следећу, грубљу мрежу, и p_2 пута након интерполације и додавања корекције.

4. $f_{2h} = I_h^{2h} r_h$;
рестрикција резидуала.
5. $v_{2h} = 0$;
поставити v_{2h} на нулу, укључујући и граничне услове.
6. $v_{2h} = \text{Vciklus}(v_{2h}, n/2, 2h, f_{2h}, p_1, p_2)$;
рекурзивни позив и решавање проблема на грубљој мрежи.
7. $v_h = v_h + I_{2h}^h v_{2h}$;
интерполација и додавање корекције.
8. $v_h = \text{relaksacija}(v_h, n, h, f_h, p_2)$;
релаксирати p_2 пута.
9. Вратити v_h .

Проучимо алгоритам корак по корак. У кораку 1 проверавамо да ли смо стигли до најгрубље мреже, тј. да ли је $n = 2$. Ако је тако, решимо проблем помоћу једне итерације (обичне) Јасобијеве итеративне методе (то је довољно, јер постоји само једна променљива). У кораку 2 изврши се p_1 итерација Јасобијеве ите-

ративне методе са релаксационим параметром (тј. релаксирамо p_1 пута). У кораку 3 рачунамо резидуал, а у кораку 4 вршимо његову рестрикцију. Сада треба да решавамо проблем на грубој мрежи и на њој искористимо апроксимацију грешке са fine мреже. С обзиром да је проблем на грубој сличан проблему на fine мрежи, у кораку 6 примењујемо рекурзивни позив. Приметимо да су гранични услови на грубој мрежи једнаки нули (корак 5) зато што не постоји грешка у граничним вредностима на fine мрежи, како је претпостављано. Такође, почетна апроксимација v_{2h} мора бити нула. Након повратка из рекурзије, интерполирамо и додајемо корекцију апроксимацији на fine мрежи (корак 7). У кораку 8 релаксирамо p_2 пута (на fine мрежи), а коначна апроксимација решења се враћа у кораку 9. Генерално, V-циклус не решава проблем егзактно и треба га примењивати итеративно. Свака таква итерација значајно смањује алгебарску грешку.

4.6 Multigrid алгоритам за дводимензиони проблем

Како бисмо применили multigrid V-циклус на дводимензиони проблем и даље морамо да дефинишемо операторе грубе мреже, заједно са одговарајућим операторима преноса између мрежа. Елиминишимо све тачке мреже у којима је i или j непарно, тако да преостане отприлике $1/4$ тачака мреже. За рестрикцију поново можемо користити оператор I_h^{2h} , који се за вектор fine мреже g^h дефинише са

$$(I_h^{2h} g^h)_{i,j} = g_{2i,2j}^h, \quad i, j = 1, 2, \dots, n/2 - 1,$$

тј. вредности грубе мреже су дате вредностима fine мреже које се налазе на одговарајућим локацијама, или су, применом локалног усредњавања, дефинисане са

$$(I_h^{2h} g^h)_{i,j} = \frac{1}{16} (g_{2i+1,2j+1}^h + g_{2i+1,2j-1}^h + g_{2i-1,2j+1}^h + g_{2i-1,2j-1}^h +$$

$$2 (g_{2i+1,2j}^h + g_{2i-1,2j}^h + g_{2i,2j+1}^h + g_{2i,2j-1}^h) + 4g_{2i,2j}^h), \quad i, j = 1, 2, \dots, n/2 - 1,$$

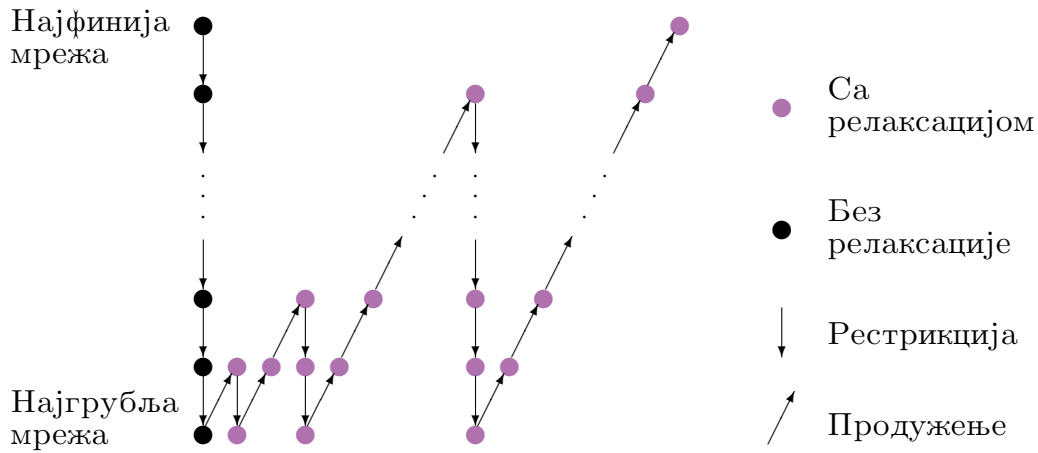
За пренос података са грубе на фину мрежу, прикладан избор је билинеарна интерполација, дефинисана за векторе грубе мреже g^{2h} са

