

文章编号:1671-5942(2013)01-0141-04

基于 OPENMP 的高性能 GNSS 法方程解算方法研究*

杨 凯

(重庆市国土资源和房屋勘测规划院,重庆 400020)

摘 要 在大规模 GNSS 基准站网数据处理中,全网整体数据处理可以避免由于分子网解算而带来的子网划分、公共站选取等技术难点,在一定程度上维持了数学模型的严密性。针对大规模 GNSS 基准站网数据处理中存在的技术难点,给出了同时处理测站数量受限制(不能超过 100 个)的解决方法,比较了滤波算法与最小二乘估计算法的优劣,对比分析了现有法方程解算方法的效率,实现了基于 OPENMP 的并行乔里斯基分解法方程求逆方法,通过实际算例验证了其解算的高效率。

关键词 GNSS 基准站网;OPENMP;法方程解算;数据处理;测站数量

中图分类号:P207

文献标识码:A

RESEARCH ON HIGH PERFORMANCE GNSS NORMAL EQUATION PROCESSING METHOD BASED ON OPENMP

Yang Kai

(Chongqing Institute of Surveying and Planning for Land Resources and Housing, Chongqing 400020)

Abstract The whole network data processing of Large-Scale GNSS reference station network can avoid the technical problems such as sub-network division and common station selection of the sub-network processing strategy. The number of stations whose observation data to be processed should be less than 100. The least square adjustment and SRIF are compared with each other, and then we point out that the least square adjustment is better than the later for large data processing. Further, the high performance computing methods for normal equation are researched; the existing normal equation computing methods are compared with each other, and it is pointed out that the Cholesky matrix decomposition method is better in computing efficiency. At last, the parallel Cholesky matrix decomposition method based on OPENMP is achieved, which can take full advantage of multi-core/thread processor to improve the computing efficiency in personal computer platform.

Key words:GNSS reference station network;OPENMP; normal equation computing;data processing;number of stations

1 引言

目前,随着全球或区域范围内大规模 GNSS 基准站网的不断建设,基准站数目不断增加,如何高效地处理拥有数百、甚至上千个 GNSS 基准站的大规

模基准网数据,是一个亟需解决的重要问题。大规模 GNSS 基准站网数据处理策略目前主要分为 PPP 解、非差网解和双差网解模式等。全球公认的高精度数据处理软件包括 GIPSY、GAMIT、BERNESE、EPOS 等,其中 GAMIT 只能解算不超过 100 个站的

* 收稿日期:2012-07-12

基金项目:“十二五”国家科技支撑计划课题(2012BAJ22B06)

作者简介:杨凯,男,博士,主要从事高精度 GNSS 数据处理方面研究工作. E-mail: ykoky2k@sohu.com

基准站网数据^[1],BERNESE 和 EPOS 等虽然可以同时解算超过 100 个基准站的观测数据,但解算时间过长、占用的计算机硬件资源过多,限制了其在大规模 GNSS 基准站网数据处理中的应用^[2]。

针对上述技术难点,本文利用 IGS 全球连续跟踪站的实际数据,论述并给出了解决同时处理的测站数量不能超过 100 个的方法,比较分析了滤波算法与最小二乘估计算法的优劣,对比分析了现有方法方程解算方法的效率,实现了基于 OPENMP 的并行乔里斯基分解法方程求逆方法,并通过实际算例验证了其解算的高效率。

2 测站数量限制的解决方法

首先,在软件编写(仅考虑 FORTRAN 语言)时需要考虑内存溢出问题。当数组规模太大时很容易引起内存溢出从而导致编译失败。因此,在测站数量超过 100 个时,应按照 FORTRAN 95 格式编写代码,并使用 FORTRAN 95 标准中规定的动态数组。

其次,在 FORTRAN 语言中对于文件的访问(包括读/写)是通过所谓“文件号”进行的。若将同一个文件号赋予多个文件,会导致文件访问出错。特别是在大规模 GNSS 基准站网软件的编写中,访问的文件数目将达到数百甚至数千个,一旦出错会导致整个数据处理失败。对此需采用如下措施加以避免:1)将文件按类型分类,例如第一类可以是 IGS 精密星历文件、钟差文件、系统参数控制文件等,该类文件数目有限,可分配 10~100 的文件号;第二类文件可以是测站 RINEX 格式观测文件,该类文件数目较多,可以分配 200~2 000(上限值按基准站数目而定)的文件号;第三类文件可以是测站信息文件,例如包含了测站周跳、野值信息的 LOG 文件,可分配 3 000~5 000 的文件号。2)在每一类文件中,可自行编写一个获得空闲文件号的子程序对文件号加以自动分配,以防止文件号分配混乱。

