

OpenMP 并行算法在卫星重力场模型反演中的应用^{*}

周 浩¹⁾ 钟 波^{1,2)} 罗志才^{1,2)} 张 坤¹⁾

(¹⁾武汉大学测绘学院,武汉 430079
(²⁾地球空间环境与大地测量教育部重点实验室,武汉 430079)

摘 要 利用卫星重力数据反演地球重力场需要解决重力场模型的高效计算问题。分析了最小二乘直接法求解重力场模型涉及的密集型计算任务,基于 OpenMP 实现了卫星重力场模型直接求解的并行算法。利用 30 天、5 秒采样间隔的沿轨扰动位 T 和径向扰动重力梯度 T_{rr} 数据,分别反演了 60 阶次的卫星重力场模型,计算结果表明,OpenMP 并行算法能够有效提高直接法求解卫星重力场模型的计算效率,并具有很好的稳定性。

关键词 并行算法;重力场模型;卫星重力;重力梯度;OpenMP

中图分类号:P223, P228 文献标志码:A

APPLICATION OF PARALLEL ALGORITHMS BASED ON OpenMP TO SATELLITE GRAVITY FIELD RECOVERY

Zhou Hao¹⁾, Zhong Bo^{1,2)}, Luo Zhicai^{1,2)} and Zhang Kun¹⁾

(¹⁾School of Geodesy and Geomatics, Wuhan University, Wuhan 430079
(²Key Lab. of Geospace Environment and Geodesy, Ministry of Education, Wuhan University, Wuhan 430079)

Abstract The massive satellite data presents a significant computational challenge to estimate the gravity field. Analyzing the computing-intensive tasks in recovering the earth gravity field model based on the direct least-squares algorithms, on applying parallel computing techniques capable of rigorously inverting geopotential coefficients using direct method on the basis of OpenMP is focused. Eventually, the earth's gravity fields complete to degree and order 60 are recovered respectively on the basis of monthly along track disturbing potential observations and gravity gradient observations (with 5 s sampling interval). The simulation shows that introducing the parallel algorithms based on OpenMP into determination of the earth's gravity field model can be more effective, and is of great reliability.

Key words: parallel algorithms; gravity field model; satellite gravimetry; gravity gradient; OpenMP

1 引言

卫星重力测量技术为解决全球高覆盖率、高空

间分辨率和高时间重复率的重力测量开辟了新途径,特别是新一代卫星重力计划的成功实施,已使得地球重力场模型的中长波部分精度提高了约 2 个量

* 收稿日期:2011-06-17

基金项目:国家自然科学基金(40874002,41104014);国家 863 计划项目(2008AA12Z105);测绘遥感信息工程国家重点实验室专项科研经费资助项目(2009);地球空间环境与大地测量教育部重点实验室开放基金(10-02-09)

作者简介:周浩,男,1987 年生,硕士研究生,主要从事卫星重力学研究. E-mail: lengyeshanren@126.com

级^[1-3]。

新一代卫星重力测量数据具有高采样率、全球观测的特点。利用卫星重力观测值反演地球重力场将面临海量数据的处理问题,特别是对于高阶重力场模型的反演,其计算量巨大,计算过程极为耗时。为了利用海量卫星重力观测值快速求解高精度的地球重力场模型,许多学者一方面是对现有重力场恢复方法进行改进,提出了一些快速解算法^[4-6];另一方面就是采用并行算法,即基于并行计算机,编写合适的并行算法来提高计算速度^[7-9]。本文对最小二乘直接求解过程中的密集型计算任务进行了分析,并基于 OpenMP 实现其并行求解,最后通过数值计算对其有效性进行了分析与验证。

2 重力场求解的并行算法^[9-14]

重力场最小二乘直接求解的并行算法的计算步骤如图 1 所示。从图 1 可以看出,构建设计矩阵 A 、构建法方程系数阵 N 和 b ,以及解算法方程 3 个部分为计算的“热点”,应该引入并行算法进行优化。下面将基于 OpenMP 实现各个计算“热点”的并行。

