Лекція 11. Мультиплікативне та адитивне передобумовлення

Розглянемо загальну СЛАР

$$Ax = b. \qquad (11.1)$$

Основна ідея передобумовлення полягає у множенні матриці СЛАР на таку матрицю M, щоб добуток MA мав більш прийнятне число обумовленості. В ідеалі, ця матриця повинна наближати обернену матрицю системи. В результаті отримуємо еквівалентну систему

$$MAx = Mb. \qquad (1.2)$$

ILU-факторизація. Одним з широко відомих методів розкладу матриць на множники являється LU-факторизація, яка дозволяє представити A у вигляді

$$A = LU \qquad (11.1)$$

де L і U – нижньо- та верхньотрикутна матриці відповідно.

Таке представлення дозволяє легко розв'язувати СЛАР $Ax = b$ шляхом виконання прямого ходу для нижньотрикутної системі $Ly = b$ і зворотного ходу для верхньотрикутної системи $Ux = y$. Однак алгоритм факторизації не придатний для СЛАР з розрідженими матрицями, оскільки призводить до появи у матрицях L і U ненульових елементів у тих позиціях, для яких $a_{ij} = 0$ і як наслідок, різкого збільшення об'єму пам'яті, необхідного для зберігання матриць.

Уведемо поняття портрету матриці, A тобто сукупності її ненульових елементів і позначимо цього як P_A.

Для того щоб уникнути заповнення розрідженої матриці додатковими ненульовими елементами, замість факторизації (11.1) подамо матрицю A у вигляді

$$A = LU + R \qquad (11.2)$$

де матриці у правій частині задовольняють наступним властивостям:

1. Матриці L та U являються нижньотрикутною та верхньотрикутною відповідно;

2. $P_L \subset P_A$ і $P_U \subset P_A$, тобто портрети матриць L і U містяться в портреті матриці A;

3. $(LU)_{ij} = a_{ij}$, *якщо* $(i, j) \in P_A$

Тоді наближене подання $A \approx LU$ незивається неповною *LU*-факторизацією матриці A, або коротко її *ILU*-розкладом.

Для знаходження матриць L і U будемо генерувати їх по рядкам. Припустимо, що перші $k-1$ рядки вже знайдені і необхідно знайти *k*-й. Запишемо у блочному вигляді перші *k* рядків розкладу (11.2):

$$\begin{pmatrix} A_{11} & A_{12} \\ a_{21}^T & a_{22}^T \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ l_{21}^T & 1 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & u_{22}^T \end{pmatrix} + \begin{pmatrix} R_{11} & R_{11} \\ r_{21}^T & r_{22}^T \end{pmatrix} \quad (4.3)$$

де $l_{21}, u_{22}, r_{21}, r_{22}$ – деякі вектори. Виконавши дії над матрицями у правій частині (4.3), отримаємо

$$\begin{pmatrix} A_{11} & A_{12} \\ a_{21}^T & a_{22}^T \end{pmatrix} = \begin{pmatrix} L_{11}U_{11} + R_{11} & L_{11}U_{12} + R_{12} \\ l_{21}^T U_{11} + r_{21}^T & l_{22}^T U_{12} + u_{22}^T + r_{22}^T \end{pmatrix}$$

З рівності матриць слідує, що шукані вектори l_{21} та u_{22} повинні задовольняти вимогам:

$$l_{21}^T U_{11} + r_{21}^T = a_{21}^T,$$
$$u_{22}^T + r_{22}^T = a_{22}^T - l_{21}^T U_{12}.$$

Розв’язавши ці системи, можна знайти коефіцієнти *k*-х рядків матриць розкладу $l_{k1}, \ldots, l_{k,k-1}, u_{kk}, \ldots, u_{kn}$.