3 现有参数估计方法比较分析

目前参数估计方法主要包括滤波算法和最小二乘估计算法两大类。滤波算法一般用以解决包含了状态参数的参数估计问题,具有实时输出状态参数及其方差,同时占用计算机内存资源较少的优点。滤波算法中的卡尔曼滤波^[3]在众多应用领域中得到了广泛的应用并出现了多种变体。特别的,在 GNSS 精密定轨定位领域,均方根信息滤波(Square-Root Information Filter,简称 SRIF)算法^[4],已在美国喷气动力实验室研发的高精度 GNSS 精密数据处理

软件 GIPSY 中得到应用^[5],在处理 GNSS 观测数据时能够有效避免滤波发散,具有较高的数值稳健性和高效性^[6,7]。

对于大规模 GNSS 基准站网的数据处理来说,虽然在解决动态误差参数估计的问题上滤波算法(特别是均方根信息滤波)具有非常优秀的表现,但 GNSS 事后精密定位所关心的是待估参数多为固定偏差(例如测站坐标),一般不关心中间过程不确定因素影响的状态参数(例如接收机钟差、对流层延迟参数、太阳光压参数等)求解。同时,滤波算法是逐历元求解一次法方程,SRIF 则需要逐历元进行信息矩阵的更新,当测站超过 100 个时,每历元进行一次法方程求解或信息矩阵更新,将耗费大量的计算时间。在 GIPSY 软件中,为了提高效率,将观测数据的采样间隔压缩为每 300 秒一个历元^[8]。

本文实现了利用 SRIF 算法进行参数估计,并采用 2009 年 5 月 2 日的 IGS 全球连续跟踪站网数据(采样率为 300s)分别在 50、100、150 和 200 站的不同测站数目情况下利用 SRIF 进行解算,各种情况下的测站分布如图 1 所示,得到的每历元 SRIF 解算所花费时间如图 2 所示。

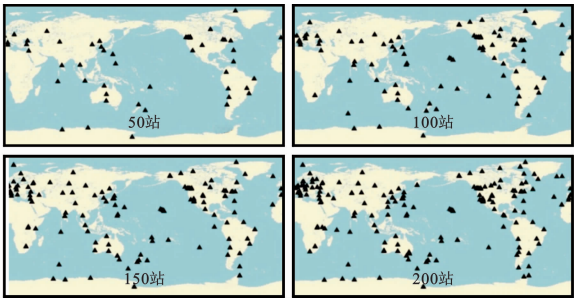


图 1 SRIF 单历元解算测站分布

Fig. 1 Site distribution in SRIF single epoch data processing

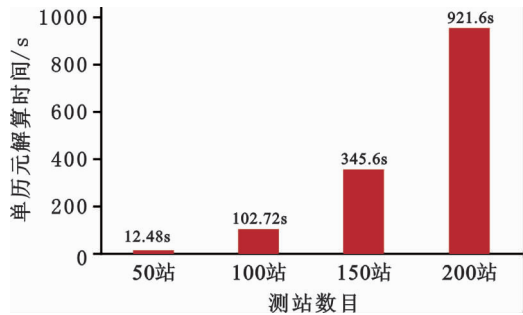


图 2 SRIF 单历元解算时间对比

Fig. 2 Comparison among time for SRIF single epoch data processing with different number of stations

由图 2 可知,随着测站数目的增加,SRIF 算法单历元解算时间增长非常迅速。当测站数目为 150 个时,单历元解算时间超过 5 分钟;而测站数目为 200 个时,单历元解算时间超过 15 分钟,由此推断,解算采样率为 300s 的全天观测数据时间将超过 72

小时。这对于大规模 GNSS 基准站网数据解算来说,是难以容忍的。

解算是在作者的个人笔记本电脑上运行的,计算机软、硬件平台信息如表 1 所示。

表 1 计算机软、硬件平台

Tab. 1 Computer hardware and software platform	
编程语言	FORTRAN 95
运行环境	LINUX
编译器	Intel Fortran 11.2 编译器
处理器	Intel I7-720QM(主频 1.6GHz)
内存	4G DDR3 1066
操作系统	OpenSuSE 11.3 64bit

