引文格式:黄炎,李姗姗,吕明昊,等.利用CPU和GPU混合并行方法快速构建海洋扰动重力梯度基准图[J].武汉大学学报(信息科学版),2025,50(3):515-527.DOI:10.13203/j.whugis20220142

Citation: HUANG Yan, LI Shanshan, LV Minghao, et al. Rapid Construction of Marine Disturbing Gravity Gradient Reference Map by CPU and GPU Hybrid Parallel Method[J]. Geomatics and Information Science of Wuhan University, 2025, 50(3):515-527. DOI:10.13203/j.whugis20220142

利用CPU和GPU混合并行方法快速构建海洋扰动重力梯度基准图

黄 炎[1,2] 李姗姗[2] 吕明昊[3] 范 雕[2] 谭勖立[2] 冯进凯[2]

1 军事科学院国防科技创新研究院,北京,100071

2 信息工程大学地理空间信息学院,河南 郑州,450001

3 96941部队,北京,100080

摘 要:高精度、高分辨率的海洋扰动重力梯度基准图是将水下扰动重力梯度辅助惯性导航付诸于实践的关键技术之一,可依据边值问题理论,基于移去-恢复技术计算求得。基于传统串行算法存在计算效率低、耗时长等问题,为满足大范围乃至全球海洋扰动重力梯度基准图的快速构建需求,利用中央处理器(central processing unit, CPU)与图形处理器(graph processing unit, GPU)混合并行(CPU+GPU)编程平台,设计了一种高效的混合并行计算方案。首先,利用数组收缩膨胀方法,有效解决了CPU进行勒让德函数递推计算过程中内存读写冲突问题;然后,引入Hilbert空间填充曲线,将二维格网重力异常数据降维成为具有高度聚簇性的一维数组,通过其在GPU显存中的快速索引,实现了Stokes积分的高效计算。在CPU型号为Intel Xeon(R) Gold 6130、GPU型号为Tesla V100的计算机上进行实验,利用该混合并行方案计算6°×6°范围、1′分辨率的全张量海洋扰动重力梯度基准图仅需15.84 s,而传统串行方法需要35 min以上;计算2°×2°范围、30″分辨率基准图仅需22.7 s,串行方法则需要52 min;计算2°×2°范围、10″分辨率基准图需要26.7 min,串行方法则需要两天以上。在保证并行计算绝对误差小于等于1×10^{-6} E、$T_{xx}+T_{yy}+T_{zz}$绝对值的最大值小于0.1 E、均方根误差小于0.01 E的前提下,实现了全球全张量海洋扰动重力梯度基准图模型的快速构建。

关键词:扰动重力梯度;边值问题;CPU;GPU;混合并行

中图分类号:P223 **文献标识码**:A **收稿日期**:2023-09-06

DOI:10.13203/j.whugis20220142 **文章编号**:1671-8860(2025)03-0515-13

Rapid Construction of Marine Disturbing Gravity Gradient Reference Map by CPU and GPU Hybrid Parallel Method

HUANG Yan[1,2] *LI Shanshan*[2] *LÜ Minghao*[3] *FAN Diao*[2] *TAN Xuli*[2] *FENG Jinkai*[2]

1 National Innovation Institute of Defense Technology, Academy of Military Science, Beijing 100071, China

2 Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001, China

3 96941 Troops, Beijing 100080, China

Abstract: **Objectives**: High precision and high resolution marine disturbing gravity gradient reference map is one of the key technologies to put underwater disturbing gravity gradient assisted inertial navigation into practice. At present, gravity anomaly is mainly measured by satellite and calculated based on the boundary value problem theory and the removal recovery technique. **Methods**: The traditional serial algorithm has some problems such as low efficiency and long time consuming. In order to meet the needs of rapid construction of marine disturbing gravity gradient reference maps on a large scale and even around the world, this paper designs an efficient hybrid parallel computing scheme using central processing unit (CPU)+graph processing unit (GPU) hybrid parallel programming platform. First, the shrink and expansion method of ar-

基金项目:国家自然科学基金(42174007,42174013)。

第一作者:黄炎,助理研究员,主要从事水下重力导航和计算机技术研究。781367531@qq.com

通信作者:李姗姗,博士,教授。zzy_lily@sina.com

ray is proposed to solve the problem of memory conflict between read and write in Legendre recursive computation. Then Hilbert space filling curve is used to reduce the dimension of two-dimensional grid gravity anomaly data into a one-dimensional array with high clustering. The efficient calculation of Stokes integral is realized by the fast index in GPU video memory. **Results**: The experiments are carried out on computers with Intel Xeon (R) Gold 6130 CPU and Tesla V100 GPU. The hybrid parallel scheme is used to calculate $6^\circ\times6^\circ$ range and $1'$ resolution full tensor marine disturbing gravity gradient reference map and only takes 15.84 seconds, while the traditional serial method takes more than 35 minutes. To calculate $2^\circ\times2^\circ$ range and $30''$ resolution map, the proposed method only takes 22.7 seconds, while the serial method takes 52 minutes. To calculate $2^\circ\times2^\circ$ range and $10''$ resolution map, the proposed method only takes 26.7 minutes, while the serial method takes more than two days. **Conclusions**: On the premise of ensuring that the absolute error of parallel computing is less than or equal to 1×10^{-6} E, the maximum absolute value of $T_{xx}+T_{yy}+T_{zz}$ is less than 0.1 E and the root mean square error is less than 0.01 E, the rapid construction of the global total tensor marine disturbing gravity gradient reference map model is realized.

Key words: disturbing gravity gradient; boundary value problem; CPU; GPU; hybrid parallel

随着海洋强国战略的不断深入,水下潜航器在军事与民事的广泛应用已成为国家海洋发展的战略重点,如何实现水下潜航器长期、自主、隐蔽、高精度航行是满足各项任务要求的先决条件。水下潜航器常使用惯性导航技术进行无源定位导航,但惯性导航系统在长时间导航过程中由于误差累积,定位导航精度大打折扣,重力辅助惯性导航技术具有弥补惯导缺陷、有效修正惯导累积误差的优势,将成为未来水下潜航器定位导航的重点研究方向之一[1-3]。重力辅助惯性导航的基本原理是利用水下潜航器搭载的相应重力测量仪器实测数据和基准图上惯导指示位置图上数据,通过匹配算法对惯性导航位置和元器件误差进行修正。常用的重力量有重力异常与扰动重力梯度,重力异常匹配算法由于只有一个匹配量,易受到各类误差和载体运动异常的影响。而重力梯度作为重力位的二阶导数,具备6个分量(其中5个独立分量)可以组合搭配进行匹配导航,且重力梯度不易受到水下潜器载体自身加速度的影响,因此扰动重力梯度匹配惯性导航理论上精度更高[4-5]。

海洋扰动重力梯度基准图是扰动重力梯度辅助惯性导航得以实现的先决工作,如何快速获取高精度扰动重力梯度基准图成为研究重点。本文依据边值问题理论,基于移去-恢复技术将实际扰动重力梯度视为由两部分组成,即地球重力场模型计算得到的模型扰动重力梯度 $T_{\alpha\beta}^{M}$ 和剩余扰动重力梯度 $\delta T_{\alpha\beta}(\alpha,\beta=x,y,z)$;然后根据 $T_{\alpha\beta}^{M}$ 和 $\delta T_{\alpha\beta}$ 的计算特性,使用基于中央处理器(central processing unit,CPU)的OpenMP(open multi-processing)和基于图形处理器(graph processing unit,GPU)的统一计算设备架构(compute unified device architecture,CUDA)并行算法,设计CPU+GPU混合编程并行计算方案,提出数组收缩膨胀方法解决了缔合勒让德函数递推过程内存读写冲突问题,并通过引入Hilbert空间填充曲线将二维格网的基准图降维至一维格网,减少GPU计算负荷和显存读取时间,最终实现全球海洋扰动重力梯度基准图的快速构建。