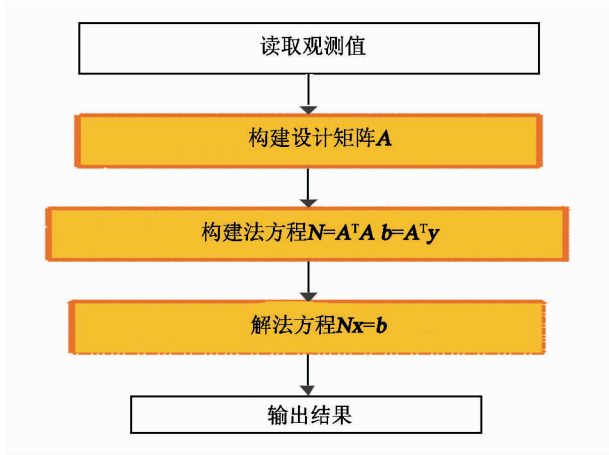


图 1 直接解法计算流程图

Fig. 1 Flow chart of the computation with direct method

2.1 构建设计矩阵 A 的并行

由于设计矩阵 A 的行与行之间无相关性,且每行的构建具有相似性,为此,程序中使用 BUILD_A_THREAD 子函数来构建矩阵 A 中的一行,然后根据 OpenMP 语法需求,加入组合并行共享任务制导语句! \$OMP PARALLEL DO/\$OMP END PARALLEL DO 实现构建设计矩阵 A 的并行。具体形式为:

```
! $OMP PARALLEL DO
DO I = 1, NOBS
CALL BUILD_A_THREAD(.....)
END DO
```

! \$OMP END PARALLEL DO

其中,NOBS 为观测值个数,即设计矩阵 A 的行数。

2.2 矩阵相乘的并行

由图 1 可知,根据最小二乘构建法方程包含两个矩阵相乘运算,均是计算的“热点”,有必要实现矩阵相乘的并行,以加速构建法方程。构建法方程可用直接法、分块法、斯特拉森法等矩阵相乘的并行算法。

1) 直接法

设矩阵 $A = (a_{ij})$ 为 $m \times n$ 矩阵,矩阵 $B = (b_{ij})$ 为 $n \times l$ 矩阵,其乘积 $C = AB = (c_{ij})$ 为 $m \times l$ 矩阵,其中:

$$C = AB = \sum_{k=1}^n A_{ik} B_{kj} \tag{1}$$

式(1)各次计算中都要用到 i, j, k ,所以在并行过程中,需要各个节点都有一份这些参量的拷贝。根据 OpenMP 语法需求,需要将 i, j, k 设置为 PRIVATE 属性,同时使用并行共享任务制导语句! \$OMP PARALLEL DO/\$OMP END PARALLEL DO 实现并行。具体形式为:

```
! $OMP PARALLEL
! $OMP DO PRIVATE(i,j,k)
DO i = 1, M
DO j = 1, L
C(i,j) = 0.0
DO k = 1, N
C(i,j) = C(i,j) + A(i,k) * B(k,j)
END DO
END DO
END DO
! $OMP END DO
! $OMP END PARALLEL
```

2) 分块法

行列分块法的具体计算过程是首先将矩阵 A 、 B 、 C 分解为:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{2}$$

然后由 A 、 B 阵中的分块相乘得到 C 阵中的对应分块,例如 $C_{11} = A_{11} B_{11} + A_{12} B_{21}$ 。至于分块矩阵乘法的并行,其处理方式与直接法相同。

3) 斯特拉森(Strassen)算法

利用此种方法主要是为了减少矩阵乘法中的运算量,降低时间复杂度。与分块法一样,首先按照式(2)将矩阵 A 、 B 、 C 进行行列分解,然后引入中间变量:

$$\begin{cases} \mathbf{M}_1 = (\mathbf{A}_{12} - \mathbf{A}_{22})(\mathbf{B}_{21} + \mathbf{B}_{22}) \\ \mathbf{M}_2 = (\mathbf{A}_{11} + \mathbf{A}_{22})(\mathbf{B}_{11} + \mathbf{B}_{22}) \\ \mathbf{M}_3 = (\mathbf{A}_{11} - \mathbf{A}_{21})(\mathbf{B}_{11} + \mathbf{B}_{12}) \\ \mathbf{M}_4 = (\mathbf{A}_{11} + \mathbf{A}_{12})\mathbf{B}_{22} \\ \mathbf{M}_5 = \mathbf{A}_{11}(\mathbf{B}_{12} - \mathbf{B}_{22}) \\ \mathbf{M}_6 = \mathbf{A}_{22}(\mathbf{B}_{21} - \mathbf{B}_{11}) \\ \mathbf{M}_7 = (\mathbf{A}_{21} + \mathbf{A}_{22})\mathbf{B}_{11} \end{cases} \quad (3)$$