另一方面,可以看到,由于最小二乘估计算法不需要逐历元求解法方程,只需在观测时段的每个历元进行法方程叠加,直到最后一个历元才显式求解 GNSS 精密定位所关心的确定性参数和状态参数,因此在计算效率上具有独特的优点。

4 现有法方程解算方法比较

法方程解算(主要是法方程求逆过程)是影响数据处理效率的关键因素之一。一般来说,对于 GNSS 精密数据处理,法方程中最多的参数是整周模糊度参数,其次为 ZTD 参数,二者占有参数的比例可能超过 90%。法方程求逆过程是一个标准的线性代数问题,可以用任意标准方法进行解算。对于大规模 GNSS 基准站网的高性能数据处理,需要找出最优法方程求逆算法。

另一方面,对于法方程求逆这样一个数值计算问题,不仅运用不同的方法求逆效率有差异,即便同一个方法、不同编程实现方式的解算效率同样可能相差较大。现有的高性能线性代数函数库,包括 BLAS、LAPACK、MKL 等,其运算效率已经被很多学者加以验证^[9]。因此,使用这些已有的函数库,可以安全可靠地让测绘科技人员将主要精力放在与 GNSS 相关的算法研究上,而不必在与计算机底层硬件相关的计算效率问题上过分纠缠。

为了研究分析适用于最小二乘估计方法的最优法方程解算方法,本文实现了利用高斯消元法、LU 分解法、Bunch-Kaufman 选主元法和乔里斯基分解法来解算法方程,并采用 2009 年 5 月 2 日的 IGS 全球连续跟踪站网数据(采样率为 300s)分别在 50、100、150 和 200 站的不同测站数目情况下进行解算,所用数据如图 1 所示,解算时的计算机软、硬件平台如表 1 所示,采用不同方法解算法方程的效率

如表 2 所示。

表 2 不同方法法方程求解时间对比(单位:s)

Tab. 2 Comparison among processing time with different normal equation methods(unit: s)

测站 数目	高斯 消元法	LU 分解法	B-K 选主元法	乔里斯 基分解法
50	542	28	23	16
100	4 573	231	167	111
150	16 136	794	566	391
200	28 925	1 416	1 050	694

由表 2 可知,对于最小二乘估计法方程的解算,高斯消元法耗时最多,接近 LU 分解法所用时间的 20 倍。LU 分解法、Bunch-Kaufman 选主元法和乔里斯基分解法所用时间基本在同一量级,乔里斯基分解法所用时间最少,约为 LU 分解法所用时间的一半。在 200 个测站情况下,乔里斯基分解法解算时间 694s,仅为同等情况下高斯消元法的 2.4%。

究其原因,乔里斯基分解法的高效率,来源于其与最小二乘估计法方程特征的匹配。最小二乘估计法方程的系数矩阵,是一个实正定对称矩阵,乔里斯基分解法可以充分利用其正定对称特性,而 LU 分解法(适用于所有可逆实矩阵)和 Bunch-Kaufman 选主元法(适用于所有可逆实对称)则不能充分利用这一特征,因此乔里斯基分解法的解算效率相对最高。

5 基于 OPENMP 的高性能并行计算

值得注意的是数据处理仅占用了计算机处理器的一个线程,即所谓的单线程解算。然而,目前的计算机技术已经进入多核处理器时代,Intel 和 AMD 公司面向个人消费者推出了众多款式的多核桌面/移动处理器。因此,在进行大规模 GNSS 基准站网的高性能数据处理时,应考虑同时使用多个线程进行并行计算,以提升解算效率。

一般来说,并行计算是由并行计算机来实现的,目前主要的并行计算机包括多计算机系统和集中式多处理器系统。并行算法在并行计算机上需要通过并行编程来实现,当前使用较多的并行编程环境主要有消息传递编程接口(Message Passing Interface,简称 MPI)以及共享存储编程接口(Open Multi-Processing,简称 OPENMP)两种^[10]。

其中,MPI 更适合于多台计算机节点组成的计算机集群系统,而 OPENMP 更适用于多个计算机内核/线程的单台计算机(节点)。本文暂不涉及基于 MPI 的并行计算。在笔者的个人笔记本计算机上实现了

基于 LAPACK/MKL 的 OPENMP 并行化计算,所用的计算机软、硬件平台如表 1 所示。其实现方式包括:1)在 Intel Fortran 11.2 编译器中打开-openmp 选项以支持 OPENMP 并行化计算;2)对上述四种最小二乘法方程解算函数源代码做适当修改以实现逻辑上的并行运算,具体步骤限于篇幅暂不赘述。