1 基于CPU的模型扰动重力梯度并行方案

随着计算机技术的快速发展,计算机硬件更新迭代频繁,CPU作为整个计算机系统的计算和控制核心发展极为迅速,已由最初的单核发展到了16核、32核乃至更多核心的众核CPU[6]。其中OpenMP就是为多核/共享内存处理器设计开发的,是一个基于共享存储器的并行环境[7]。通常运算程序仅使用CPU的一个线程(即串行程序),对于硬件快速发展的当下,造成了计算资源的极大浪费。OpenMP则可以通过对串行程序进行简单的改进,实现多线程并行计算任务。同时共享存储系统也有着其天然的缺点,在进行并行计算时,由于存储系统由各个线程共享,因此在计算过程中易导致内存的读写冲突,因此如何对共享内存进行分配和隔离成为需要研究的重点。

1.1 模型扰动重力梯度计算

考虑到梯度的对称性以及主对角线元素之和为0,构建海洋扰动重力梯度基准图通常选取计算6个分量(5个独立分量,1个计算检核条件),各个分量计算过程与方法相似,本文仅以其

中1个分量 T_{yy} 为例介绍扰动重力梯度计算过程。模型扰动重力梯度计算过程可以分为两步:缔合勒让德函数的计算和球谐级数的计算。

模型扰动重力梯度的球谐级数计算形式是较为简单的级数求和形式,可以直接进行并行化处理,其表达式为:

$$\begin{cases} T_{yy}^{M}=\sum\limits_{m=0}^{N}\left(E_{m}^{yy}\cos(m\lambda)+F_{m}^{yy}\sin(m\lambda)\right) \\ E_{m}^{yy}=\dfrac{GM}{r^{3}}\sum\limits_{n=2}^{N}\left(\dfrac{R}{r}\right)^{n}\bar{C}_{nm}^{*}P_{nm}^{yy}(\sin\varphi) \\ F_{m}^{yy}=\dfrac{GM}{r^{3}}\sum\limits_{n=2}^{N}\left(\dfrac{R}{r}\right)^{n}\bar{S}_{nm}P_{nm}^{yy}(\sin\varphi) \end{cases} \tag{1}$$

式中,n、m 分别表示截断阶数;R 表示平均地球半径;r 表示地心向径,海洋处近似认为 $R/r\approx1$;φ 表示地心纬度;λ 表示地心经度;$\bar{C}_{nm}^{*}$ 表示剔除正常重力场偶阶带谐项的模型系数;$\bar{S}_{nm}$ 为地球重力场模型系数;$P_{nm}^{yy}(\sin\varphi)$ 表示计算 T_{yy}^{M} 所使用的勒让德函数。

$P_{nm}^{yy}(\sin\varphi)$ 的计算需要进行递推运算,即后一步的计算需要前一步的结果作为起算值,因此若直接对其计算过程进行并行化处理,会导致内存读写冲突造成计算崩溃,且其在极点附近(即 φ 接近90°)会产生奇异,因此需要首先对其去奇异,然后解决递推计算并行化问题。$P_{nm}^{yy}(\sin\varphi)$ 去奇异后的递推计算式为:

$$P_{nm}^{yy}(\sin\varphi)=\begin{cases} -\dfrac{1}{2}\sqrt{\dfrac{n(n-1)(n+1)(n+2)}{2}}\sin^{2}\varphi\bar{P}_{n,2}(\sin\varphi)-\sqrt{\dfrac{n(n+1)}{2}}\cos\varphi\sin\varphi\bar{P}_{n,1}(\sin\varphi)- \\ \qquad\dfrac{1}{2}(n+1)(n\sin^{2}\varphi+2)\bar{P}_{n,0}(\sin\varphi),m=0 \\ -\dfrac{1}{4}\sqrt{(n-2)(n-1)(n+2)(n+3)}\sin^{2}\varphi\bar{P}_{n,3}(\sin\varphi)-\sqrt{(n-1)(n+2)}\cos\varphi\sin\varphi\cdot \\ \qquad\bar{P}_{n,2}(\sin\varphi)-\dfrac{1}{4}\left[(n-1)\sin^{2}\varphi+4\right](n+2)\bar{P}_{n,1}(\sin\varphi),m=1 \\ -\dfrac{1}{4}\sqrt{(n-3)(n-2)(n+3)(n+4)}\,\bar{P}_{n,4}(\sin\varphi)-\dfrac{1}{2}(n^{2}+3n+6)\bar{P}_{n,2}(\sin\varphi)- \\ \qquad\dfrac{1}{4}\sqrt{2n(n-1)(n+1)(n+2)}\,\bar{P}_{n,0}(\sin\varphi),m=2 \\ -\dfrac{1}{4}\sqrt{(n-m-1)(n-m)(n+m+1)(n+m+2)}\,\bar{P}_{n,m+2}(\sin\varphi)- \\ \qquad\dfrac{1}{4}\sqrt{(n+m-1)(n+m)(n-m+1)(n-m+2)}\,\bar{P}_{n,m-2}(\sin\varphi)- \\ \qquad\dfrac{1}{2}(n^{2}+3n+m^{2}+2)\bar{P}_{n,m}(\sin\varphi),m>2 \end{cases} \tag{2}$$

式中,$\bar{P}_{n,m}(\sin\varphi)$ 表示纬度 φ 处的缔合勒让德函数。

由式(2)即可计算全球任意点位的缔合勒让德函数,随后结合CPU单线程计算能力和整体调控能力强的特点,设计利用OpenMP算法对区域模型扰动重力梯度进行并行计算,对缔合勒让德函数递推和球谐级数计算进行整体并行,充分利用CPU计算性能和任务调控能力。

1.2　基于OpenMP的CPU并行方案

由于缔合勒让德函数的计算是一个递推的过程,其计算过程难以进行拆分,需要将整个勒让德函数递推过程作为一个整体,将不同位置点缔合勒让德函数计算作为不同的并行单元(即每一个线程执行的最小单元),因此每一个并行单元中至少需要划分一部分独立存储空间用来存储缔合勒让德函数,并且并行单元的个数也会直接影响所需存储空间的大小,在计算过程中为保证一定精度,如果缔合勒让德函数计算至2 160阶,共2 160×2 160/2个双精度型的数据需要存储,一个双精度型数据需要占用8 byte的存储空间,此时需要开辟一个17.8 Mbyte的存储空间,这仅仅是一个并行单元只存储一个 $\bar{P}_{nm}(\sin\varphi)$ 所需要的空间[6]。显然,计算过程不仅仅只需要一个这样的数组,而且并行单元的个数也会因并行方案和计算量的变化而变化,GPU的显存往往没有计算机内存大,且GPU显存通常是不可拓展的[7],因此勒让德函数计算的并行方案更适合使用CPU并行算法,而后续球谐级数计算部分,为保证计算整体性,减少数据在CPU和GPU之间的传输所消耗的时间,均采用基于OpenMP的CPU并行算法。