利用中间变量相加得到最后结果:

$$\begin{cases} \mathbf{C}_{11} = \mathbf{M}_1 + \mathbf{M}_2 - \mathbf{M}_4 + \mathbf{M}_6 \\ \mathbf{C}_{12} = \mathbf{M}_4 + \mathbf{M}_5 \\ \mathbf{C}_{21} = \mathbf{M}_6 + \mathbf{M}_7 \\ \mathbf{C}_{22} = \mathbf{M}_2 - \mathbf{M}_3 + \mathbf{M}_5 - \mathbf{M}_7 \end{cases} \quad (4)$$

斯特拉森算法的实质是反复使用二阶矩阵乘法,使计算复杂度从 $O(n^3)$ 降为 $O(n^{2.81})$ 。为了保证在 \mathbf{M} 阵计算完成以后才计算 \mathbf{C} 阵,可以引入隐含并行同步语句! \$OMP BARRIER,具体形式为:

```
! $OMP PARALLEL
! $OMP DO PRIVATE(i,j,k)
实现公式(3)
! $OMP END DO
! $OMP BARRIER
! $OMP DO PRIVATE(i,j,k)
实现公式(4)
! $OMP END DO
! $OMP END PARALLEL
```

2.3 求解法方程的并行算法

根据计算流程,得到法方程中各个矩阵后,则可对法方程进行求解。并用直接法、平方根法和改进的平方根法实现相应的 OpenMP 并行算法。

1) 直接法

直接法分为矩阵求逆和矩阵相乘。其中,采用高斯-约当方法对矩阵求逆。由于高斯-约当法求逆是按照行序 i 进行的,故按照下面方式实现并行:

```
DO i = 1, U
处理对角线元素
! $OMP PARALLEL SHARED(N,i)
! $OMP DO PRIVATE(j)
! $OMP FLUSH(N)
处理主行元素
! $OMP END DO
! $OMP BARRIER

! $OMP DO PRIVATE(j,k)
! $OMP FLUSH(N)
处理非主行非主列元素
```

```
! $OMP END DO
! $OMP BARRIER

! $OMP DO PRIVATE(k)
! $OMP FLUSH(N)
处理主列元素
! $OMP END DO
! $OMP BARRIER
! $OMP END PARALLEL
END DO
```

其中, U 为法方程系数阵 \mathbf{N} 的维数。加入! \$OMP FLUSH 和! \$OMP BARRIER 语句,是为了保证数据的实时刷新与严格同步^[13]。

2) 平方根法^[15]

由于法方程系数阵 $\mathbf{N} = \mathbf{A}^T \mathbf{A}$ 为对称正定阵,故可以利用平方根法求解法方程。法方程系数阵 \mathbf{N} 可以作乔列斯基(Cholesky)分解,如果设定一次乘法和加法运算时间或一次除法运算时间为一个单位时间,则采用平方根方法的计算时间为 $(n^3 + 3n^2 + 2n)/6$,计算整个方程的时间为 $(n^3 + 9n^2 + 2n)/6$ 。

3) 改进的平方根法

改进的平方根法是针对 $\mathbf{N} = \overline{\mathbf{L}}\mathbf{L}^T$ 中的开方运算而提出来的,它主要是将平方根法中的 $\mathbf{N} = \overline{\mathbf{L}}\mathbf{L}^T = \mathbf{L}\mathbf{D}^{1/2}\mathbf{D}^{1/2}\mathbf{L}^T$ 转换为 $\mathbf{N} = \mathbf{L}\mathbf{D}\mathbf{L}^T$,相比平方根法可减少 n 次开方运算,提高了计算效率。采用 Fortran 语言对法方程系数阵作分解的算法:

```
! $OMP PARALLEL DO SHARED(N,j) num_
threads(U - j - 1)
DO j = 1, U
! $OMP PARALLEL DO SHARED(N, p, j)
num_threads(U - j - 1)
DO p = j + 1, U
DO q = j + 1, p
N(p,q) = N(p,q) - N(p,j) * N(q,j)/N(j,j)
END DO
END DO
! $OMP PARALLEL DO SHARED(N,j)
DO i = j + 1, U
N(i,j) = N(i,j)/N(j,j)
END DO
END DO
```