Визначимо l_{kj} припускаючи, що $l_{k1},...,l_{k,j-1}$. Згідно раніше сформульованим умовам, якщо $a_{kj}=0$, то одразу ж $l_{kj}=0$. Інакше $r_{kj}=0$ і $l_{21}^T U_{11}+r_{21}^T=a_{21}^T$ можна записати у вигляді

$$\sum_{i=1}^{j} l_{ki}u_{ij}=\sum_{i=1}^{j-1} l_{ki}u_{ij}+l_{kj}u_{jj}=a_{kj}.$$

Це дозволяє знайти l_{kj} наступним чином:

$$l_{kj}=\frac{1}{u_{ij}}\left(a_{kj}-\sum_{i=1}^{j-1} l_{kj}u_{ij}\right).$$

Враховуючи, що $l_{ij}=1$, аналогічно отримуємо u_{kj}:

$$u_{kj}=a_{kj}-\sum_{i=1}^{k-1} l_{ki}u_{ij}.$$

Оскільки матриця $M=LU$ наближає матрицю A і легко інвертується як добуток двох трикутних матриць, вона є непоганим передобумовлювачем.

SOR-передобумовлення Задану матрицю A подамо у вигляді $A=L+D+U$, де

$$D=\begin{pmatrix} a_{11} & 0 & ... & 0 \\ 0 & a_{22} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & a_{nn} \end{pmatrix},$$

$$L=\begin{pmatrix} 0 & 0 & ... & 0 \\ a_{21} & 0 & ... & 0 \\ ... & ... & ... & 0 \\ a_{n1} & a_{n2} & ... & 0 \end{pmatrix},$$

$$U = \begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Матриця SOR-передобумовлення:

$$M = M_1 M_2,$$

$$M_1 = \frac{1}{\omega} D + L,\ M_2 = \frac{1}{2-\omega}\left(I - \omega D^{-1} U\right),$$

$$0 < \omega < 2.$$

Скалярні множники при практичній реалізації зазвичай не враховуються, оскільки дають лише загальне масштабування, яке практично не впливає на швидкість збіжності.

Двоїста адитивна попередня обробка. Як альтернатива мультиплікативним передобумовлювачам, пропонується адитивний препроцесор $A \to C = A + UV^T$, тобто ми додаємо матрицю UV^T до погано обумовленої вхідної матриці A, щоб отримати її краще обумовлену адитивну модифікацію C.

Для не виродженої матриці A розмірності $n \times n$ можна додати двоїстий адитивний препроцесор VU^T рангу $q \le n$ матриці A^{-1} і визначити $C_- = A^{-1} + VU^T$. Можна обчислити матриці C_- та $A^{-1} = C_- - VU^T$, знаходячи

$$C_-^{-1} = \left(A^{-1} + VU^T\right)^{-1} = A - AVH^{-1}U^T A,$$

$$H = I_q + U^T AV.$$

Останню формулу називатимемо двоїстою формулою обернення Шермана-Моррісона-Вудбері (*SMW*),

$$A^{-1} = \left(C - UV^T\right)^{-1} = C^{-1} + C^{-1}UG^{-1}V^TC^{-1},$$

$$G = I_r - V^TC^{-1}U.$$

Для пошуку розв’язку (4.1) ми можемо обійти обернену матрицю C_{-}^{-1} використовуючи формули

$$x = A^{-1}b = z - VU^Tb,$$

$$\left(C_{-}^{-1}\right)^{-1} z = b\,.$$

При практичній реалізації, для тестування в якості матриць U та V генеруються матриці Тепліца (симетричний та несиметричний випадки), тобто матриці, у якій на неголовних діагоналях стоять рівні елементи:

$$a_{ij} = a_{i-1,\,j-1}\,.$$

Ми можемо контролювати або уникати похибок заокруглення при обчисленні на комп’ютері, якщо заповнювати згенеровані матриці U та V лише цілими числами {-2,-1,0,1,2}.

Передобумовлення методом усікнених рядів

Розглянемо загальний підхід до передобумовлення, в якому матриця A^{-1} у вигляді ряду. В основі цього методу лежить така теорема.

Теорема. Нехай A — невироджена матриця $n \times n$, а $A = P - Q$ — таке розщеплення матриці A, що матриця P є невиродженою і найбільший з модулей власних чисел матриці $P^{-1}Q$ не перевищує одиниці. Тоді

$$A^{-1} = \left(\sum_{k=0}^{\infty} H^k \right) P^{-1},$$

де $H = P^{-1}Q$.

