

高精度曲面建模的一种快速算法

赵娜, 岳天祥*

(中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室, 北京 100101)

摘要: 高精度曲面建模(HASM)是一种全新的曲面建模方法,其整个过程可分为偏微分方程的离散、采样方程建立和代数方程组求解3个阶段。目前所采用的求解对称正定方程组的方法主要是共轭梯度法。为了解决 HASM 的计算速度问题,本文给出了2种新的预处理共轭梯度算法,分别为不完全 Cholesky 分解共轭梯度法和对称逐步超松弛预处理共轭梯度法。实验表明,不完全 Cholesky 分解共轭梯度法收敛速度最快,且这2种预处理方法均比其他方法具有更快的收敛速度。

关键词: 预处理共轭梯度法;不完全 cholesky 分解;对称逐步超松弛算法;HASM

DOI: 10.3724/SP.J.1047.2012.00281

1 引言

高精度曲面建模理论(HASM)根据曲面论基本定理,结合高斯-科达齐方程,对所模拟的区域进行均匀正交剖分建立数值方程,是一种全新的曲面建模方法。其在采样数据的约束下,对正交剖分的均匀格网点进行求解,从而获得高精度的拟合曲面^[1]。数值实证表明,HASM 的计算精度高于以往的建模方法^[2]。

为了解决高精度曲面建模方法的速度问题和超大计算量问题,在大量数值实验的基础上,岳天祥等人给出了 HASM 的最佳表达形式^[3-4]。HASM 的计算速度及模拟精度与以往相比有了大幅度的提高^[4]。HASM 计算的整个过程大致可分为3个阶段:HASM 方程组系数矩阵的形成、采样方程的建立和代数方程组的求解。

为了进一步解决 HASM 模型的计算速度问题,我们对模型中的线性方程组,采用不同的预处理共轭梯度法,即不完全 Cholesky 分解预共轭梯度法 (ICCG, Incomplete Cholesky Decomposition Conjugate Gradient Method) 和对称逐步超松弛预

共轭梯度法 (SSORCG, Symmetric Successive Over Relaxation-preconditioned Conjugate Gradient Method) 求解,试验结果表明:与以往预处理方法相比,这两种方法均大大降低了 CPU 时间与收敛到一定精度解的迭代步数。同时,考虑了稀疏矩阵的压缩存储方式,这对大规模问题而言,解决了以往预处理共轭梯度算法中的内存不足等问题。

2 预处理共轭梯度法

设 HASM 模型中的方程组为:

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

其中, $\mathbf{A} \in \mathbf{R}^{n \times n}$ 为对称正定稀疏矩阵。

对于方程组(1)的求解,分为直接法和迭代法。直接法针对中小规模的问题,典型的直接法为高斯消元法,但此方法的时间复杂度与模拟区域网格数的三次方成正比^[5]。在迭代法中,共轭梯度(CG)方法^[6]被广泛用于求解大型稀疏对称正定线性系统。对于大型稀疏对称正定系统 $\mathbf{Ax} = \mathbf{b}$, 共轭梯度法(CG)在理论上具有有限步收敛性^[7]。特别是当系数矩阵的绝大部分特征值在1附近或者该问题良

收稿日期: 2011-11-07; 修回日期: 2012-04-18.

基金项目: 国家杰出青年科学基金项目(40825003); 全球变化研究国家重大科学研究计划项目课题(2010CB950904); 国家自然科学基金重点项目(41023010)。

作者简介: 赵娜(1986-), 女, 山东莱芜人, 博士研究生。主要研究方向: 生态模型与系统模拟、气候变化的模拟与研究。
E-mail: zhaon@lreis.ac.cn

* 通讯作者: 岳天祥(1963-), 男, 甘肃庆阳人, 博士生导师, 研究员。主要研究方向: 资源环境模型与系统模拟。
E-mail: yue@lreis.ac.cn

