

Параллельное формирование предобуславливателя на основе обращения Шермана-Моррисона

Н.С. Недожогин, С.П. Копысов, А.К. Новиков

Институт механики УрО РАН

Международная научная конференция
Параллельные вычислительные технологии (ПаВТ) 2015
Екатеринбург, 30 марта — 2 апреля 2015 г.

- 1 Предобуславливание
 - Предобуславливатель AISM
 - Алгоритмы формирования AISM
- 2 Результаты
 - Численные эксперименты
 - Симметричные матрицы
 - Несимметричные матрицы
 - Деформирование сеток
- 3 Вывод

Введение

Исходная система:

$$Ax = b$$

Предобусловленная система:

$$P^{-1}Ax = P^{-1}b$$

Предобуславливатели:


- Неявные
 - Метод неполного LU-разложения
- Явные
 - Разреженные аппроксимации обратной матрицы (AINV)
 - Аппроксимации обратной матрицы в факторизованной форме (FSAI)
 - Полиномиальные (TNS)
 - Основанный на модификации Шермана-Моррисона (AISM)

Пусть B — невырожденная матрица и вектора u и v такие, что $r = 1 + v^T B^{-1} u \neq 0$, матрица $A = B + uv^T$ является обратимой¹ и её обращение находится как

$$A^{-1} = B^{-1} - r^{-1} B^{-1} uv^T B^{-1}. \quad (1)$$

Обозначим u_k, v_k и A_0 — невырожденная матрица, обращение которой просто вычисляется (т.е. A_0 может быть диагональной или даже единичной матрицей). Тогда $A_k = A_0 + \sum_{i=1}^k u_i v_i^T$, где $k = 1, n$ и $A = A_n$. Если A_k, u_k, v_k удовлетворяют выражению (1), тогда обращение матрицы A может быть вычислено при n -кратном использованием (1)

$$A^{-1} = A_0^{-1} - \sum_{k=1}^n r_k^{-1} A_{k-1}^{-1} u_k v_k^T A_{k-1}^{-1}. \quad (2)$$


¹Sherman J., Morrison W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // Ann. Math. Statistics. 1950. V. 21, № 1. P. 124–127. 

Рассмотрим ² вариант когда матрица представима в виде разложения $A = W - Z$, где W — обратимая матрица, $Z = UV^T = \sum_{k=1}^n u_k v_k^T$, а v_k, u_k такие, что $d_k = 1 - v_k^T W_{k-1}^{-1} u_k \neq 0$, где $W_k = W_0 - \sum_{i=1}^k u_i v_i^T$. Задавая выбор матриц $W = \beta \cdot \text{diag}(A)$, $\beta > 0$, $U = I$, $V = Z^T$, получим выражения для вычисления столбцов матриц S и T

$$s_k = u_k - \sum_{i=1}^{k-1} \frac{t_i^T W^{-1} u_k}{d_i} s_i, \quad t_k = v_k - \sum_{i=1}^{k-1} \frac{v_k^T W^{-1} s_i}{d_i} t_i$$

Тогда предобуславливатель, аппроксимирующий обратную матрицу запишем в виде

$$P = W^{-1} - W^{-1} S D^{-1} T^T W^{-1}.$$

²Bru M., Cerdán J., Marín J., Mas J. Preconditioning sparse nonsymmetric linear systems with Sherman-Morrison formula // SIAM J. Sci. Comput. 2003. V. 28, № 2. P. 701–715. 