Це значить, що як наближення матриць A і A^{-1} можно брати матриці

$$M = P\left(I + H + ... + H^{m-1}\right)^{-1},$$

$$M^{-1} = \left(I + H + ... + H^{m-1}\right)P^{-1}.$$

Отже, розв’язок допоміжної системи

$$M\tilde{r} = r$$

можна знайти як

$$\tilde{r} = M^{-1}r = \left(I + H + ... + H^{m-1}\right)P^{-1}r.$$

Цей вираз обчислювати не обов’язково. Замість цього, вектор $\tilde{r}$ обчислюється як результат допоміжного ітераційного процесу

$$Pr^{i+1} = Qr^i + r,\ i = 0,1,...,m-1,\ r^0 = 0,\ \tilde{r} = r^m.$$

Результати обчислювального експерименту

Для тестування ефективності використання передобумовлювачів розв'язувались СЛАР з матрицею Гільберта $\left(a_{ij} = \dfrac{1}{i+j-1}\right)$. Для неї одержані наступні результати з критерієм зупинки $\left\|Ax^{(n)} - b\right\| < \varepsilon$, де $x^{(n)}$ — наближення до розв'язку на n-ій ітерації, A – матриця Гільберта, ε — задана точність. В обчисленнях використовувалась норма $\left\|\cdot\right\|_\infty$, що робить критерії зупинки найбільш близькими в машинному розумінні точності.

Достовірність результатів підтверджується перевіркою коректності програмної реалізації на прикладах знаходження заздалегідь відомого розв'язку, співпаданням результатів, отриманих декількома методами.

Література

1. Ортега Дж. Введение в параллельные и векторные методы решения систем. — М.: Мир, 1991.
2. Саад Ю. Итерационные методы для разреженных Московского университета, 2013.
3. Баландин М.Ю., Шурина Э.П. Методы решения СЛАУ большой размерности. Новосибирск: Изд-во НГТУ, 2000.
4. Pan V.Y. et al. Additive preconditioning for matrix computations // Linear Algebra and its Applications. — 2010. — vol. 432, No. 4. — p. 1070–1089.

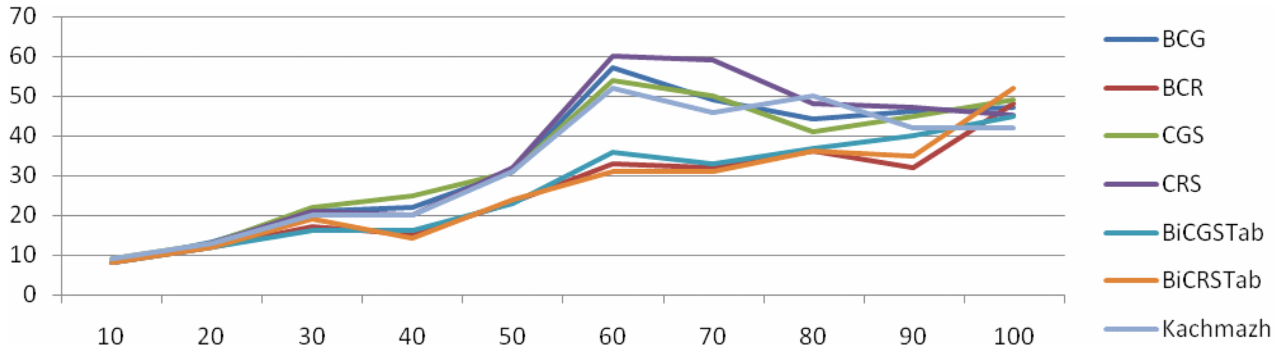
Таблиця 1

$\varepsilon = 10^{-6}$ без передобумовлення

Розмір матриці	**Метод розв'язку СЛАР**						
N	**BCG**	**BCR**	**CGS**	**CRS**	**BiCGStab**	**BiCRStab**	**Kachmaz**
10	8	8	11	11	6	7	12
20	21	29	97	213	30	42	104
30	39	50	164	194	60	74	125
40	65	64	2325	364	121	137	199
50	50	80	>10000	359	229	149	>10000
60	84	131	>10000	>10000	353	230	>10000
70	120	292	>10000	>10000	574	365	>10000
80	216	352	>10000	>10000	1871	1090	>10000
90	186	361	>10000	>10000	1776	937	>10000
100	190	426	>10000	>10000	1910	1172	>10000