态时,CG 方法的收敛速度很快。即 CG 方法的收敛速度依赖于系数矩阵 \mathbf{A} 的条件数或更一般地依赖于 \mathbf{A} 的特征值的分布。但在实际问题中,由于舍入误差的影响,以往经验表明,为了达到比较准确的解,CG 方法的收敛步数往往远远大于系数矩阵的阶数甚至不收敛。而对于 HASM 模型,此方程组的系数矩阵条件数较大。因此,在实际计算前,先通过改善系数矩阵的条件数来提高 CG 方法的收敛速度,即预处理共轭梯度法。预处理共轭梯度法的思想就是把共轭梯度法用到预处理后的代数系统。

预处理共轭梯度法算法如下^[7]:

初始化, $\mathbf{x}_0, \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

while($\mathbf{r}_k \neq 0$)

 solve $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$

$k = k + 1$

 If $k = 1$

$\mathbf{p}_1 = \mathbf{z}_0$

 else

$\beta_k = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1} / \mathbf{r}_{k-2}^T \mathbf{z}_{k-2}$

$\mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k \mathbf{p}_{k-1}$

 end

$\alpha_k = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1} / \mathbf{p}_k^T \mathbf{A} \mathbf{p}_k$

$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k$

$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{A} \mathbf{p}_k$

end

$\mathbf{x} = \mathbf{x}_k$

该算法中,主要的计算量来自于求解方程组 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 。因此,下面我们寻找预处理矩阵 \mathbf{M} ,一方面使得该预处理方法的收敛速度很快,另一方面使求解方程组 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 的计算量尽可能小。

2.1 不完全 Cholesky 分解预处理方法

将方程组(1)转化为易于求解的方程组:

$$\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}}, \tag{2}$$

其中, $\bar{\mathbf{A}} = \mathbf{M}\mathbf{A}, \bar{\mathbf{b}} = \mathbf{M}\mathbf{b}$ 。

若 $k(\bar{\mathbf{A}}) \ll k(\mathbf{A})$ ($k(\mathbf{A})$ 表示矩阵 \mathbf{A} 的条件数), 则同样的算法求解式(2)比求解式(1)具有更快的收敛速度。

首先,我们采用的预处理方法为 \mathbf{A} 的不完全 Cholesky 分解法^[8]。

设 \mathbf{A} 的不完全 Cholesky 分解为 $\mathbf{A} = \mathbf{M} + \mathbf{R} = \mathbf{L}\mathbf{L}^T + \mathbf{R}$,

其中, \mathbf{L} 是下三角阵,使 $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ 尽可能接近

\mathbf{A} ,且 \mathbf{L} 保持跟 \mathbf{A} 一样的稀疏性或具有其他指定的稀疏性。完全 Cholesky 分解是对系数矩阵 \mathbf{A} 进行三角分解 $\mathbf{A} = \mathbf{L}\mathbf{L}^T$,不完全分解是对矩阵 $\mathbf{A} - \mathbf{R}$ 进行三角分解 $\mathbf{L}\mathbf{L}^T$ 。由于矩阵 \mathbf{R} 可以变化,因此, \mathbf{L} 的稀疏性结构可以预先适当控制,即 \mathbf{L} 中哪些元素为 0,可以预先规定,同时还要考虑到 $\mathbf{L}\mathbf{L}^T$ 要尽可能地接近 \mathbf{A} 。这样克服了完全 Cholesky 分解破坏 \mathbf{A} 的稀疏性的缺点。在实际计算中,常常考虑 \mathbf{R} 有较多的零元素,且 \mathbf{R} 元素不应太大。本文我们考虑没有填充的 Cholesky 分解算法,即对 \mathbf{A} 进行 Cholesky 分解时,对应于 \mathbf{A} 的零元处的位置,在分解过程中不再引入非零元。下面给出此算法的代码^[7]:

for $k = 1:n$

$\mathbf{A}(k,k) = \sqrt{\mathbf{A}(k,k)}$

 for $i = k+1:n$

 if $\mathbf{A}(i,k) \neq 0$

$\mathbf{A}(i,k) = \mathbf{A}(i,k) / \mathbf{A}(k,k)$

 end

 end

 for $j = k+1:n$

 for $i = j:n$

 if $\mathbf{A}(i,j) \neq 0$

$\mathbf{A}(i,j) = \mathbf{A}(i,j) - \mathbf{A}(i,k)\mathbf{A}(j,k)$

 end

 end

 end

end

以 $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ 作为预处理算子,应用共轭梯度法,不难验证预处理后的系数矩阵 $\bar{\mathbf{A}} \approx \mathbf{I}$ 。在具体实现过程中, \mathbf{A} 采用行压缩存储, $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 可转化为求解 $\mathbf{L}^T \mathbf{z}_k = \mathbf{y}_k$, 及 $\mathbf{L}\mathbf{y}_k = \mathbf{r}_k$ 。这样,解方程组的计算量为 $O(n^2)$,比直接求解 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 的计算量提高了一个数量级。ICCG 方法对小型问题是有效的,针对大型问题,我们可以预先对系数矩阵进行 Cholesky 分解,将分解后的下三角矩阵参与 HASM 模型的计算,作为 HASM 模型的输入参数。该方法的空间复杂度与时间复杂度均为 $O(n^2)$ 。

2.2 对称逐步超松弛预处理方法

HASM 的主要计量来自于矩阵的乘积运算和求逆,且求逆运算所花费的时间远远大于矩阵的乘积。上述的不完全 Cholesky 分解预处理方法

中每次内迭代都需要求解方程组 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 。陈传法等^[9]采用了改进的 Gauss-Seidel 算法对 HASM 进行求解。Gauss-Seidel 法的收敛性问题受到 HASM 方程中稀疏矩阵的制约。对称逐步超松弛 (SSOR) 方法是 Gauss-Seidel 的一种加速方法,是求解大型稀疏矩阵方程组的有效方法,它具有计算公式简单,程序设计容易,占用计算机内存少等优点。同时,该预处理方法可通过直接计算系数矩阵的近似逆矩阵,以避免每次迭代求解方程组 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$,从而减少计算量。该方法中只涉及矩阵与向量的乘积,且很容易实现并行运算。

应用对称逐次超松弛方法 (SSOR)^[10],设系数矩阵 \mathbf{A} 可以分解为 $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$,其中 \mathbf{D} 为 \mathbf{A} 的对角线元素构成的对角阵, \mathbf{L} 为 \mathbf{A} 的下三角部分构成的下三角矩阵。定义 SSOR 预处理算子如下^[11]:

$$\mathbf{M} = \mathbf{K}\mathbf{K}^T,$$

$$\text{其中, } \mathbf{K} = \frac{1}{\sqrt{2-\omega}} \left(\frac{1}{\omega} \mathbf{D} + \mathbf{L} \right) \left(\frac{1}{\omega} \mathbf{D} \right)^{-\frac{1}{2}}, 0 < \omega < 2 \quad (3)$$

下面对 \mathbf{K} 的逆矩阵作进一步的估计,由 (3) 式得,

$$\mathbf{K} = \frac{1}{\sqrt{2-\omega}} \left(\frac{1}{\omega} \mathbf{D} \right) (I + \omega \mathbf{D}^{-1} \mathbf{L}) \left(\frac{1}{\omega} \mathbf{D} \right)^{-\frac{1}{2}},$$

$$\text{则 } \mathbf{K}^{-1} = \sqrt{2-\omega} \left(\left(\frac{1}{\omega} \mathbf{D} \right) \right)^{\frac{1}{2}} (I + \omega \mathbf{D}^{-1} \mathbf{L})^{-1} \left(\frac{1}{\omega} \mathbf{D} \right)^{-1},$$

记 $\frac{1}{\omega} \mathbf{D} = \bar{\mathbf{D}}$, 由于 $(I + \bar{\mathbf{D}}^{-1} \mathbf{L})^{-1} = I - \bar{\mathbf{D}}^{-1} \mathbf{L} + (\bar{\mathbf{D}}^{-1} \mathbf{L})^2 - \dots$, 去掉该展开式中的高阶项,得