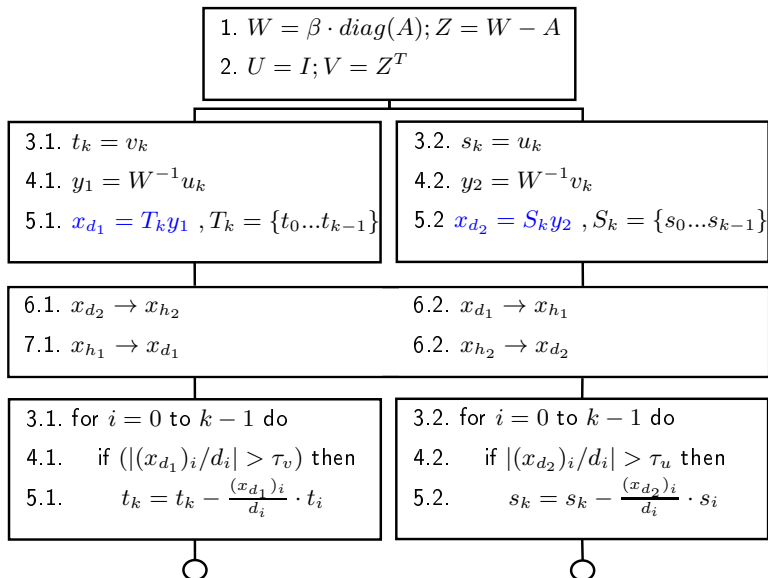
Последовательное формирование AISM

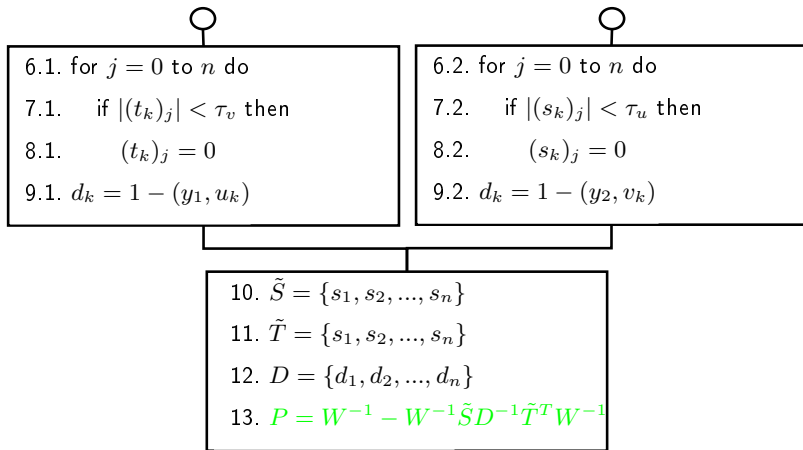
```

1:  $A = W - Z$ 
2:  $W = \beta \cdot \text{diag}(A); Z = W - A$ 
3:  $U = I; V = Z^T$ 
4: for  $k = 0$  to  $n$  do
5:    $s_k = u_k, t_k = v_k$ 
6:   for  $i = 0$  to  $k - 1$  do
7:      $\delta = (t_i^T W^{-1}, u_k)$ 
8:     if  $|\frac{\delta}{d_i}| > \tau_u$  then
9:        $s_k = u_k - \frac{\delta}{d_i} \cdot s_i$ 
10:    end if
11:     $\delta = (v_k^T W^{-1}, s_i)$ 
12:    if  $|\frac{\delta}{d_i}| > \tau_v$  then
13:       $t_k = v_k - \frac{\delta}{d_i} \cdot t_i$ 
14:    end if
15:  end for
16:  for  $j = 0$  to  $n$  do
17:    if  $|(s_k)_j| < \tau_u$  then
18:       $(s_k)_j = 0$ 
19:    end if
20:    if  $|(t_k)_j| < \tau_v$  then
21:       $(t_k)_j = 0$ 
22:    end if
23:  end for
24:   $d_k = 1 - (t_k^T W^{-1}, u_k)$ 
25: end for
26:  $\tilde{S} = \{s_1, s_2, \dots, s_n\},$ 
27:    $\tilde{T} = \{t_1, t_2, \dots, t_n\}$ 
28:  $D = \{d_1, d_2, \dots, d_n\}$ 
29:  $P = W^{-1} - W^{-1} \tilde{S} D^{-1} \tilde{T}^T W^{-1}$ 

```

Параллельное формирование AISM





Численные эксперименты

Параллельные алгоритмы тестировались на GPU-ускорителе GeForce GTX 780 с 3 ГБ графической памяти и на восьми ядрах CPU (два четырехядерных процессора Intel Xeon E5-2609, 2.4 ГГц) и 64 ГБ оперативной памяти).