确定并行算法后,需要针对计算过程和使用

的并行算法进行并行方案设计,方案设计过程中应着重解决内存占用过高和读写冲突问题。为有效解决上述两个问题,本文提出了计算数组收缩膨胀方法如图1所示。

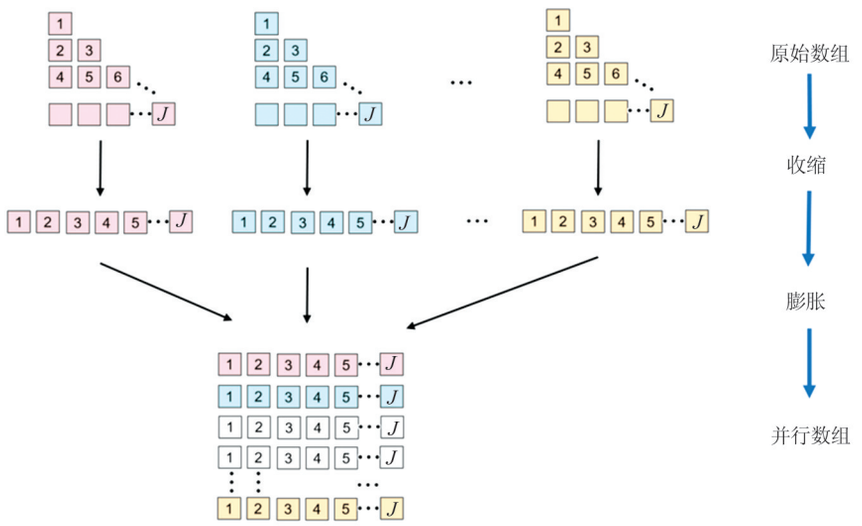


图1 数组收缩膨胀示意图

Fig. 1 Schematic Diagram of the Shrink and Expansion Method of Array

以存储缔合勒让德函数$\bar{P}_{nm}(\sin\varphi)$为例,由于缔合勒让德函数所占内存随着截断阶数和级次的增加呈二次方增长,通常计算缔合勒让德函数时,需要开辟一个二维数组$P[n][m]$对其进行存储,但由缔合勒让德函数特性$m\leqslant n$可知,$P[n][m]$仅需要存储下三角部分,其余部分需补0,如果直接使用二维数组进行存储,在计算时有一半的内存被浪费。且在进行并行计算时,为了保证每一个线程所计算的数组独立,需要根据并行单元个数开辟相应个数的数组进行计算。计算过程中还需要对每个线程所用数组进行准确索引,避免因多个线程同时调用一个内存位置而产生内存读写冲突的问题。对此,本文提出的数组收缩膨胀方法利用CPU完成模型扰动重力梯度的计算,既可有效减少内存空间的浪费,又可解决不同线程所用数组的索引问题。具体步骤如下:

1)数组收缩:利用二维矩阵存储$\bar{P}_{nm}(\sin\varphi)$,当阶数为$N$时,用以存储$P[n][m]$是一个$N\times N$的二维数组,由于其仅存储下三角部分,一半的内存空间被占用但并未使用,而一维数组则不存在这种问题,因此本文考虑将二维数组的下三角部分以一维数组的形式进行存储,即$P[nm]$,此时一维数组$P[nm]$仅需占用$(N+1)\times N/2$的内存空间。如图1所示,将二维数组的下三角部分以行为单位,每行内从左至右依次排列到收缩维度后的一维数组中,易得出二维数组$P[n][m]$中对应位置的勒让德函数在一维数组$P[nm]$中的对应元素为$P[n\times(n+1)/2+m]$。

2)数组膨胀:在进行收缩后,每一个并行单元都需要独立的$P[nm]$进行计算,即有t个并行单元就需要t个$(N+1)\times N/2$大小的一维数组,如果简单地开辟t个一维数组$P_1[nm]$,$P_2[nm]$,…,$P_t[nm]$用来存储,则在进行并行计算过程中不易对其进行索引,易导致因多线程重复使用相同内存位置而产生内存读写冲突。因此本文提出数组膨胀方法,将收缩过后的一维数组膨胀至二维数组$P[t][nm]$,在二维数组中通过索引行数t将不同内存位置分配给不同的线程,既可以解决索引问题,也能避免内存读写冲突问题。

2 基于GPU的剩余扰动重力梯度并行方案

尽管CPU有着强大的单线程计算能力和较大的计算机内存,对于整体并行有着先天优势[8]。但是受制于材料与工艺,其计算主频和总核心数难以呈量级式增长[9]。对于单个并行单元计算并不复杂但计算单元众多的任务,CPU的并行加速效果有限,但却十分适合GPU[10]。其中具有代表性的就是英伟达公司为其旗下GPU产品开发的异构并行平台CUDA[11]。目前GPU的硬件发展十分迅速,GPU内线程块和线程(CUDA核心)数已由几百提升至上千个,以Tesla V100显卡为例,其CUDA核心数已达到5 120个[12-14]。并且GPU相较于并行计算机成本较低,可以普遍被大

众接受。但是,GPU也存在缺点,尽管GPU有大量的CUDA核心,但是其每一个核心的计算能力无法与CPU相提并论,因此在利用GPU并行计算时,需要对并行单元进行细化,尽可能在充分利用GPU计算能力的条件下,使得CUDA核心运算量不超负荷。在剩余扰动重力梯度的计算过程中,每一个格网点的计算互不相关,可以独立进行计算且单个格网点剩余扰动重力梯度的计算量不大,但格网点数众多,因此不同于CPU的整体并行,GPU则选择计算单一格网的剩余扰动重力梯度作为一个并行单元,增加并行单元个数,减少单个并行单元计算量。

2.1　剩余扰动重力梯度计算

依据移去-恢复技术思路,其目的就是移去扰动重力梯度中的长波分量,而剩余部分则利用计算点周围积分半径内流动点的重力异常剩余量$\delta\Delta g$($\delta\Delta g=\Delta g-\Delta g_M$,$\Delta g$表示实际重力异常,$\Delta g_M$表示模型重力异常)进行计算。由于中央区计算存在奇异情况,因此单个并行单元计算过程需要分为内区和中央区。

1)内区剩余扰动重力梯度计算

利用剩余重力异常计算剩余扰动重力梯度的公式为:

$$\delta T_{yy}=\frac{R}{4\pi}\iint_{\sigma}\left[\frac{1}{r^2}\left(\frac{\partial^2 S(r,\psi)}{\partial\psi^2}\left(\frac{\partial\psi}{\partial\varphi}\right)^2+\frac{\partial S(r,\psi)}{\partial\psi}\frac{\partial^2\psi}{\partial\varphi^2}\right)+\frac{1}{r}\frac{\partial S(r,\psi)}{\partial r}\right]\delta\Delta g\mathrm{d}\sigma \tag{3}$$

$$\begin{cases}\dfrac{\partial S(r,\psi)}{\partial r}=-\dfrac{t^2}{R}\left[\dfrac{1-t^2}{D^3}+\dfrac{4}{D}+1-6D-t\cos\psi\left(13+6\ln\dfrac{1-t\cos\psi+D}{2}\right)\right]\\ \dfrac{\partial S(r,\psi)}{\partial\psi}=-t^2\sin\psi\left[\dfrac{2}{D^3}+\dfrac{6}{D}-8-3\dfrac{1-t\cos\psi-D}{D\sin\psi}-3\ln\dfrac{1-t\cos\psi+D}{2}\right]\\ \dfrac{\partial^2 S(r,\psi)}{\partial\psi^2}=-t^2\cos\psi\left[\dfrac{2}{D^3}+\dfrac{6}{D}-8-3\dfrac{1-t\cos\psi-D}{D\sin^2\psi}-3\ln\dfrac{1-t\cos\psi+D}{2}\right]+\\ \quad t^3\sin^2\psi\left[\dfrac{6}{D^5}+\dfrac{6}{D^3}+3\dfrac{D-1}{D^2\sin^2\psi}+3\dfrac{D+1}{D(1-t\cos\psi+D)}-3\dfrac{1-t\cos\psi-D}{D\sin^2\psi}\left(\dfrac{2\cos\psi}{t\sin^2\psi}+\dfrac{1}{D^2}\right)\right]\end{cases} \tag{4}$$

式中,σ表示整个积分区域;$\mathrm{d}\sigma$为球面上的流动面元;ψ是球外计算点与流动面元的球心角距;$S(r,\psi)$表示Stokes核函数,其解析式可表示为:

$$S(r,\psi)=t\left[\frac{2}{D}+1-3D-t\cos\psi\left(5+3\ln\frac{1-t\cos\psi+D}{2}\right)\right] \tag{5}$$