$$(I_{2h}^h g^{2h})_{ij} = \begin{cases} g_{i/2,j/2}^{2h}, & i \text{ паран, } j \text{ паран,} \\ \frac{1}{2} \left(g_{(i+1)/2,j/2}^{2h} + g_{(i-1)/2,j/2}^{2h} \right), & i \text{ непаран, } j \text{ паран,} \\ \frac{1}{2} \left(g_{i/2,(j+1)/2}^{2h} + g_{i/2,(j-1)/2}^{2h} \right), & i \text{ паран, } j \text{ непаран,} \\ \frac{1}{4} \left(g_{(i+1)/2,(j+1)/2}^{2h} + g_{(i-1)/2,(j+1)/2}^{2h} \right. \\ \left. + g_{(i+1)/2,(j-1)/2}^{2h} + g_{(i-1)/2,(j-1)/2}^{2h} \right), & i \text{ непаран, } j \text{ непаран,} \end{cases}$$

$$i = 1, 2, \dots, n - 1.$$

4.7 FMG алгоритам

У нумеричким експериментима можемо користити случајно изабрану почетну вредност решења и употребити је за тестирање. Међутим, у пракси желимо да изаберемо најбољу могућу почетну вредност. Природан и ефикасан приступ је да прво решимо проблем приближно на грубљој мрежи и оператором интерполације пренесемо то решење на фину мрежу како бисмо га користили као почетну апроксимацију. Ово се примењује рекурзивно и тако настаје FMG (Full Multigrid) алгоритам. Слика 11 даје шематски приказ. Потребан нам је позитиван параметар μ , који одређује



Слика 11: Шематски приказ FMG алгоритма за $\mu = 1$. Алгоритам напредује с лева на десно и с врха (најфинија мрежа) ка дну (најгрубља мрежа). На почетку преносимо проблем до најгрубље мреже. Затим решавамо проблем на најгрубљој мрежи, па оператором интерполације преносимо решење на следећу, финију мрежу и применимо V-циклус. Ова два корака се рекурзивно понављају ка све финијој и финијој мрежи, док не дођемо до најфиније.

број V-циклуса примењених на сваком нивоу алгоритма. FMG алгоритам је дат са:

FMG алгоритам: $v_h = \text{FMG}(n, h, f_h, p_1, p_2, \mu)$

1. Ако је $n = 2$, решити $L_h v_h = f_h$ и вратити v_h ;
2. $f_{2h} = I_h^{2h} f_h$;
рестрикција f_h .
3. $v_{2h} = \text{FMG}(n/2, 2h, f_{2h}, p_1, p_2, \mu)$;
рекурзивни позив.
4. $v_h = \hat{I}_{2h}^h v_{2h}$;
интерполација почетне апроксимације, без граничних услова.
5. Поновити μ пута $v_h = \text{Vciklus}(v_h, n, h, f_h, p_1, p_2)$;
примена V-циклуса μ пута.
6. Вратити v_h .

\hat{I}_{2h}^h је оператор продужења и може се разликовати од оног који користимо у V-циклусу. Генерално, треба да буде прецизнији, тако да је добар избор (би)кубна интерполација. Подсетимо се раз-

лике које смо направили раније између грешке дискретизације и алгебарске грешке. Укупна грешка је сума ових двеју грешака - тј. разлика између решења диференцијалне једначине и тренутне дискретне апроксимације. Ово је релевантна мера прецизности тренутне апроксимације. Када алгебарска грешка постане мања од грешке дискретизације, ова друга почиње да доминира у укупној грешци.

Очигледно нема користи од покушаја да се даље смањује алгебарска грешка. Заиста, ако бисмо због тога хтели да уложимо додатни напор, морали бисмо да смањимо грешку дискретизације употребивши још финију мрежу. За многе проблеме је показано да FMG алгоритам може алгебарску грешку учинити много мањом од грешке дискретизације, чак и за $\mu = 1$.

4.8 Анализа сложености (дводимензиони проблем)

4.8.1 Анализа сложености MG V алгоритма

За грубу анализу сложености MG V -а довољно је знати да сваки од оператора замењује вредност у поједином чвору мреже неком тежинском средњом вредношћу у том чвору и константном броју његових суседа. Дакле, за сваки чвор мреже потребно је (највише) константан број рачунских операција, па је сложеност сваког оператора за n непознатих једнака $O(n)$, тј. линеарна по n . Потпуно исто важи за све кораке у MG V алгоритму, до на рекурзивни позив. Рекурзивни позиви MG V алгоритма описани су V -циклом. У свакој тачки на кораку i у V -циклусу пре и после рекурзивног позива, потребно је

$$O(n_i^2) = O((2^i - 1)^2) = O(4^i)$$

рачунских операција. Квадрат долази из димензије проблема (решавамо дводимензиони проблем). Ако се најфинија мрежа налази на растојању k , са $n = (2^k - 1)^2 \approx 4^k$ непознатих, онда укупан број рачунских операција у MGV алгоритму има ред величине описан геометријском сумом

$$\sum_{i=1}^k O(4^i) = O(4^k).$$

Дакле, број операција у MGV алгоритму је линеаран по броју непознатих. У секвенцијалној имплементацији алгоритма, то важи и за временску сложеност.