$$\begin{aligned} \mathbf{K}^{-1} &= \sqrt{2-\omega} (\bar{\mathbf{D}})^{\frac{1}{2}} (I + \bar{\mathbf{D}}^{-1} \mathbf{L}) (\bar{\mathbf{D}})^{-1} \\ &\approx \sqrt{2-\omega} (\bar{\mathbf{D}})^{\frac{1}{2}} (I - \bar{\mathbf{D}}^{-1} \mathbf{L}) (\bar{\mathbf{D}})^{-1} \\ &= \sqrt{2-\omega} ((\bar{\mathbf{D}})^{\frac{1}{2}} - (\bar{\mathbf{D}})^{\frac{1}{2}} \mathbf{L} \bar{\mathbf{D}}^{-1}) \\ &= \sqrt{2-\omega} (\bar{\mathbf{D}})^{-\frac{1}{2}} (I - \mathbf{L} \bar{\mathbf{D}}^{-1}) \equiv \bar{\mathbf{K}} \end{aligned}$$

则 \mathbf{A} 的近似逆可表示为

$$\bar{\mathbf{M}} = \bar{\mathbf{K}}^T \bar{\mathbf{K}}.$$

从而使得预处理共轭梯度法中每次内迭代求解方程组 $\mathbf{M}\mathbf{z}_k = \mathbf{r}_k$ 转化为计算 $\mathbf{z}_k = \bar{\mathbf{M}}\mathbf{r}_k$, 进而减少了计算量。但矩阵 $\bar{\mathbf{M}}$ 的稀疏性要比 \mathbf{A} 的稀疏性差。在具体计算过程中,把计算 $\mathbf{z}_k = \bar{\mathbf{M}}\mathbf{r}_k$ 转化为计算 $\mathbf{y}_k = \bar{\mathbf{K}}\mathbf{r}_k$ 和 $\mathbf{z}_k = \bar{\mathbf{K}}^T \mathbf{y}_k$ 。这样保证了 $\bar{\mathbf{K}}$ 与 \mathbf{A} 有相同的稀疏结构,从而减少了内存占用率。从上述分析可以看出,在程序实现过程中,预处理矩阵 $\bar{\mathbf{K}}$ 可以显式地实现,从而使得 SSORCG 的空间复杂度为 $O(n)$ 。

3 高斯合成曲面的数值试验

该试验以高斯合成曲面为研究对象,以 Dell-OptiPlex 990 为计算机实验平台验证不同的预处理共轭梯度方法对 HASM 收敛速度的影响。

高斯合成曲面(图 1)的数学表达式为:

$$f(x, y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10(x/5 - x^3 - y^5) e^{-x^2-y^2} - e^{-(x+1)^2-y^2}/3$$

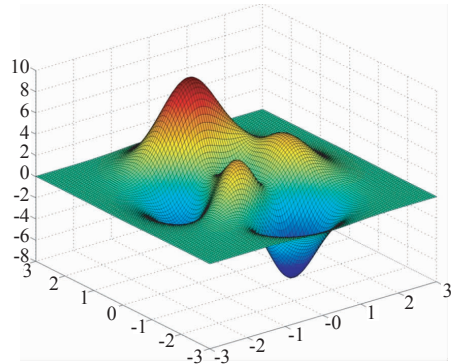


图 1 高斯合成曲面

Fig. 1 Gaussian synthetic surface

假定研究区域为 $[-3, 3] \times [-3, 3]$, 此时, $-6.5510 < f(x, y) < 8.1062$ 。设计了 3 组试验,分别改变模拟区域网格点数,内外迭代次数以比较不同方法的计算效率。

试验 1 为固定采样间隔 ($m=4$), 外迭代停止准则 $\|f(n+1) - f(n)\| / \|f(n)\| \leq 1e-6$, 内迭代停止准则为 $\|b - A * x\| / \|b - A * x_0\| \leq 1e-9$, 改变模拟区域的网格数(分别为 101×101 , 301×301 , 501×501 , 1001×1001) 比较不同预处理共轭梯度法的计算时间及停止迭代时的内迭代步数。同时给出了 ICCG 法与对角线预处理共轭梯度法的计算时间的差值与网格数的关系。试验结果记录在表 1 和图 2 中。表 1 中, S 表示对角线预处理共轭梯度法, T 表示三对角预处理共轭梯度法。由表 1 和图 2 表明, ICCG 法及 SSORCG 法的计算时间及内迭代步数均小于对角线预处理方法及三对角预处理方法; 且 ICCG 法与对角线预处理方法的计算时间的差值随着网格数的增多呈现较好的线性关系, 即网格数越多, ICCG 法的优势越明显, 两种方法的时间差值与网格数的关系为:

$$gn = -2.3729 \times 10^3 t^2 + 1.5062 \times 10^5 t$$

+ 1.5211 × 10⁴,

其中, t 为计算时间差值, gn 为模拟区域的网格数目。试验结果表明, ICCG 法与 SSORCG 法不论在计算时间上还是迭代步数上, 较以往的对角线预处理方法与三对角预处理方法均有显著提高。从图 2 中可见, 随着模拟网格数的增多, ICCG 法的计算时间越少其优势越明显。

试验 2 为固定采样间隔($m=4$), 外迭代次数设为 5 次, 模拟区域网格 101×101 , 比较不同方法的

收敛精度。试验 3 为固定采样间隔($m=4$), 设内迭代次数为 10 次, 模拟区域网格数为 501×501 , 比较不同的外迭代次数(分别为 2、4、6、8 次)下各种方法的收敛精度。试验结果如表 2、3 所示。

综合以上试验可得, ICCG 法收敛速度最快, SSORCG 法次之, 二者均比对角线预处理法、三对角预处理法收敛快; 且随着网格数的增多, ICCG 法的优势越明显。

表 1 不同预处理方法的计算效率的比较
Tab. 1 Comparison of computational efficiency with different pre-processing

网格数	计算时间(秒)				时间差(秒)		内迭代次数			
	ICCG	SSORCG	S	T	S-ICCG	ICCG	SSORCG	S	T	
101×101	0.0797	0.0922	0.1178	0.2418	0.0381	23	42	74	66	
301×301	0.6826	0.7896	1.0882	1.7557	0.4056	23	38	69	61	
501×501	1.9908	2.4962	3.6256	5.2110	1.6348	22	37	69	59	
1001×1001	8.1942	10.4086	15.6115	21.3498	7.4173	22	37	69	59	

表 2 不同内迭代次数各种方法的精度比较
Tab. 2 Comparison of simulation accuracy with different inner-iteration numbers

内迭代次数	ICCG	SSORCG	S	T
5	2.3482	7.4335	6.4740	7.9532
10	0.0395	0.8129	4.7385	1.9689
20	0.000 026	0.0145	0.4666	0.0876
50	0	0	0.000 513	0.000 044

表 3 不同外迭代次数, 各种方法的精度比较
Tab. 3 Comparison of simulation accuracy with different outer-iteration numbers

外迭代次数	ICCG	SSORCG	S	T
2	0.001163	0.017035	0.113418	0.094594
4	0.000138	0.002427	0.013710	0.007262
6	0.000064	0.001169	0.006030	0.003271
8	0.000042	0.000821	0.007483	0.002054

4 结论

HASM 模型由于其收敛速度问题, 严重制约了其推广及应用。本文针对该模型的病态性等特点, 在共轭梯度法的基础上, 分别采用不完全 Cholesky 分解预处理方法及对称逐步超松弛预处理方法对系数矩阵进行预处理。在具体实现过程中, 充分考虑了两种不同的预处理方法, 在计算量与存储量方

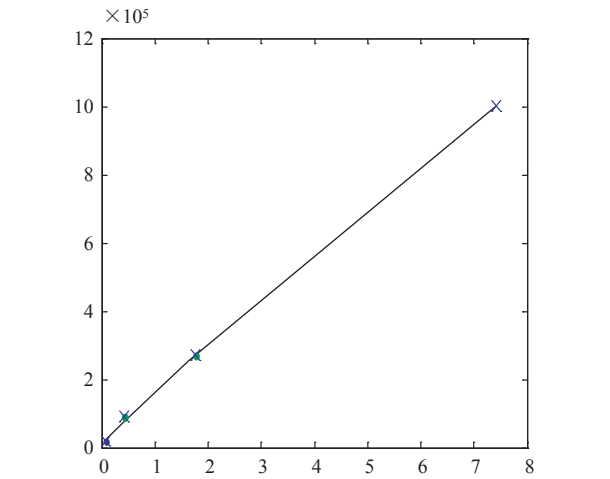


图 2 S 与 ICCG 时间差值与网格数的关系
Fig. 2 Relationship between time interval (S method and ICCG method) and the number of grid