式中,$\psi=\cos^{-1}(\sin\varphi\sin\varphi'+\cos\varphi\cos\varphi'\cos(\lambda'-\lambda))$,$\varphi'$、$\lambda'$分别为球面流动面元的纬度与经度;$t=R/r$;$D=\sqrt{1-2t\cos\psi+t^2}$。

利用式(3)与式(4)即可计算全球任意点内区剩余扰动重力梯度δT_{yy}分量。

2)中央区剩余扰动重力梯度计算

由式(4)可知,Stokes核函数及其导数在计算过程中均用到了$1/D$,因此在流动点接近计算点时,$S(r,\psi)$及其导数会异常增大,导致奇异性,这点在计算径向扰动重力梯度时尤为明显[15-16]。因此,在计算中央区扰动重力梯度时,不能使用原公式计算。本文选用Taylor级数展开法对中央区奇异问题进行求解。

首先,区域大地水准面可以近似看作平面,则大地水准面上的计算点坐标为$(x,y,0)$,并将中央区视为以计算点为圆心、邻近近1′~2′距离为半径的圆形区域,则半径$s_0=R\psi_0$,ψ_0为中央区圆心与边缘位置的球心角距。在对中央区进行格网化后,半径s_0可近似等于$\sqrt{\Delta x\times\Delta y}$[16]。

将直角坐标转化到以计算点为原点的极坐标系下,则流动点在该极坐标系下的坐标为(s,α),其中$s=\sqrt{(x'-x)^2+(y'-y)^2}$,$(x,y)$为计算点坐标,$(x',y')$为流动点坐标,故易得$x=s\cos\alpha$,$y=s\sin\alpha$,$\mathrm{d}x'\mathrm{d}y'=s\mathrm{d}s\mathrm{d}\alpha$。因此,可得中央区剩余扰动重力梯度计算式为:

$$\delta T_{yy}=\frac{1}{2\pi}\int_0^{2\pi}\int_0^{s_0}\frac{\delta\Delta g}{s^2}(3\sin^2\alpha-1)\mathrm{d}s\mathrm{d}\alpha \tag{6}$$

对式(6)中$\delta\Delta g$在计算点处进行Taylor级数展开可得:

$$\delta\Delta g=\delta\Delta g_P+xg_1+yg_2+\frac{1}{2}(x^2g_{11}+2xyg_{12}+y^2g_{22})+R_{\delta\Delta g} \tag{7}$$

式中,$\delta\Delta g_P$表示泰勒级数展开点P的重力异常剩余量;$R_{\delta\Delta g}$表示阶段误差;系数参数$g_1=\dfrac{\partial\Delta g}{\partial x}$,

$$g_2=\frac{\partial \Delta g}{\partial y},g_{11}=\frac{\partial^2 \Delta g}{\partial x^2},g_{22}=\frac{\partial^2 \Delta g}{\partial y^2},g_{12}=\frac{\partial^2 \Delta g}{\partial x \partial y}。$$

利用上述直角坐标与极坐标转换关系,可将式(7)转换为极坐标形式:

$$\delta \Delta g=\delta \Delta g_P+s(g_1\cos\alpha+g_2\sin\alpha)+\frac{s^2}{2}\cdot(g_{11}\cos^2\alpha+2g_{12}\sin\alpha\cos\alpha+g_{22}\sin^2\alpha)+R_{\delta\Delta g} \tag{8}$$

为求得式(8)中系数参数,仅考虑线性项(假设高阶项所占比重较低),利用剩余重力异常格网数据可得:

$$\begin{cases} g_{1_{p,q}}=\dfrac{\delta \Delta g_{p+1,q}-\delta \Delta g_{p-1,q}}{2\Delta x} \\ g_{2_{p,q}}=\dfrac{\delta \Delta g_{p,q+1}-\delta \Delta g_{p,q}}{2\Delta y} \\ g_{11_{p,q}}=\dfrac{g_{1_{p+1,q}}-g_{1_{p-1,q}}}{2\Delta x} \\ g_{12_{p,q}}=\dfrac{g_{1_{p,q+1}}-g_{1_{p,q-1}}}{2\Delta y} \\ g_{22_{p,q}}=\dfrac{g_{2_{p,q+1}}-g_{2_{p,q-1}}}{2\Delta y} \end{cases} \tag{9}$$

式中,p表示格网行数,$p=1,2,\cdots,P$;q表示格网列数,$q=1,2,\cdots,Q$。

将式(8)、式(9)代入式(6)可得:

$$\delta T_{yy}=\frac{5g_{22}-g_{11}}{16}s_0 \tag{10}$$

利用式(10)即可计算全球任意点中央区剩余扰动重力梯度δT_{yy}分量。

2.2 基于CUDA的GPU并行算法

在确定了并行计算单元后,设计利用CUDA算法对剩余扰动重力梯度进行并行计算,在保证GPU单线程计算不超负荷的情况下,充分发挥GPU每一个核心的计算能力。为了区分于CPU程序任务,CUDA有一套独立的核函数形式:kernel_function<<<num_block,num_thread>>>(param1,param2),其中num_blocks表示并行计算时分配的线程块个数;num_threads表示每个线程块中所包含的CUDA核心个数。为了更好地控制和调配这些线程块和CUDA核心,CUDA可以对使用的线程块和线程进行编码从而精确划分计算任务。线程块与线程编码如图2所示。

CUDA为了在线程格网中索引特定的线程,根据线程层级划分4个索引变量,其中($\mathrm{Thread}_{\mathrm{Id},x}$,$\mathrm{Thread}_{\mathrm{Id},y}$)分别为线程在线程块中的索引,($\mathrm{Block}_{\mathrm{Id},x}$,$\mathrm{Block}_{\mathrm{Id},y}$)分别为线程所在线程块在线程格网中的索引,$D_{\mathrm{block},x}$和$D_{\mathrm{block},y}$分别表示线程块在$x$、$y$方向的维度[6]。根据上述索引变量,

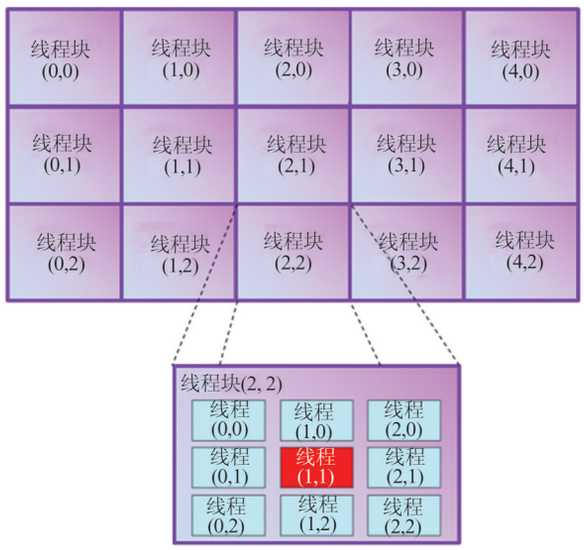


图2 CUDA线程索引示意图

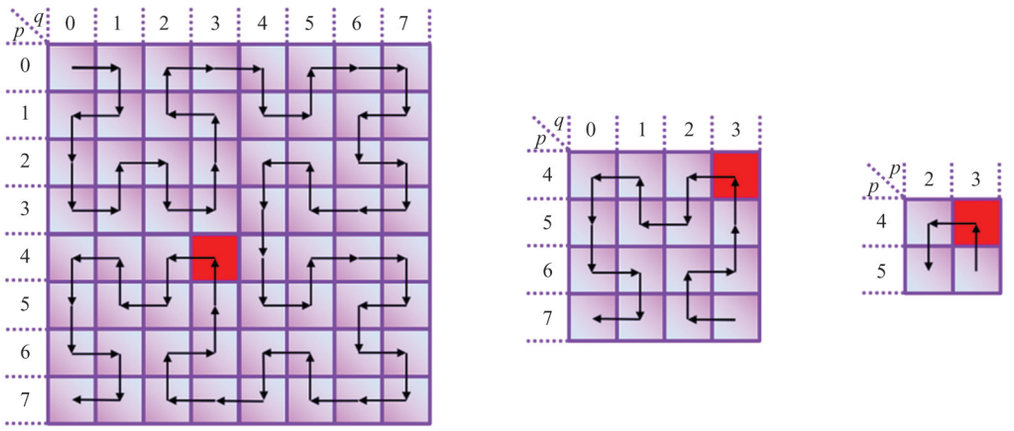
Fig. 2 Diagram of CUDA Thread Index

可以得到任意线程在线程格网中的x、y方向索引计算式为:

$$\begin{cases} \mathrm{Id}x=\mathrm{Block}_{\mathrm{Id},x}\cdot D_{\mathrm{block},x}+\mathrm{Thread}_{\mathrm{Id},x} \\ \mathrm{Id}y=\mathrm{Block}_{\mathrm{Id},y}\cdot D_{\mathrm{block},y}+\mathrm{Thread}_{\mathrm{Id},y} \end{cases} \tag{11}$$