В численных экспериментах были использованы матрицы из коллекции The University of Florida Sparse Matrix Collection (<http://www.cise.ufl.edu/>).

Решались системы уравнений $Ay = f$ с известным точным решением $y = [1, 1, \dots, 1]$, матрицы которых хранились в сжатом строчном формате (CSR). В качестве начального приближения выбиралось $y_0 = [0, 0, \dots, 0]$, а критерий сходимости — $\|r_i\| \leq 10^{-6} \|r_0\|$, где $r_i = f - Ay_i$.

Симметричные матрицы

Матрица A	N/NNZ	P_{AINV} CPU/GPU	P_{FSAI} CPU/GPU
nasa2910	2910 / 174296	2.5/0.65 (314)	19.5/0.04 (129)
bcsstk15	3948 / 117816	0.15/0.6 (293)	1.47/0.03 (109)
Kuu	7102 / 340200	4.03/0.16 (75)	11.7/0.02 (45)
msc10848	10848 / 1229778	1.48/2.68 (1190)	1168/0.02 (35)
vibrobox	12328 / 301700	1.29/1.42 (683)	85/0.05 (92)

Матрица A	P_{TNS} GPU/GPU	P_{AISM} GPU/GPU	2GPU/GPU
nasa2910	0.002/0.32 (962)	8.78/0.62 (387)	5.62/0.57 (339)
bcsstk15	0.002/0.08 (259)	20.37/0.17 (81)	11.23/0.16 (70)
Kuu	0.004/0.1 (241)	119.03/0.18 (103)	66.7/0.18 (102)
msc10848	0.006/14.7 (21871)	422.45/5.12 (846)	230.715/7.03 (757)
vibrobox	0.003/1.98 (4682)	690.84/0.81 (52)	391.8/0.8 (52)

Несимметричные матрицы

Матрица	N/NNZ	$P_{ILU(0)}$ (CPU/GPU)	$P_{ILU(1)}$ (CPU/GPU)
cdde5	961/ 4681	0.0002/0.13 (140)	0.002/0.12 (106)
ex37	3565 / 67591	0.003/0.03 (3)	1.4/0.03 (2)
rajat03	7602 / 32653	—	—
flowmeter5	9669 / 67391	0.002/0.36 (63)	0.04/0.34 (32)
ex19	12005 / 259577	0.04/—	699/0.5 (279)
sme3Da	12504 /874887	0.15/13.23(1700)	495/16.61(158)
poisson3Da	13514 / 352762	0.04/0.13 (24)	56.3/0.58 (11)

Матрица	P_{AISM} (GPU/GPU)		P_{AISM} (2GPU/GPU)
cdde5	0.56/0.13 (179)	0.57/0.13 (167)	0.76/0.13 (165)
ex37	15.66/0.004 (5)	15.44/0.004 (4)	8.76/0.004 (4)
rajat03	169/0.07 (84)	169/0.11 (135)	79.08/0.28 (333)
flowmeter5	357.6/0.11 (120)	357.9/0.15 (112)	165.41/0.14 (105)
ex19	700/0.35 (132)	703/0.39 (128)	325.96/0.44 (127)
sme3Da	814/11.83(2032)	843/13.5(1338)	389.113/11.5 (1005)
poisson3Da	1049/0.05 (28)	1066/0.24 (30)	495.475/0.24 (30)

Задача деформирования сеток

Основной идеей метода является поиск необходимого интерполянта в виде линейной комбинации, образованной радиальными базисными функциями³. В общем случае, для задачи деформирования Ω_g^h , интерполянт запишется следующим образом:

$$w_i(\mathbf{x}) = \sum_{j=1}^{n_\Gamma} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + v(x), \quad (3)$$

здесь $\mathbf{x} = (x_j, y_j, z_j)$, коэффициенты λ_j и полиномы v определяются из условия $w_i(\mathbf{x}_j) = \mathbf{W}_j^i$ и требования $\sum_{j=1}^{n_\Gamma} \lambda_j b(\mathbf{x}_j) = 0$ для всех полиномов b степени меньше или равной степени полинома v .