Исти закључак излази и директно - рекурзијом за сложеност. Нека је $T(i)$ број рачунских операција у MGV алгоритму на улазној мрежи са индексом i , тј. на мрежи са $n_i^2 = (2^i - 1)^2 = O(4^i)$ непознатих. Онда је

$$T(i) = T(i - 1) + O(4^i), \quad i > 1,$$

уз $T(1) = const$, па је $T(k) = O(4^k)$.

4.8.2 Анализа сложености FMG алгоритма

Сложеност FMG алгоритма се одређује тривијално. Сваки V - циклус (слика 11) представља један позив MGV алгоритма унутар петље FMG алгоритма. С обзиром да V-циклус почиње (и завршава) на нивоу i , потребно је $T(i) = O(4^i)$ операција.

Сложеност FMG алгоритма је збир сложености свих позива MGV алгоритма. Укупан број операција је

$$\sum_{i=1}^k T(i) = \sum_{i=1}^k O(4^i) = O(4^k) = O(\text{број непознатих}).$$

Константа пропорционалности скривена у ознаци O је мало већа (али не пуно) од оне за основни алгоритам.

Број операција у целом FMG алгоритму је линеаран по броју непознатих n , што је оптимално, до на мултипликативну константу, јер захтева константан број операција по свакој непознатој.

4.9 Пример

Нека је $\Omega = (0, 1)^2$. Решити гранични проблем

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= -2\pi^2 \sin(\pi x) \sin(\pi y), \quad (x, y)^T \in \Omega, \\ u(x, y) &= 0, \quad (x, y)^T \in \Gamma \equiv \partial\Omega. \end{aligned}$$

Решење Овде се ради о дводимензионој Poisson-овој једначини са хомогеним Dirichlet-овим граничним условом. За решавање датог проблема користимо multigrid методу. Притом се ослањамо на приложене кодове писане у Matlab-у (у сваком кораку позивамо одговарајућу функцију, а исти резултат се добија и једним позивом FMG функције).

Оператор рестрикције задајмо са

$$(I_h^{2h} g^h)_{i,j} = \frac{1}{16} (g_{2i-1,2j-1}^h + g_{2i-1,2j+1}^h + g_{2i+1,2j-1}^h + g_{2i+1,2j+1}^h +$$

$$2 (g_{2i-1,2j}^h + g_{2i+1,2j}^h + g_{2i,2j-1}^h + g_{2i,2j+1}^h) + 4g_{2i,2j}^h), \quad i, j = 1, 2, \dots, n/2 - 1,$$

а оператор продужења са

$$(I_{2h}^h g^{2h})_{ij} = \begin{cases} g_{i/2,j/2}^{2h}, & i \text{ паран, } j \text{ паран,} \\ \frac{1}{2} \left(g_{(i+1)/2,j/2}^{2h} + g_{(i-1)/2,j/2}^{2h} \right), & i \text{ непаран, } j \text{ паран,} \\ \frac{1}{2} \left(g_{i/2,(j+1)/2}^{2h} + g_{i/2,(j-1)/2}^{2h} \right), & i \text{ паран, } j \text{ непаран,} \\ \frac{1}{4} \left(g_{(i+1)/2,(j+1)/2}^{2h} + g_{(i-1)/2,(j+1)/2}^{2h} \right. \\ \left. + g_{(i+1)/2,(j-1)/2}^{2h} + g_{(i-1)/2,(j-1)/2}^{2h} \right), & i \text{ непаран, } j \text{ непаран,} \end{cases}$$

$i = 1, 2, \dots, n - 1$.

Како бисмо употребили FMG алгоритам, морамо одабрати улазне податке. Нека је $n = 8 = 2^3$, што значи да ћемо проблем решавати на 3 мреже; n одговара најфинијој мрежи, која има $(n + 1)^2 = 81$ чвор. Корак најфиније мреже је $h = 1/n = 0.125$. Узмимо још $p_1 = p_2 = 10$ и нека је $\mu = 1$.