图2以一个包含3×5个线程块的线程格网为例,每个线程块中包含3×3个线程。以图2中红色线程为例,其所在的线程块x、y方向维度均为3,其所在线程块在线程格网中x、y方向的索引均为2,该线程在其所在线程块中x、y方向的索引均为1,因此该线程的Idx=7,Idy=7。

在计算过程中由于格网点数众多,利用二维线程格网进行索引增加了线程调度过程的耗时[6],因此考虑通过一定方法将二维格网进行降维处理,利用一维线程格网进行计算以最大限度地提高计算效率。又由于在进行降维过程中会使原来相邻的格网点分离,使得整体空间聚簇性变差[17-23],即原相邻格网在一维格网中编码序号相隔较远,因此在存储器中索引时需要读取更多的数据。尤其是在使用GPU时,由于显存读写速度不及主板内存[6],因此尽量减小二维格网中临近格网在一维格网中的编码差值,提高空间聚簇型,可以有效提高GPU的计算效率。经国内外学者对比研究表明,在众多空间填充曲线中,Hilbert空间填充曲线的空间聚簇性最好[24-26],二维格网临近点通过Hilbert曲线进行降维后在一维格网中的编码能够尽可能保持接近,有效提高二维格网在一维存储器(显存)上的索引效率[27]。因此,本文考虑使用Hilbert空间填充曲线对二维格网进行降维处理,其空间填充方式如图3所示。



(a) 3阶曲线　　(b) 2阶曲线　　(c) 1阶曲线

图 3　Hilbert空间填充曲线示意图

Fig. 3　Schematic Diagram of Hilbert Space Filling Curve

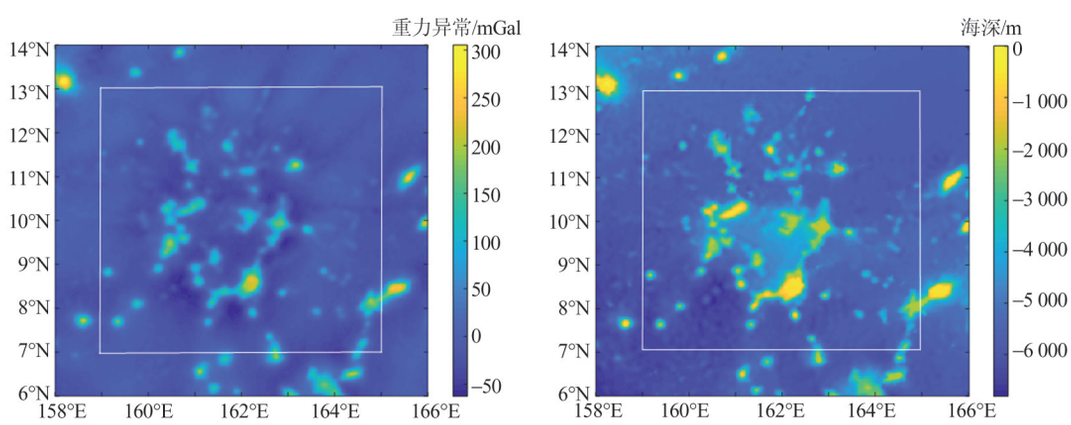
Hilbert空间填充曲线建立二维以及高维空间格网和一维格网间的映射关系,这样的映射关系序列码被称为Hilbert空间排列码[23-25]。常用的二维Hilbert空间填充曲线有三阶,如图3所示。1阶Hilbert曲线由4个相邻格网构成,空间曲线不重复地贯穿整个格网;2阶Hilbert曲线由4个1阶曲线通过旋转构成,曲线贯穿区域;同样,3阶Hilbert曲线是由1个2阶曲线通过旋转构成,曲线贯穿整个区域。为了对计算区域进行有效降维编码,避免多余的计算量,本文仅使用最高为3阶的Hilbert曲线,通过将计算区域划分为K个3阶Hilbert曲线首尾相连的方式对计算区域进行降维。Hilbert空间填充曲线的格网编码与解码可采用递归算法和迭代算法,因此,只需要对3阶Hilbert曲线格网进行顺序编码,得到待求格网点所在的3阶Hilbert格网编码以及其在该格网中的编码,即可在一维数组中对其进行索引。以图3中红色格网点为例,其在3阶Hilbert曲线上的值为53。经过上述并行方案,即可利用GPU完成剩余扰动重力梯度的计算。

3　实验验证与分析

为验证本文所提方法的有效性,在下述硬件、软件实验平台对该算法进行实验验证:实验环境为高性能台式机,处理器为Intel Xeon(R) Gold 6130 @ 2.10 GHz;GPU为Tesla V100,内存64 GB;软件环境为64位Windows 10操作系统,VS 2013+CUDA 9.2平台。

3.1　精度分析

为验证本文所提并行算法精度,选取6°×6°范围研究区(海洋格网范围为8°×8°,位于6°N~14°N,158°E~166°E)。该区域海深最深6 872 m,最浅0 m,区域平均高程−3 657.22 m;该区域重力异常最大值313.8 mGal,最小值−59.3 mGal,平均重力异常12.4 mGal。区域重力异常与海深如图4所示。