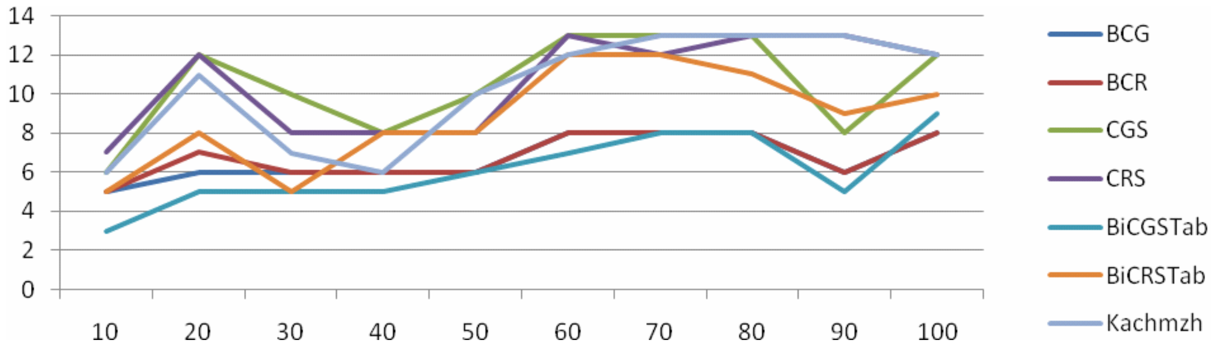
$\varepsilon = 10^{-6}$, SOR-передобумовлення

Розмір матриці	**Метод розв’язку СЛАР**						
N	**BCG**	**BCR**	**CGS**	**CRS**	**BiCGStab**	**BiCRStab**	**Kachmaz**
10	8	8	9	8	8	8	9
20	13	12	13	13	12	12	13
30	21	17	22	21	16	19	20
40	22	15	25	20	16	14	20
50	31	23	31	32	23	24	31
60	57	33	54	60	36	31	52
70	49	32	50	59	33	31	46
80	44	36	41	48	37	36	50
90	46	32	45	47	40	35	42
100	47	48	49	45	45	52	42


70
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90
100
BCG
BCR
CGS
CRS
BiCGSTab
BiCRSTab
Kachmazh

ILU-передобумовлення

Розмір матриці	**Метод розв'язку СЛАР**						
N	**BCG**	**BCR**	**CGS**	**CRS**	**BiCGStab**	**BiCRStab**	**Kachmaz**
10	5	5	6	7	3	5	6
20	6	7	12	12	5	8	11
30	6	6	10	8	5	5	7
40	6	6	8	8	5	8	6
50	6	6	10	8	6	8	10
60	8	8	13	13	7	12	12
70	8	8	13	12	8	12	13
80	8	8	13	13	8	11	13
90	6	6	8	13	5	9	13
100	8	8	12	12	9	10	12


14
12
10
8
6
4
2
0
10
20
30
40
50
60
70
80
90
100
BCG
BCR
CGS
CRS
BiCGSTab
BiCRSTab
Kachmzh

Адитивне передобумовлення з Toeplitz APP

N	Cond(A)	min_Cond(C) Toeplitz APP	min_Cond(C) Toeplitz_symm APP
10	1.89739e+008	1598.7	12912.6
20	7.61909e+008	3797.06	44857.4
30	1.4116e+008	7758.21	804220
40	1.29543e+008	653949	53680
50	3.67404e+008	17931.1	17693.9
60	1.2271e+009	493403	44294.9
70	6.19364e+008	136325	18999.8
80	4.61803e+009	16002	132834
90	1.29082e+009	3.30387e+006	47141.5
100	1.50598e+009	3.26978e+006	38020.9