Поред наведеног, потребно је одредити и f_h (такође улазни податак), које можемо представити као матрицу димензије $(n + 1) \times (n + 1) = 9 \times 9$ са нулама у првој и последњој врсти, као и у првој и последњој колони, а остали чланови су одговарајуће вредности десне стране $f(x, y) = -2\pi^2 \sin(\pi x) \sin(\pi y)$ у унутрашњим чворовима најфиније мреже:

$$f_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2.8908 & -5.3414 & -6.9789 & -7.5539 & -6.9789 & -5.3414 & -2.8907 & 0 \\ 0 & -5.3414 & -9.8697 & -12.8954 & -13.9578 & -12.8953 & -9.8696 & -5.3413 & 0 \\ 0 & -6.9789 & -12.8954 & -16.8486 & -18.2368 & -16.8485 & -12.8953 & -6.9788 & 0 \\ 0 & -7.5539 & -13.9578 & -18.2368 & -19.7393 & -18.2367 & -13.9577 & -7.5538 & 0 \\ 0 & -6.9789 & -12.8953 & -16.8485 & -18.2367 & -16.8485 & -12.8952 & -6.9788 & 0 \\ 0 & -5.3414 & -9.8696 & -12.8953 & -13.9577 & -12.8952 & -9.8695 & -5.3413 & 0 \\ 0 & -2.8907 & -5.3413 & -6.9788 & -7.5538 & -6.9788 & -5.3413 & -2.8907 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Пошто је $n = 8 \neq 2$, вршимо рестрикцију f_h :

$$f_{2h} = I_h^{2h} f_h,$$

и добијамо

$$f_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -9.1327 & -12.9156 & -9.1326 & 0 \\ 0 & -12.9156 & -18.2653 & -12.9155 & 0 \\ 0 & -9.1326 & -12.9155 & -9.1326 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Следи рекурзивни позив FMG алгоритма, чији су аргументи $n/2 = 4$, $2h = 0.25$, управо израчунато f_{2h} и већ задати p_1 , p_2 и μ .

Како је $n/2 = 4 \neq 2$, опет вршимо рестрикцију, овог пута f_{2h} :

$$f_{4h} = I_{2h}^{4h} f_{2h},$$

и добијамо

$$f_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -13.3072 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

Следи још један рекурзивни позив FMG алгоритма. Овог пута су аргументи $n/4 = 2$, $4h = 0.5$, f_{4h} , а p_1 , p_2 и μ остају непромењени. Како је $n/4 = 2$ решавамо

$$L_{4h}\mathbf{v}_{4h} = f_{4h} ,$$

што, с обзором на димензију проблема и хомогеност граничних услова, није тешко. Из једначине

$$\frac{v_{4h,21} - 2v_{4h,11} + v_{4h,01}}{(4h)^2} + \frac{v_{4h,12} - 2v_{4h,11} + v_{4h,10}}{(4h)^2} = f_{4h,11} ,$$

уврстивши $v_{4h,21} = v_{4h,01} = v_{4h,12} = v_{4h,10} = 0$, добијамо

$$v_{4h,11} = -\frac{1}{4}(4h)^2 f_{4h,11} = 0.8317 ,$$

па је

$$\mathbf{v}_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.8317 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

Ово је податак са којим излазимо из другог и враћамо се тамо где смо стали у првом рекурзивном позиву FMG алгоритма.

Сада оператором продужења дејствујемо на \mathbf{v}_{4h} :

$$\mathbf{v}_{2h} = I_{4h}^{2h}\mathbf{v}_{4h} ,$$

и добијамо

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2079 & 0.4159 & 0.2079 & 0 \\ 0 & 0.4159 & 0.8317 & 0.4159 & 0 \\ 0 & 0.2079 & 0.4159 & 0.2079 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

Следи позив V-цикла, са аргументима \mathbf{v}_{2h} , $n/2$, $2h$, f_{2h} , p_1 и p_2 . У првом кораку V-цикла уочавамо да је $n/2 = 4 \neq 2$, па прелазимо на следећи корак у коме решавамо проблем

$$L_{2h}\mathbf{v}_{2h} = f_{2h}$$

помоћу Јасоби-јеве итеративне методе са релаксационим параметром. Извршићемо $p_1 = 10$ итерација. Користећи

$$v_{2h,ij}^{(m)} = \frac{1}{3}v_{2h,ij}^{(m-1)} + \frac{2}{3} \left(v_{2h,i+1,j}^{(m-1)} + v_{2h,i-1,j}^{(m-1)} + v_{2h,i,j+1}^{(m-1)} + v_{2h,i,j-1}^{(m-1)} - h^2 f_{2h,ij} \right),$$

$$i, j = 1, 2, \dots, n/2 - 1, 1 \leq m \leq 10,$$

добивамо

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3031 & 0.4811 & 0.3031 & 0 \\ 0 & 0.4811 & 0.7447 & 0.4811 & 0 \\ 0 & 0.3031 & 0.4811 & 0.3031 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Рачунамо резидуал по формули

$$r_{2h} = f_{2h} - L_{2h}\mathbf{v}_{2h},$$

па је

$$\mathbf{r}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -9.5159 & -13.4551 & -9.5159 & 0 \\ 0 & -13.4551 & -19.0318 & -13.4551 & 0 \\ 0 & -9.5159 & -13.4551 & -9.5159 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

У наредном кораку вршино рестрикцију резидуала:

$$f_{4h} = I_{2h}^{4h} r_{2h}$$

и добијамо

$$f_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -13.8644 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Сада поставимо

$$\mathbf{v}_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Следи рекурзивни позив V-цикла са аргументима \mathbf{v}_{4h} , $n/4$, $4h$, f_{4h} , p_1 и p_2 .