利用上述代码实现 \mathbf{N} 阵的分解后,利用简单的矩阵乘法即可获得法方程的解。

3 算例与分析

计算采用 Dell R900 共享内存并行计算机,处

理器为 4 颗 4 核的 Intel E7430,主频 2.13GHz,内存 64GB,编译环境为 Intel Visual Fortran 10.1。基于 GOCE 卫星模拟轨道数据,采用 60 阶次的 EIGEN - GL04C 模型分别模拟了沿轨 30 天、5 秒采样间隔 (总共 518 400 个观测数据)的扰动位 T 和径向扰动重力梯度 T_{zz} 数据,并进行重力场反演和精度比较分析。

3.1 设计矩阵计算与分析

首先设定不同的线程数来构建设计矩阵 A (维数为 518 400 × 3 721)。以 T_{zz} 数据计算为例,获得计算所需时间如表 1 所示。

表 1 构建设计矩阵 A 的时间分析 (单位:s)

Tab.1 Runtime of building design matrix A (unit:s)

线程数	1	2	4	8	16
计算时间	128.48	64.89	32.81	20.29	15.20
加速比	1	1.98	3.91	6.33	8.45

表 1 中,“加速比”是指在相同任务量下的并行执行时间与串行执行时间之比。若加速比与线程数的比例能够接近于 1,则表明该算法能够达到线性加速比,并行效率高^[9,14]。由表 1 中第 1 ~ 3 列可知,对构建设计矩阵 A 的并行基本上可达到线性加速比,由此可知构建设计矩阵 A 的并行计算效率较高;而第 4 列和第 5 列之所以未能够达到线性加速比,是因为本文采用的数据量较小,导致有些线程还没有参与计算就已结束了计算任务。

3.2 矩阵乘法的计算与分析

设定不同的线程数,采用直接法、分块法和 Strassen 法分别实现矩阵乘法的并行计算,数据计算结果如图 2 所示。

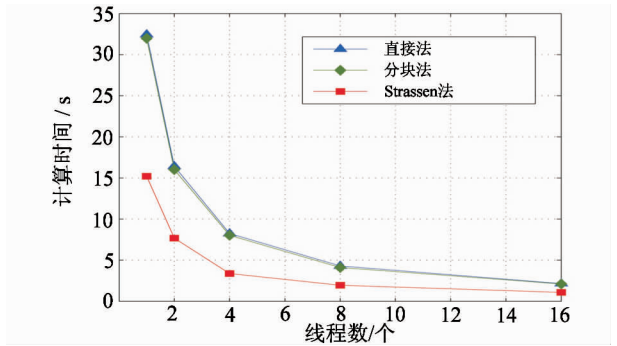


图 2 不同矩阵相乘的并行计算效率比较
Fig.2 Comparison among the efficiency of parallel methods for matrix multiplication