(a) DTU18重力异常　　(b) 海底地形

图 4　实验区域物理特性图

Fig. 4　Physical Characteristic Map of the Study Area

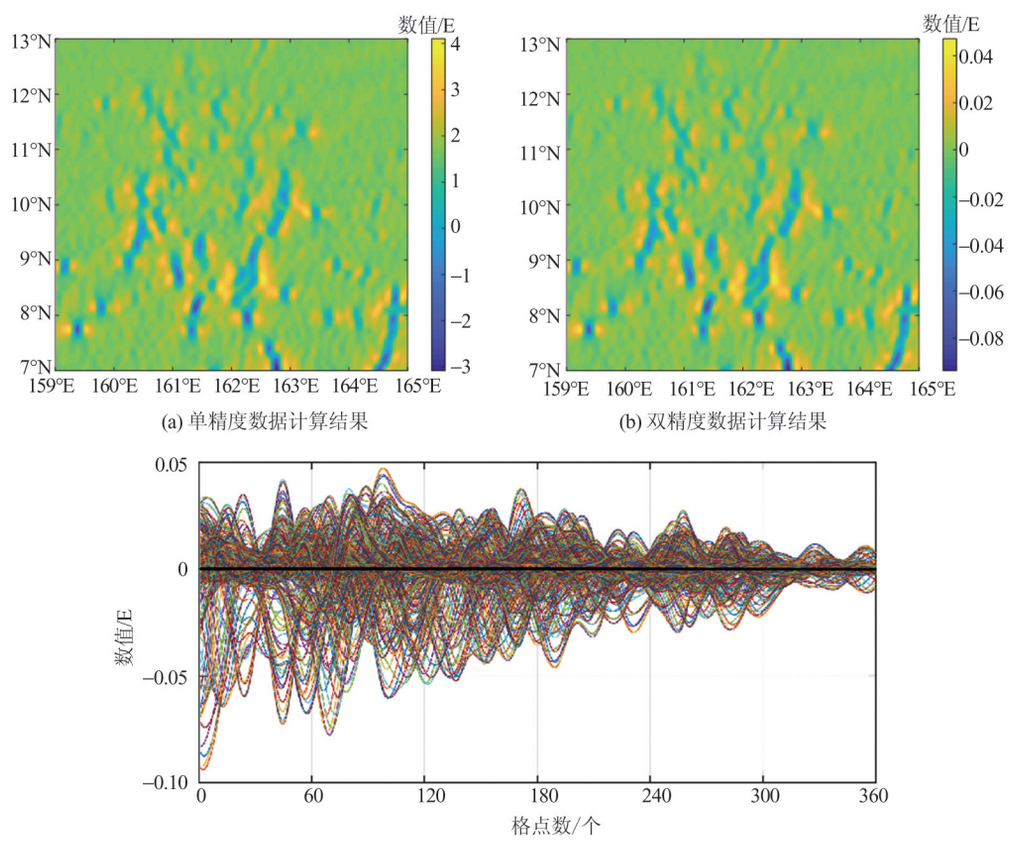
目前,GPU在进行单精度浮点运算的效率要远远高于双精度浮点运算[6],对于同一区域扰动重力梯度基准图的计算,使用单精度数据计算剩余扰动梯度的耗时仅为利用双精度数据计算耗时的1/4,但由于在计算剩余扰动重力梯度过程中存在大量正、余弦、正切、开方等函数,仅使用单精度浮点计算结果不能满足计算精度的要求,有限的有效位数带来的舍入误差会变得更加明显,甚至影响计算精度。为此,本文利用GPU分别使用单精度浮点数据和双精度数据,计算实验区域剩余扰动重力梯度6个分量,并利用CPU使用双精度数据串行计算实验区域剩余扰动重力梯度,以此结果作为真值,与GPU计算结果作差并进行对比分析,精度分析结果见表1(1 E=10^{-9} s^{-2})。

为进一步验证并行方案数学模型的可靠性和正确性,根据上述计算结果,基于$T_{xx}+T_{yy}+T_{zz}=0$的理论约束条件,分别将单精度浮点数据与双精度浮点数据运算结果的内符合情况进行统计,结果如图5所示,数值特征统计见表2。

表1 不同数据类型并行计算精度分析/E

Table 1 Accuracy Analysis of Parallel Computing for Different Data Types/E

数据类型	最大差值	最小差值	平均差值	RMS
双精度数据	1×10^{-6}	-1×10^{-6}	$<1\times10^{-12}$	$<1\times10^{-12}$
单精度数据	5.32	−1.36	0.18	0.42



(a) 单精度数据计算结果

(b) 双精度数据计算结果

(c) 双精度数据结果曲线图

图5 不同数据类型计算$T_{xx}+T_{yy}+T_{zz}$结果

Fig. 5 Result of $T_{xx}+T_{yy}+T_{zz}$ for Different Data Types

表2 $T_{xx}+T_{yy}+T_{zz}$数值特征分析/E

Table 2 Analysis of $T_{xx}+T_{yy}+T_{zz}$ Numerical Characteristics/E

数据类型	最大值	最小值	平均值	标准差	RMS
单精度数据	4.43	−3.21	0.20	0.023	0.047
双精度数据	0.047	−0.094	$<1\times10^{-4}$	0.002 6	0.003 9

由图5(b)、图5(c)和表2可知,利用双精度数据计算所得$T_{xx}+T_{yy}+T_{zz}$绝对值最大值小于0.1 E,平均值小于1×10^{-4} E,标准差为0.002 6 E,均方根值为0.003 9 E,$T_{xx}+T_{yy}+T_{zz}$围绕着0随机波动,符合扰动重力梯度的内符合条件,能够在一定程度上表明上述计算方法数学模型的可靠性和软件编程的正确性。由表1、表2和图5(a)可知,利用单精度数据进行计算,由于数据有效位数的限制,由舍入误差带来的剩余扰动重力梯度的误差绝对值最大可达5.32 E,平均误差也有

0.18 E,且在利用$T_{xx}+T_{yy}+T_{zz}$进行内符合精度分析时,最小达到−3.21 E,最大达到4.43 E,无法满足海洋扰动重力梯度基准图的精度需求[4]。而使用双精度数据进行计算,并行计算绝对误差绝对值最大仅为1×10^{-6} E,平均误差小于1×10^{-12} E,并行计算精度满足海洋扰动重力梯度基准图的计算精度需求。因此在利用GPU进行剩余扰动梯度计算时,应牺牲一定计算效率以满足精度需求,选取双精度数据进行计算。

3.2　效率分析

3.2.1　不同并行方案效率分析

在使用双精度数据保证计算精度的前提下,为验证本文所提并行算法的效率,分别使用串行计算方案、纯CPU并行方案、纯GPU并行方案及CPU+GPU混合并行方案,对实验区域扰动重力梯度进行计算,模型扰动重力梯度计算采用EIGEN6C4地球重力场模型,截断阶数计算至2 160阶[28];使用卫星测高数据反演所得DTU18模型数据(1′分辨率)作为实际重力异常值进行剩余扰动重力梯度计算。计算剩余扰动重力梯度时,内区选取积分半径1°范围进行计算,中央区选取中心格网点和围绕其周围的8个格网点共计9个格网区域。进行3次计算,计算耗时取3次结果的平均值。计算结果如图6所示,计算耗时统计见表3。