Како је $n/3 = 2$, није тешко одредити решење једначине

$$L_{4h}\mathbf{v}_{4h} = f_{4h},$$

које износи

$$\mathbf{v}_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.8665 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Овим излазимо из рекурзивног позива V-циклуса и враћамо се на његов први позив (онај који је извршен у FMG алгоритму).

У следећем кораку оператор продужења дејствује на \mathbf{v}_{4h} и додајемо корекцију:

$$\mathbf{v}_{2h} = \mathbf{v}_{2h} + I_{2h}^{4h}\mathbf{v}_{4h},$$

па је

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5197 & 0.9143 & 0.5197 & 0 \\ 0 & 0.9143 & 1.6113 & 0.9143 & 0 \\ 0 & 0.5197 & 0.9143 & 0.5197 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Поново помоћу Јасоби-јеве итеративне методе са релаксационим параметром, сада са $p_2 = 10$ итерација, решавамо

$$L_{2h}\mathbf{v}_{2h} = f_{2h}$$

и добијамо

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5731 & 0.8811 & 0.5731 & 0 \\ 0 & 0.8811 & 1.3369 & 0.8811 & 0 \\ 0 & 0.5731 & 0.8811 & 0.5731 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Враћамо се на FMG алгоритам. Пошто је $\mu = 1$, са овим резултатом излазимо из првог рекурзивног позива FMG алгоритма и враћамо се полазном алгоритму.

У наредном кораку оператором продужења дејствујемо на \mathbf{v}_{2h} :

$$\mathbf{v}_h = I_{2h}^h\mathbf{v}_{2h}$$

и добијамо

$$\mathbf{v}_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1433 & 0.2866 & 0.3636 & 0.4405 & 0.3636 & 0.2866 & 0.1433 & 0 \\ 0 & 0.2866 & 0.5731 & 0.7271 & 0.8811 & 0.7271 & 0.5731 & 0.2866 & 0 \\ 0 & 0.3636 & 0.7271 & 0.9181 & 1.1090 & 0.9181 & 0.7271 & 0.3636 & 0 \\ 0 & 0.4405 & 0.8811 & 1.1090 & 1.3369 & 1.1090 & 0.8811 & 0.4405 & 0 \\ 0 & 0.3636 & 0.7271 & 0.9181 & 1.1090 & 0.9181 & 0.7271 & 0.3636 & 0 \\ 0 & 0.2866 & 0.5731 & 0.7271 & 0.8811 & 0.7271 & 0.5731 & 0.2866 & 0 \\ 0 & 0.1433 & 0.2866 & 0.3636 & 0.4405 & 0.3636 & 0.2866 & 0.1433 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

Следећи корак је позив V -цикласа, са аргументима \mathbf{v}_h , n , h , f_h , p_1 , p_2 . Пошто је $n = 8 \neq 2$, редом рачунамо следеће вредности:

$$\mathbf{v}_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1508 & 0.2894 & 0.3817 & 0.4346 & 0.3817 & 0.2894 & 0.1508 & 0 \\ 0 & 0.2894 & 0.5546 & 0.7319 & 0.8307 & 0.7319 & 0.5546 & 0.2894 & 0 \\ 0 & 0.3817 & 0.7319 & 0.9619 & 1.0928 & 0.9619 & 0.7319 & 0.3817 & 0 \\ 0 & 0.4346 & 0.8307 & 1.0928 & 1.2364 & 1.0928 & 0.8307 & 0.4346 & 0 \\ 0 & 0.3817 & 0.7319 & 0.9619 & 1.0928 & 0.9619 & 0.7319 & 0.3817 & 0 \\ 0 & 0.2894 & 0.5546 & 0.7319 & 0.8307 & 0.7319 & 0.5546 & 0.2894 & 0 \\ 0 & 0.1508 & 0.2894 & 0.3817 & 0.4346 & 0.3817 & 0.2894 & 0.1508 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{r}_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3.0468 & -5.6341 & -7.3702 & -7.9820 & -7.3701 & -5.6341 & -3.0467 & 0 & 0 \\ 0 & -5.6341 & -10.4189 & -13.6280 & -14.7602 & -13.6279 & -10.4188 & -5.6340 & 0 & 0 \\ 0 & -7.3702 & -13.6280 & -17.8268 & -19.3057 & -17.8267 & -13.6279 & -7.3700 & 0 & 0 \\ 0 & -7.9820 & -14.7602 & -19.3057 & -20.9093 & -19.3057 & -14.7601 & -7.9818 & 0 & 0 \\ 0 & -7.3701 & -13.6279 & -17.8267 & -19.3057 & -17.8267 & -13.6278 & -7.3700 & 0 & 0 \\ 0 & -5.6341 & -10.4188 & -13.6279 & -14.7601 & -13.6278 & -10.4188 & -5.6340 & 0 & 0 \\ 0 & -3.0467 & -5.6340 & -7.3700 & -7.9818 & -7.3700 & -5.6340 & -3.0467 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$f_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -9.6461 & -13.6576 & -9.6460 & 0 \\ 0 & -13.6576 & -19.3368 & -13.6575 & 0 \\ 0 & -9.6460 & -13.6575 & -9.6460 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