在评价并行算法时,我们通常假定任务量 W 一定,若随着线程数 p 的增加,计算所需时间 t 缩短,且近似满足 $t = W/p$,则该并行算法是成功的。由图

2 可知,3 种方法的并行效率都趋近于反比例函数,这说明 3 种矩阵乘法的并行算法均是成功的。

从图 2 还可知:在相同线程数、相同任务量的前提下,分块法计算时间略小于直接法,而 Strassen 法的计算时间则最小。

3.3 法方程解算与分析

采用直接法、平方根法和改进的平方根法,使用不同阶数的法方程阵,分别实现了法方程求解的并行算法(计算中设定 12 个线程数),计算结果如图 3 所示。

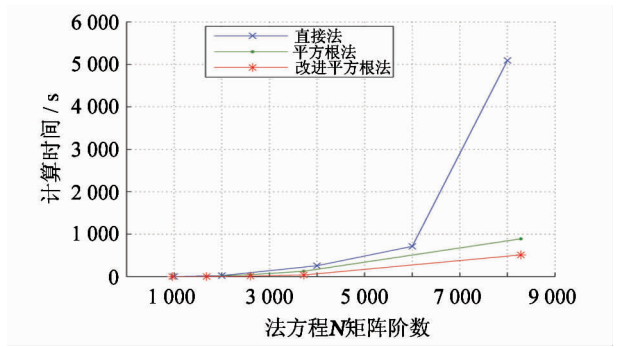


图 3 不同求解法方程方法的并行效率比较
Fig.3 Comparison among the parallel efficiency of different methods for solving the normal equation

从图 3 可以看出:在相同线程数的前提下,随着任务量的增加,计算所需时间迅速增加;在解算所需时间上,改进的平方根法相比于直接法和平方根法具有极大的优势,这说明改进的平方根法的时间复杂度最小。但为了获得更高精度的解算结果和精度评定的参数协因数阵,本文选用直接法对法方程进行求解。

3.4 重力场反演计算与分析

分别采用 518 400 个观测历元的扰动位 T 和二阶径向扰动重力梯度 T_{zz} 数据来反演 60 阶次的地球重力场模型,其中 3 个计算“热点”分别使用构建设计矩阵的并行算法、Strassen 矩阵相乘的并行算法、直接法求解法方程的并行算法,计算时间如表 2 所示。

表 2 反演卫星重力场模型计算时间表 (单位:s)
Tab.2 Computational runtime of gravity model inversion (unit:s)

观测值	计算过程				
	构建矩阵 A	构建矩阵 N	构建矩阵 b	求解法方程	总时间
$T(1)$	133.321	43 476.091	16.954	1 908.215	45 534.581
$T(12)$	17.255	13 745.325	2.063	200.344	13 964.987
$T_{zz}(1)$	134.332	43 467.234	16.999	1 909.232	45 527.797
$T_{zz}(12)$	17.794	13 513.134	2.062	200.323	13 733.313

从表 2 可知,在本文计算条件下,使用串行算法利用一个月的观测数据反演 60 阶的重力场模型,约

需要 12 小时,当采用 12 个线程的 OpenMP 计算时,计算时间可节约 8 个小时。若使用更长时间跨度的观测数据,理论上节省的计算时间将更长。

采用模型阶误差 RMS 分别对扰动位 T 和二阶径向扰动重力梯度 T_{zz} 数据反演的重力场模型精度进行评价:

$$RMS = \sqrt{\frac{1}{2n+1} \sum_{m=0}^n [(C_{nm} - C_{nm}^{mod})^2 + (S_{nm} - S_{nm}^{mod})^2]} \quad (5)$$

式中, C_{nm} 、 S_{nm} 为反演模型的位系数, C_{nm}^{mod} 、 S_{nm}^{mod} 为“真实”模型位系数(本文为 EIGEN-GL04C), 计算结果如图 4 所示。由图 4 可知, 无误差的扰动位 T 和二阶径向扰动重力梯度 T_{zz} 数据反演模型的阶误差 RMS 可达到 10^{-20} 量级, 该误差仅为模型误差和计算误差, 可以忽略。

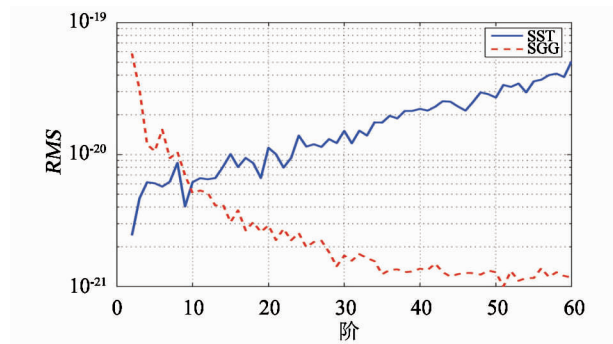


图 4 利用 T 和 T_{zz} 求解模型的阶误差 RMS

Fig. 4 Degree error RMS of different models for solving the T and T_{zz}

4 结束语

基于 OpenMP 实现了由最小二乘直接法求解卫星重力场模型的并行算法。通过数值计算分析表明:采用按行存储方式实现的构建设计矩阵的并行算法,基本可达到线性加速比;采用 Strassen 法实现的矩阵乘法并行计算能够减小计算的复杂度,极大提高计算效率;采用直接解法求解法方程的并行算法能提高计算效率,并可保证求解结果的准确性和可靠性。因此,基于 OpenMP 的并行算法是反演地球重力场模型的有效工具。由于 OpenMP 是基于共享存储系统上的并行编程标准,系统开销较大,将在后续工作中利用 OpenMP 与 MPI 的联合编程,实现更长时间序列数据反演超高阶次地球重力场的并行运算。