面的实现细节。以高斯合成曲面为研究对象, 通过数值试验, 在 Dell-Optiplex 990 MT 机器上验证了两种预处理方法的有效性。试验结果表明, 所采用的两种预处理方法比以往的对角线预处理法及三对角预处理方法在收敛速度上均有明显的改善, 进而解决了 HASM 推广使用中的计算速度问题。

参考文献:
[1] 岳天祥, 杜正平, 刘纪远. 高精度曲面建模与误差分析[J]. 自然科学进展, 2004, 14(2): 83-89.

- [2] 岳天祥,杜正平. 高精度曲面建模与经典模型的误差比较分析[J]. 自然科学进展, 2007, 16(8): 986 - 991.
- [3] 岳天祥,杜正平. 高精度曲面建模最佳表达形式的数值实验分析[J]. 地球信息科学, 2006, 8(3): 83 - 87.
- [4] Yue T X. Surface modeling: High accuracy and high speed methods[J]. CRC Press, 2010, 39 - 63.
- [5] Davis T J. Direct methods for Sparse Linear Systems [J]. SIAM Philadelphia. 2006, 85 - 94.
- [6] Hestenes M R, Stiefel E F. Methods of conjugate gradients for solving linear systems[J]. J. Res. Nat. Bur. Stand. 1952, 49: 409 - 436.
- [7] Golub G H, Van Loan C F. Matrix computations[J]. Posts & Telecompress. 2009, 530 - 535.
- [8] Meijerink J A, Vorst van der H A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix[J]. Math. Comp. 1977(31):148 - 162.
- [9] 陈传法,岳天祥,张照杰. 高精度曲面模型的解算[J]. 武汉大学学报(信息科学版), 2010, 35(3): 365 - 368.
- [10] Evans D J, Forrington C V D. An iterative process for optimizing symmetric successive over-relaxation [J]. The Computer Journal, 1963, 6(3): 271 - 273.
- [11] Helfenstein R, Koko J. Parallel preconditioned conjugate gradient algorithm on GPU[J]. Journal of Computational and Applied Mathematics, 2012, 236 (15): 3584 - 3590.

Fast Methods for High Accuracy Surface Modeling

ZHAO Na and YUE Tianxiang

(State Key Laboratory of Resources and Environment Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101, China)

Abstract: As a novel surface modeling method, the whole computing process of HASM can be divided into three parts: deriving finite difference approximations to differential equations, establishing the sampling point equations and solving the algebra equations. The well known conjugate gradient method has been used extensively to solve this symmetric positive definite system instead of Gaussian's elimination based direct methods, especially for very large systems when parallel solution environment is preferred. However, a difficulty associated with the method of conjugate gradients is that it works well on matrices that are either well conditioned or have just a few distinct eigenvalues and the coefficient matrix of the algebra equation in HASM is ill-conditioned. In this paper, we show how to preprocess a linear system so that the matrix of coefficients assumes one of these nice forms. We give two other preconditioners, i. e. incomplete Cholesky decomposition conjugate gradient method (ICCG) and symmetric successive over relaxation-preconditioned conjugate gradient method (SSORCG), so as to improve the convergence rate of HASM. Furthermore, we give adequate consideration in storage scheme of the large sparse matrix and optimize the performance of sparse matrix-vector multiplication. The cost of the computation is also considered in each iteration. We implement and test the proposed method on a Dell OptiPlex 990MT machine. Numerical tests show that ICCG has the fastest convergence rate of HASM. We also find that both ICCG and SSORCG have much faster convergence rates than others.

Key words: preconditioned conjugate gradients method; incomplete Cholesky factorization; successive over-relaxation algorithm; HASM