и постављамо

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Следи рекурзивни позив V-циклуса, са аргументима \mathbf{v}_{2h} , $n/2$, $2h$, f_{2h} , p_1 , p_2 . Пошто је $n/2 = 4 \neq 2$, редом рачунамо следеће вредности:

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1005 & 0.1423 & 0.1005 & 0 \\ 0 & 0.1423 & 0.2014 & 0.1423 & 0 \\ 0 & 0.1005 & 0.1423 & 0.1005 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{r}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -9.8680 & -13.9716 & -9.8679 & 0 \\ 0 & -13.9716 & -19.7813 & -13.9715 & 0 \\ 0 & -9.8679 & -13.9715 & -9.8678 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$f_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -14.3981 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

и постављамо

$$\mathbf{v}_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Сада вршимо други рекурзивни позив V-циклуса, са аргументима \mathbf{v}_{4h} , $n/4$, $4h$, f_{4h} , p_1 , p_2 . Пошто је $n/4 = 2$, решавамо

$$L_{4h}\mathbf{v}_{4h} = f_{4h}$$

и добијамо

$$\mathbf{v}_{4h} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.8999 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Овим излазимо из другог рекурзивног позива V -цикла и враћамо се на прву рекурзију. Добијамо редом:

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3255 & 0.5922 & 0.3254 & 0 \\ 0 & 0.5922 & 1.1013 & 0.5922 & 0 \\ 0 & 0.3254 & 0.5922 & 0.3254 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

$$\mathbf{v}_{2h} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4064 & 0.6317 & 0.4064 & 0 \\ 0 & 0.6317 & 0.9633 & 0.6317 & 0 \\ 0 & 0.4064 & 0.6317 & 0.4064 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

Сада излазимо и из првог рекурзивног позива V -цикла и враћамо се на почетни позив. Редом добијамо:

$$\mathbf{v}_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2524 & 0.4926 & 0.6413 & 0.7504 & 0.6412 & 0.4926 & 0.2524 & 0 \\ 0 & 0.4926 & 0.9610 & 1.2510 & 1.4624 & 1.2510 & 0.9610 & 0.4926 & 0 \\ 0 & 0.6413 & 1.2510 & 1.6202 & 1.8904 & 1.6202 & 1.2510 & 0.6412 & 0 \\ 0 & 0.7504 & 1.4624 & 1.8904 & 2.1997 & 1.8904 & 1.4624 & 0.7504 & 0 \\ 0 & 0.6412 & 1.2510 & 1.6202 & 1.8904 & 1.6202 & 1.2510 & 0.6412 & 0 \\ 0 & 0.4926 & 0.9610 & 1.2510 & 1.4624 & 1.2510 & 0.9610 & 0.4926 & 0 \\ 0 & 0.2524 & 0.4926 & 0.6412 & 0.7504 & 0.6412 & 0.4926 & 0.2524 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

$$\mathbf{v}_h = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2559 & 0.4872 & 0.6476 & 0.7273 & 0.6476 & 0.4872 & 0.2559 & 0 \\ 0 & 0.4872 & 0.9272 & 1.2314 & 1.3809 & 1.2314 & 0.9272 & 0.4872 & 0 \\ 0 & 0.6476 & 1.2314 & 1.6311 & 1.8280 & 1.6311 & 1.2314 & 0.6476 & 0 \\ 0 & 0.7273 & 1.3809 & 1.8280 & 2.0449 & 1.8280 & 1.3809 & 0.7273 & 0 \\ 0 & 0.6476 & 1.2314 & 1.6311 & 1.8280 & 1.6311 & 1.2314 & 0.6476 & 0 \\ 0 & 0.4872 & 0.9272 & 1.2314 & 1.3809 & 1.2314 & 0.9272 & 0.4872 & 0 \\ 0 & 0.2559 & 0.4872 & 0.6476 & 0.7273 & 0.6476 & 0.4872 & 0.2559 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

Последња матрица представља и крајње решење које добијамо применом FMG алгорита.