Пусть $\hat{\mathbf{u}}_g$ перемещения узлов границы Γ_g^h . Тогда система уравнений в матричной форме для неизвестных коэффициентов λ и μ :

$$\begin{bmatrix} \Phi_\Gamma & \mathbf{B}_\Gamma \\ \mathbf{B}_\Gamma^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{u}}_g \\ \mathbf{0} \end{bmatrix}, \quad \text{где } \Phi_\Gamma = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad \mathbf{B}_\Gamma = [1 \ x_j \ y_j \ z_j]. \quad (4)$$

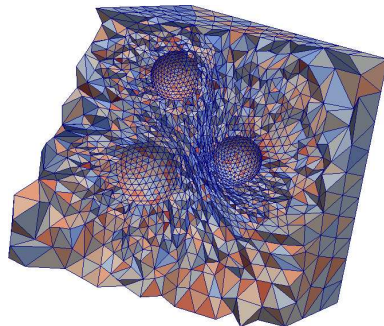
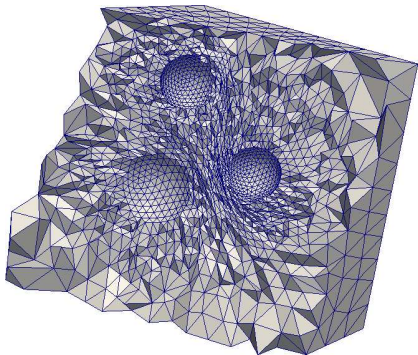
³Schaback R. Creating Surfaces from Scattered Data Using Radial Basis Functions // Mathematical Methods in Computer Aided Geometric Design III. M. Dohlen, T. Lyche and L.L. Schumaker (eds.). Vanderbilt Univ. Press, 1995. P. 477–496.

Функции радиального базиса

Функция	Вид	Функция	Вид
F1. Гауссиан	$e^{-\ \mathbf{x}\ }$	F5. Сплайн тон-	
F2.	$e^{-\ \mathbf{x}\ ^2}$	кой пластины	$\ \mathbf{x}\ ^2 \log \ \mathbf{x}\ $
F3. Мульти-		F6. Вы	$1 - \ \mathbf{x}\ $
квадратичная	$\sqrt{1 + \ \mathbf{x}\ ^2}$	F7. Вы	$(1 - \ \mathbf{x}\)^3 * (1 + 3 * \ \mathbf{x}\ ^3)$
F4. Обратная		F8. Вы	$(1 - \ \mathbf{x}\)^5 * (1 + 5 * \ \mathbf{x}\ +$
мульти-	$1/(\sqrt{1 + \ \mathbf{x}\ ^2})$		$+ 9 * \ \mathbf{x}\ ^2 + 5 * \ \mathbf{x}\ ^3 + \ \mathbf{x}\ ^4)$
квадратичная		F9. Вендланда	$(1 - \ \mathbf{x}\)^2$

Число итераций		Число итераций		Число итераций	
F1	15	F4	99	F7	4972
	12		122		5221
	14		76		8156
F2	—	F5	154	F8	351
	—		112		317
	—		140		224
F3	—	F6	7867	F9	11
	—		5599		9
	—		5294		10

Результат деформирования



Выводы

- Возможность использования как для симметричных так и для несимметричных систем
- Возможность использования как для разреженных так и для заполненных матриц
- При использовании BiCGStab, время выполнения одной итерации с AISM меньше, чем с ILU
- Модификация алгоритма для симметричных систем
- Не смотря на то, что формирование AISM медленнее распрастранённых предобуславливателей, он позволяет решать структурно-независимые системы
- Независимое формирование матриц S и T даёт ускорение в 1,8 раза