在将上文中的四种最小二乘法方程求解方法进行并行化改造之后,采用 2009 年 5 月 2 日的全球 IGS 连续运行跟踪站数据进行了算例分析,所用测站数目分别为 50 站、100 站、150 站与 200 站,采样率为 300s。测站分布如图 1 所示。乔里斯基分解法在并行化之后的解算效率如图 3 所示,解算时同时使用了四个线程进行并行计算。

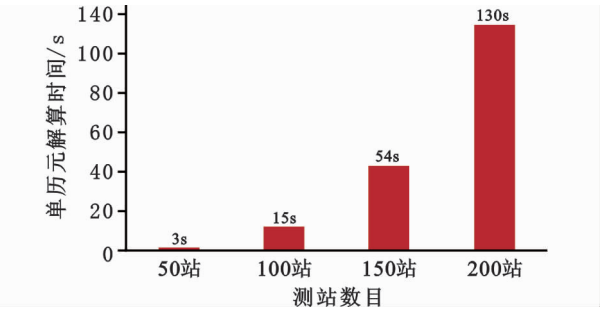


图 3 法方程解算时间对比图
Fig. 3 Comparison chart of normal equation computing time (unit:s)

结合表 2 和图 3 可知,在实现了并行化计算后,法方程解算效率提升超过 400%,在 200 个测站情况下,乔里斯基分解法求解法方程用时 130s,仅为串行计算时 694s 解算时间的 18%,可见并行计算对解算效率的巨大影响。

6 结论

- 1)对于大规模 GNSS 基准站网全网整体处理时,不能同时解算 100 个测站以上数据的问题可以通过编程技巧解决;
- 2)大规模 GNSS 基准站网的事后静态数据处理,采用最小二乘算法进行参数估计在解算效率上具有独特的优势;
- 3)法方程的解算,建议采用 BLAS/LAPACK/MKL 等高性能线性代数函数库以提升解算效率;

4)基于 OPENMP 的并行化乔里斯基分解方法,可充分利用多核/多线程处理器的优势进行法方程求逆解算,从而可以大幅提高个人计算机平台的解算效率。

参 考 文 献

1 Herring T A, King R W and McClusky S C. GAMIT reference manual [EB/OL]. 2009 [2011 - 02 - 10]. [http://www - gpsg. mit. edu/ ~ simon/ gtgk/ docs. htm](http://www-gpsg.mit.edu/~simon/gtgk/docs.htm).

2 Ge M, et al. A new data processing strategy for huge GNSS global networks[J]. Journal of Geodesy, 2006, 80 (4): 199 - 203.

3 Kalman R E. A new approach to linear filtering and prediction problems [J]. Journal of Basic Engineering, 1960, 82 (1): 35 - 45.

4 Bierman G J. The treatment of bias in the square-root information filter/smoothing [J]. Journal of Optimization Theory and Applications, 1975, 16(1): 165 - 178.

5 Webb F H and Zumbege J F. An Introduction to GIPsy/oasis-II [R]. JPL Publ D-11088, Jet Propul Lab., Pasadena, California, 1993.

6 赵齐乐,等. 均方根信息滤波和平滑及其在低轨卫星星载 GPS 精密定轨中的应用[J]. 武汉大学学报(信息科学版), 2006, (1): 12 - 15. (Zhao Qile, et al. Applications of square-root information filtering and smoothing on orbit determination of LEO satellites with on bord GPS data [J]. Geomatics and Information Science of Wuhan University, 2006, (1): 12 - 15.)

7 施闯,等. 卫星导航系统综合分析处理软件 PANDA 及研究进展[J]. 航天器工程, 2009, (4): 64 - 70. (Shi Chuang, et al. PANDA: Comprehensive processing software for satellite navigation systems and its research progress [J]. Spacecraft Engineering, 2009, (4): 64 - 70.)

8 Lichten S M. Towards GPS orbit accuracy of tens of centimeters [J]. Geophysical Research Letters, 1990, 17(3): 215 - 218.

9 K Gstr M B and Poromaa P. LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs [J]. ACM Transactions on Mathematical Software, 1996, 22 (1): 78 - 103.

10 Quinn M J. Parallel programming in C with MPI & openMP [M]. New York: McGraw-Hill Press, 2003.