Аналитичко решење разматраног проблема је $\sin(\pi x) \sin(\pi y)$, а тачне вредности у чворовима најфиније мреже су приказане у следећој матрици:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1464 & 0.2706 & 0.3536 & 0.3827 & 0.3536 & 0.2706 & 0.1464 & 0 \\ 0 & 0.2706 & 0.5000 & 0.6533 & 0.7071 & 0.6533 & 0.5000 & 0.2706 & 0 \\ 0 & 0.3536 & 0.6533 & 0.8536 & 0.9239 & 0.8536 & 0.6533 & 0.3535 & 0 \\ 0 & 0.3827 & 0.7071 & 0.9239 & 1.0000 & 0.9239 & 0.7071 & 0.3827 & 0 \\ 0 & 0.3536 & 0.6533 & 0.8536 & 0.9239 & 0.8536 & 0.6533 & 0.3535 & 0 \\ 0 & 0.2706 & 0.5000 & 0.6533 & 0.7071 & 0.6533 & 0.5000 & 0.2706 & 0 \\ 0 & 0.1464 & 0.2706 & 0.3535 & 0.3827 & 0.3535 & 0.2706 & 0.1464 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Нумеричко решавање је урађено помоћу следећих matlab кодова:

desnastrana.m

```
function f = desnastrana(x, y)
f = - 2 * 3.1416^2 * sin(3.1416 * x) * sin(3.1416 * y);
```

fh.m

```
function fh = fh(n, h)
fh = zeros(n + 1, n + 1);
for i = 2 : n
    for j = 2 : n
        fh(i, j) = desnastrana((i - 1) * h, (j - 1) * h);
    end
end
```

restrikcija.m

```
function f2h = restrikcija(n, h, fh)
f2h = zeros(n / 2 + 1, n / 2 + 1);
for i = 2 : n / 2
    for j = 2 : n / 2
        f2h(i, j) = 1 / 16 * (fh(2 * i - 2, 2 * j - 2) + fh(2 * i - 2, 2 * j)
```

```

+ fh(2 * i, 2 * j - 2) + fh(2 * i, 2 * j)
+ 2 * (fh(2 * i - 2, 2 * j - 1) + fh(2 * i, 2 * j - 1)
+ fh(2 * i - 1, 2 * j - 2) + fh(2 * i - 1, 2 * j))
+ 4 * fh(2 * i - 1, 2 * j - 1));

```

```
end
```

```
end
```

korak1.m

```
function vh = korak1(n, h, fh)
```

```
vh = zeros(n + 1, n + 1);
```

```
vh(n / 2 + 1, n / 2 + 1) = - h^2 * fh(2, 2) / 4;
```

produzenje.m

```
function vh = produzenje(v2h, n, h)
```

```
vh = zeros(n + 1, n + 1);
```

```
for i = 2 : n
```

```
    for j = 2 : n
```

```
        if mod(i, 2) ~= 0
```

```
            if mod(j, 2) ~= 0
```

```
                vh(i, j) = v2h((i + 1) / 2, (j + 1) / 2);
```

```
            else
```

```
                vh(i, j) = (v2h((i + 1) / 2, (j + 2) / 2)
+ v2h((i + 1) / 2, j / 2)) / 2;
```

```
            end
```

```
        else
```

```
            if mod(j, 2) ~= 0
```

```
                vh(i, j) = (v2h((i + 2) / 2, (j + 1) / 2)
+ v2h(i / 2, (j + 1) / 2)) / 2;
```

```
            else
```

```
                vh(i, j) = (v2h((i + 2) / 2, (j + 2) / 2) + v2h(i / 2, (j + 2) / 2)
+ v2h((i + 2) / 2, j / 2) + v2h(i / 2, j / 2)) / 4;
```

```
            end
```

```
    end
  end
end
```

relaksacija.m

```
function vh = relaksacija(VH, n, h, fh, p1)
vh = zeros(n + 1, n + 1);
for k = 1 : p1
  for i = 2 : n
    for j = 2 : n
      vh(i, j) = (VH(i, j) + (VH(i + 1, j) + VH(i - 1, j) + VH(i, j + 1)
      + VH(i, j - 1) - h^2 * fh(i, j)) / 2) / 3;
    end
  end
end
end
```

rezidual.m

```
function rh = rezidual(vh, n, h, fh)
Lhvh = zeros(n + 1, n + 1);
for i = 2 : n
  for j = 2 : n
    Lhvh(i, j) = (vh(i + 1, j) + vh(i - 1, j) + vh(i, j + 1) + vh(i, j - 1)
    - h^2 * fh(i, j)) / 4;
  end
end
end
rh = fh - Lhvh;
```

Vciklus.m

```
function vh = Vciklus(VH, n, h, fh, p1, p2)
if n == 2
  vh = korak1(n, h, fh);
```

```

else
    VH = relaksacija(VH, n, h, fh, p1);
    rh = rezidual(VH, n, h, fh);
    f2h = restrikcija(n, h, rh);
    V2H = zeros(n / 2 + 1, n / 2 + 1);
    V2H = Vciklus(V2H, n / 2, 2 * h, f2h, p1, p2);
    VH = VH + produzenje(V2H, n, h);
    vh = relaksacija(VH, n, h, fh, p2);
end