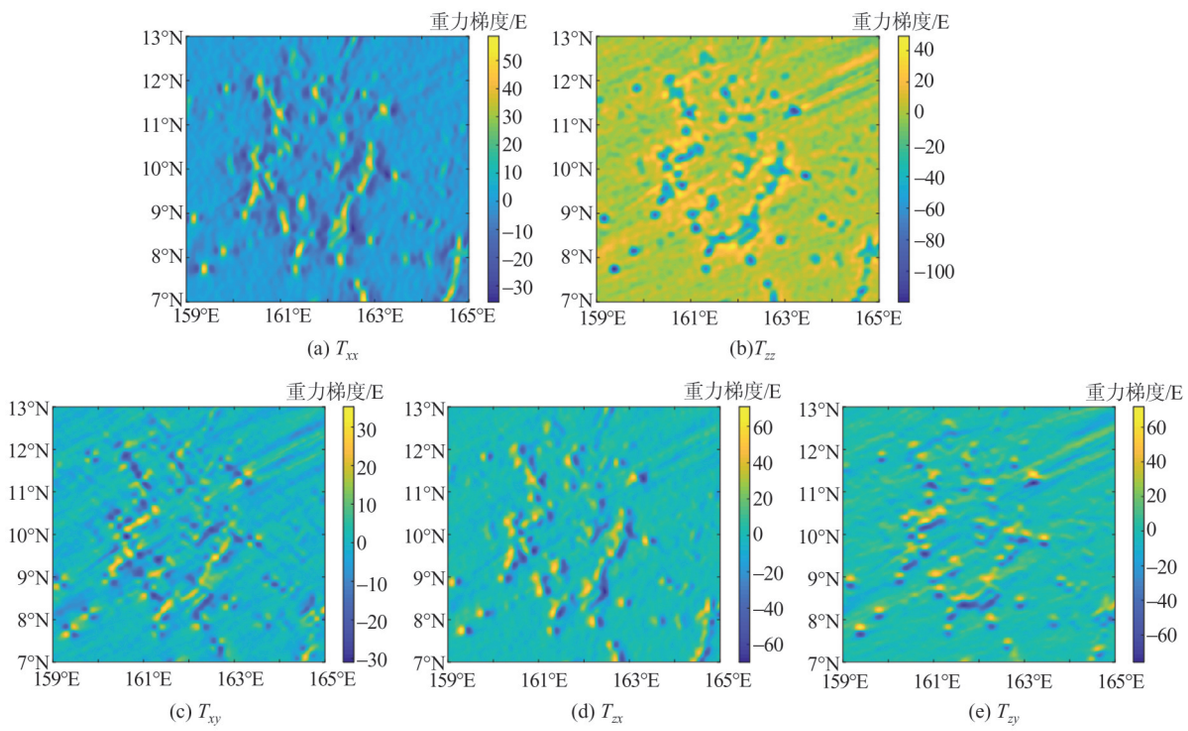


图6　实验区域重力梯度图

Fig. 6　Disturbing Gravity Gradient Map of the Study Area

表3　不同并行方案计算耗时

Table 3　Time Consuming of Different Parallel Schemes

方案	模型扰动重力梯度计算耗时/s	模型扰动重力梯度计算加速比	剩余扰动重力梯度计算耗时/s	剩余扰动重力梯度计算加速比	总耗时/s	总加速比
串行计算	544.99	1.00	1 582.560	1.00	2 127.55	1.00
纯CPU并行	14.36	37.95	41.270	38.34	55.63	38.24
纯GPU并行	31.63	17.23	1.484	1 066.21	33.11	64.26
CPU+GPU混合并行	14.36	37.95	1.484	1 066.21	15.84	134.26

图6显示,利用双精度数据进行计算,4种计算方案的计算结果与该区域重力异常图和海底地形图有相关性,计算精度一致,任意两种计算方案间的最大偏差仅为1×10^{-6} E,且均符合$T_{xx}+T_{yy}+T_{zz}=0$的内符合条件,因此4种方案计算精度均可靠。由表3可知,利用传统串行方法计算该区域扰动重力梯度全张量需要2 127.55 s(35.46 min),利用并行方案进行计算耗时均小于

1 min,其中,纯CPU并行方案计算耗时55.63 s,总加速比约38倍;纯GPU并行方案计算耗时33.11 s,总加速比约64倍;CPU+GPU混合并行方案计算耗时15.84 s,总加速比约134倍,加速效果优于单系统并行方案。在扰动重力梯度计算过程中,模型扰动重力梯度部分计算耗时占比25.6%,剩余扰动重力梯度部分计算耗时占比74.4%,利用CPU并行方案计算模型部分和剩余部分的加速比分别为37.95倍和38.34倍,而利用GPU计算两部分的加速比分别为17.23倍和1 066.21倍,因此,本文所提CPU+GPU混合并行方案可以充分利用计算设备的运算能力,取长补短,最大限度地提高扰动重力梯度基准图的构建效率。

3.2.2 不同分辨率计算效率分析

为进一步验证本文算法的有效性,使用不同分辨率数据进行数值实验,计算范围选取2°×2°海洋区域,模型扰动重力梯度计算至2 160阶。计算剩余扰动重力梯度时,内区积分半径为1°,中央区选取中心格网点和围绕其周围的8个格网点共计9个格网区域,分辨率分别为1′、30″、10″,计算耗时如表4所示。

表4 不同分辨率计算耗时

Table 4 Calculation Time of Different Resolutions

分辨率	串行计算耗时	并行计算耗时	加速比
1′	4.02 min	1.92 s	125.36
30″	52.16 min	22.65 s	138.16
10″	64.85 h	26.67 min	145.86

由表4可以看出,随着分辨率的提高,计算耗时呈指数上涨,大范围高分辨率海洋扰动重力梯度基准图难以在短时间内得到,以2°×2°范围海洋扰动重力梯度基准图的构建任务为例进行实验。利用本文所提CPU+GPU并行算法计算不同分辨率海洋扰动重力梯度基准图的加速比均可达到125倍以上,最高达到145倍。完成1′分辨率海洋扰动重力梯度基准图构建仅需1.92 s;完成30″分辨率基准图构建仅需22.65 s,而传统串行方法则需要约52 min;完成10″分辨率基准图构建需要26.67 min,而串行方法则需要2 d以上。证明了本文所提并行方案对不同分辨率扰动重力梯度基准图的构建有着普遍有效性,可以有效提高各个分辨率扰动重力梯度基准图的构建速度。

4 全球海洋扰动重力梯度基准图构建与结果分析

通过上述CPU+GPU混合并行方案,可以实现大范围海洋扰动重力梯度基准图快速构建任务,本文对全球范围、1′分辨率海洋区域的扰动重力梯度基准图进行构建,计算使用EIGEN6C4地球重力场模型,截断阶数计算至2 160阶;使用卫星测高数据反演所得DTU18模型数据(1′分辨率)作为实际重力异常值进行剩余扰动重力梯度计算。计算剩余扰动重力梯度时,内区选取积分半径1°范围进行计算,中央区选取中心格网点和围绕其周围的8个格网点共计9个格网区域。并行算法耗时4.52 h,串行计算耗时超过28 d。计算结果如表5、图7所示。

表5 全球海洋扰动重力梯度数值特征

Table 5 Numerical Characteristics of Global Ocean Gravity Gradient

重力梯度	最大值/E	最小值/E	平均值/E	中位数/E
T_{xx}	127.33	−116.93	$<1\times10^{-6}$	0.082 7
T_{yy}	100.73	−105.30	$<1\times10^{-6}$	0.060 7
T_{zz}	228.28	−114.65	$<1\times10^{-6}$	−0.020 2
T_{xy}	80.22	−71.67	$<1\times10^{-6}$	−0.002 0
T_{zx}	106.48	−112.24	$<1\times10^{-6}$	0.001 5
T_{zy}	106.13	−104.44	$<1\times10^{-6}$	−0.005 9

由表5、图7可知,全球海洋扰动重力梯度平均值均小于1×10^{-6} E,且中位数在0 E附近,说明海洋区域大部分点位扰动重力梯度绝对值较小;扰动重力梯度与海底地形呈相关性,地形起伏剧烈区域,扰动重力梯度值变化较为明显。

5 结 语

本文利用CPU+GPU混合并行方法,针对扰动重力梯度计算各过程特点,提出了一种高效的并行计算方案,可以快速高效地完成大范围乃至全球范围海洋扰动重力梯度基准图的构建任务。经实验计算分析表明:

1)本文所提方法通过数组收缩膨胀方法和引入Hilbert空间填充曲线,有效克服了缔合勒让德函数递推计算的并行化问题以及GPU显存快速索引问题,可以充分利用设备计算能力。