参 考 文 献

1 宁津生. 卫星重力探测技术与地球重力场研究[J]. 大地测量与地球动力学, 2002, (1): 1-5. (Ning Jinsheng. The satellite gravity surveying technology and research of Earth's

gravity field[J]. Journal of Geodesy and Geodynamics, 2002, (1): 1-5)

2 许厚泽. 卫星重力研究: 21 世纪大地测量研究的新热点[J]. 测绘科学, 2001, 26(3): 1-3. (Xu Houze. Satellite gravity mission-new hot point in Geodesy[J]. Science of Surveying and Mapping, 2001, 26(3): 1-3)

3 孙文科. 低轨道人造卫星(CHAMP、GRACE、GOCE)与高精度地球重力场——卫星重力大地测量的最新发展及其对地球科学的重大影响[J]. 大地测量与地球动力学, 2002, (1): 92-100. (Sun Wenke. Satellite in low orbit (CHAMP, GRACE, GOCE) and high precision Earth gravity field; the latest progress of satellite gravity geodesy and its great influence on geosciences[J]. Journal of Geodesy and Geodynamics, 2002, (1): 92-100)

4 Sanso F and Tscherning C C. Fast spherical collocation: theory and examples[J]. Journal of Geodesy, 2003, 77: 101-112.

5 Klees R, et al. Efficient gravity field recovery from GOCE gravity gradient observations[J]. Journal of Geodesy, 2000, 74: 561-571.

6 Sneeuw N. A semi-analytical approach to gravity field analysis from satellite observations [D]. University of München, 2000.

7 Klees R, et al. GOCE gravity field recovery using massive parallel computing [A]. Gravity, Geoid and Geodynamics 2000 [C]. F Sansò (eds.): IAG symposium, 2002, 123: 109-116.

8 Pail R and Plank G. Assessment of three numerical solution strategies for gravity field recovery from GOCE satellite gravity gradiometry implemented on a parallel platform[J]. Journal of Geodesy, 2002, 76: 462-474.

9 Wittwer T. An introduction to parallel programming [M/OL]. <http://www.vssd.nl/hlf/a019.htm>.

10 罗志才. 利用卫星重力梯度数据确定地球重力场的理论和方法[D]. 武汉测绘科技大学, 1996. (Luo Zhicai. The theory and methodology for determining the earth's gravity field using satellite gravity gradiometry data [D]. Wuhan Technical University of Surveying and Mapping, 1996)

11 王正涛. 卫星跟踪卫星测量确定地球重力场的理论与方法[D]. 武汉大学, 2005. (Wang Zhengtao. Theory and methodology of earth gravity field recovery by satellite-to-satellite tracking data [D]. Wuhan University, 2005)

12 钟波. 基于 GOCE 卫星重力测量技术确定地球重力场的研究[D]. 武汉大学, 2010. (Zhong Bo. Study on the determination of the earth's gravity field from satellite gravimetry mission GOCE [D]. Wuhan University, 2010)

13 Hermanns M. Parallel programming in Fortran 95 using OpenMP [M/OL]. http://www.openmp.org/presentations/miguel/F95_OpenMPv1_v2.pdf.

14 陈国良, 等. 并行算法实践[M]. 北京: 高等教育出版社, 2004. (Chen Guoliang, et al. Parallel algorithm practice [M]. Beijing: Higher Education Press, 2004)

15 朱方生, 李大美, 李素贞. 计算方法[M]. 武汉: 武汉大学出版社, 2003. (Zhu Fangsheng, Li Damei and Li Shuzhen. Numerical method [M]. Wuhan: Wuhan University Press, 2003)