```

FMG.m

```

function vh = FMG(n, h, fh, p1, p2, mi)
if n == 2
    vh = korak1(n, h, fh);
else
    f2h = restrikcija(n, h, fh);
    V2H = FMG(n / 2, 2 * h, f2h, p1, p2, mi);
    VH = produzenje(V2H, n, h);
    for i = 1 : mi
        vh = Vciklus(VH, n, h, fh, p1, p2);
    end
end
end

```

4.10 Примене multigrid-a

У раду смо као моделни задатак разматрали Poisson-ову једначину са Dirichlet-овим граничним условом, у једнодимензионом и дводимензионом случају. Примена multigrid методе је много шира. Она се користи за решавање разних ПДЈ елиптичког, параболичког и хиперболичког типа, са различитим граничним условима, као и за решавање система ПДЈ, а корисна је и код неких интегралних једначина.

Multigrid се примењује и у другим областима математике, као што су интегрални оператори (који се користе за прорачун силе n тела, при дејству гравитације, молекуларне интеракције или термалне радијације), глобална оптимизација (примена у проучавању протеина), оптимално управљање (користи се за планирање кретања робота),...

Најширу практичну примену multigrid налази у динамици флуида, нпр. за предвиђање временских услова. Користи се у динамици крутог тела, затим у статистичкој физици и физици честица (за извођење макроскопских својстава (попут нелинеарне еластичности) применом multigrid-a на микроскопском нивоу, тј. на нивоу атомских сила), као и за симулацију електромагнетне радијације. Multigrid се употребљава и за сегментацију слика и изоштравање ивица слике. Користи се и у медицини, прецизније у томографији (нпр. САТ скенирање). Има значајну примену и у квантној хемији (служи за одређивање сопствених функција Schrödinger²²-ове једначине, помоћу којих се одређује функција густине електрона).

²²Erwin Rudolf Josef Alexander Schrödinger (1887 -1961), аустријски физичар.

5 Закључак

На основу изложеног у раду можемо закључити да multigrid има две предности: прва се огледа у сложености алгоритма, која износи $O(n)$, где је n број непознатих, и која је оптимална (до на мултипликативну константу), а друга у смањењу грешке апроксимације, коју реализујемо редуковањем њених високофреквентних компоненти.

Као што смо на крају претходног поглавља видели, multigrid има широку примену. Предности ове методе и могућности њеног варирања и прилагођавања проблему који разматрамо, мотивишу нас да наставимо са изучавањем ове тематике.

Литература

- [1] R. P. Fedorenko: A Relaxation Method for Solving Elliptic Difference Equations, USSR Computational Mathematics and Mathematical Physics, vol. 1, no. 4, 1962, pp. 1092-1096.
- [2] R. P. Fedorenko: The Speed of Convergence of one Iterative Process, USSR Computational Mathematics and Mathematical Physics, vol. 4, no. 3, 1964, pp. 227-235.
- [3] N. S. Bakhvalov: On the Convergence of a Relaxation Method with Natural Constraints on the Elliptic Operator, USSR Computational Mathematics and Mathematical Physics, vol. 6, no. 5, 1966, pp. 101-135.
- [4] A. Brandt: A Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems, Proc. 3rd Int'l Conf. Numerical Methods in Fluid Mechanics, H. Cabannes and R. Temam, eds., Lecture Notes in Physics 18, Springer, 1973, pp. 82-89.
- [5] A. Brandt: Multi-Level Adaptive Technique (MLAT): The Multigrid Method,, research report RC 6026, IBM T. J. Watson Research Ctr., 1976.
- [6] A. Brandt: Multi-Level Adaptive Solution to Boundary-Value Problems, Mathematics of Computation, vol. 31, no. 138, 1977, pp. 333-390.
- [7] I. Yavneh: Why Multigrid Methods Are So Efficient, Computing in Science and Engineering 8(6): 12-22 (2006).
- [8] W. L. Briggs, V. E. Henson, S. F. McCormick: A Multigrid Tutorial, Second Edition, SIAM, 1999.
- [9] P. Heckbert: Survey of Multigrid Applications, lecture for Hierarchical Methods for Simulation (15-859E), Carnegie Mellon University, 1998.
- [10] S. Singer, S. Singer: Porelni algoritmi 2, predavanja i vježbe (verzija 1.02), Sveučilište u Zagrebu, PMF - Matematički odjel, Zagreb, 2003.
- [11] J. Knežević Miljanović, S. Janković, J. Manojlović, V. Jovanović: Parcijalne diferencijalne jednačine, Univerzitet u Beogradu, Beograd, 2000.
- [12] B. Jovanović: Parcijalne jednačine, Matematički fakultet u Beogradu, Beograd, 1999.

[13] Б. Јовановић, Д. Радуновић: Нумеричка анализа, Математички факултет, Београд, 2003.

[14] www.wikipedia.org