2)本文所提方法的并行计算误差小于1×10^{-6} E,均方根误差小于1×10^{-12} E,并行计算过程产生的误差较小,计算精度较高。所得T_{xx}+

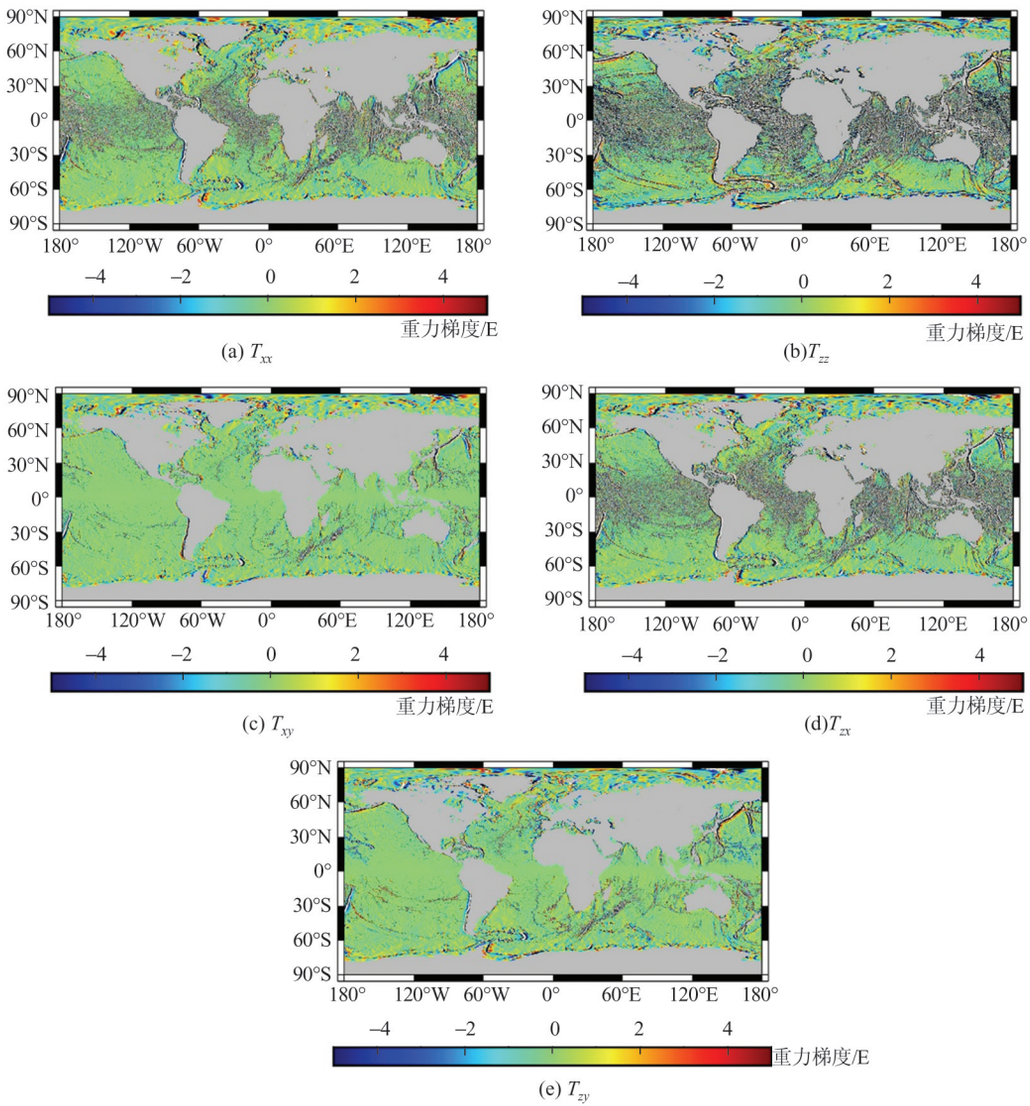


图 7　全球海洋扰动重力梯度图

Fig. 7　Global Ocean Disturbing Gravity Gradient Map

$T_{yy}+T_{zz}$绝对值最大值小于 0.1 E,平均值小于 1×10^{-6} E,标准差为 0.002 6 E,均方根误差为 0.003 9 E,$T_{xx}+T_{yy}+T_{zz}$围绕着 0 随机波动,符合扰动重力梯度的内符合条件,表明该并行方案具有一定的可靠性。

3)利用该并行方案计算 6°×6°范围、1′分辨率海洋扰动重力梯度基准图仅需 15.84 s,而传统串行方法需要 35 min 以上;计算 2°×2°范围、30″分辨率海洋扰动重力梯度基准图仅需 22.65 s,而串行方法则需要约 52 min;计算 2°×2°范围、10″分辨率基准图需要 26.67 min,而串行方法则需要 2 d 以上时间。利用该CPU+GPU混合并行方案进行海洋扰动重力梯度基准图构建,相较于传统串行计算,加速比最高可达 140 倍以上,有效减少了计算耗时,提高了计算效率。

4)全球全张量海洋扰动重力梯度基准图可在 4.6 h 内完成构建,相较于串行算法效率提高了 150 倍以上,由实验结果可知,全球海洋扰动重力梯度平均值均小于 1×10^{-6} E,中位数在 0 E 附近。

参　考　文　献

[1] 赵伟. 实时重力梯度补偿的惯性导航方法研究[D]. 武汉:武汉大学, 2014.
ZHAO Wei. Research on Inertial Navigation Method with Real-Time Gravity Gradient Compensation [D]. Wuhan: Wuhan University, 2014.

[2] LI Z W, ZHENG W, WU F. Improving the Reliability of Underwater Gravity Matching Navigation Based on a Priori Recursive Iterative Least Squares Mismatching Correction Method[J]. *IEEE Access*, 2020, 8: 8648-8657.

[3] YOO Y M, LEE W H, LEE S M, et al. Improvement of TERCOM Aided Inertial Navigation Sys-

tem by Velocity Correction[C]//IEEE/ION Position, Location and Navigation Symposium. Myrtle Beach, USA, 2012.

[4] 王博, 付梦印, 李晓平, 等. 水下重力匹配定位算法综述[J]. 导航与控制, 2020, 19(S1): 170-178.
WANG Bo, FU Mengyin, LI Xiaoping, et al. Overview of Underwater Gravity Matching Positioning Algorithms[J]. *Navigation and Control*, 2020, 19(S1): 170-178.

[5] 赵伟, 王正涛, 马强. 重力梯度数据模拟及其在惯性导航中的应用[J]. 大地测量与地球动力学, 2014, 34(5): 134-138.
ZHAO Wei, WANG Zhengtao, MA Qiang. Simulation of Gravity Gradient Data and Its Applications on Inertial Navigation[J]. *Journal of Geodesy and Geodynamics*, 2014, 34(5): 134-138.

[6] 黄炎. 基于并行计算的重力场若干问题方法研究[D]. 郑州: 信息工程大学, 2020.
HUANG Yan. Research on Several Problems of Gravity Field Based on Parallel Computing[D]. Zhengzhou: Information Engineering University, 2020.

[7] 黄佳喜. 扰动重力场快速(并行)计算方法研究[D]. 郑州: 信息工程大学, 2017.
HUANG Jiaxi. Study on Fast (Parallel) Calculation Method of Disturbed Gravity Field[D]. Zhengzhou: Information Engineering University, 2017.

[8] 都志辉. 高性能计算并行编程技术: MPI并行程序设计[M]. 北京: 清华大学出版社, 2001.
DU Zhihui. Parallel Programming Technology for High Performance Computing: MPI Parallel Programming[M]. Beijing: Tsinghua University Press, 2001.

[9] 周雪, 于平, 翁爱华, 等. 基于MPI和OpenMP的重力及重力梯度数据并行正演算法研究[J]. 世界地质, 2018, 37(3): 897-904.
ZHOU Xue, YU Ping, WENG Aihua, et al. Parallel Forward Modelling Algorithm with Gravity and Gravity Gradient Data Based on MPI and OpenMP[J]. *Global Geology*, 2018, 37(3): 897-904.

[10] 张晗, 钱育蓉, 侯海耀. CUDA下地质图像边缘检测算法并行优化[J]. 计算机工程与设计, 2019, 40(3): 691-698.
ZHANG Han, QIAN Yurong, HOU Haiyao. Parallel Optimization of Edge Detection Algorithm for Geological Images Under CUDA[J]. *Computer Engineering and Design*, 2019, 40(3): 691-698.

[11] NVIDIA Technical Report. Nvidia® Tesla® p100 - the Most Advanced Data Center Accelerator Ever Built[R]. Santa Clara, USA, 2017.

[12] 李午阳, 张健, 林巍. 基于CUDA的GPU并行优化重力三维反演[J]. 物探与化探, 2016, 40(1): 179-184.
LI Wuyang, ZHANG Jian, LIN Wei. Regularized Inversion of 3D Gravity Data: A New GPU Parallelized Method Based on CUDA[J]. *Geophysical and Geochemical Exploration*, 2016, 40(1): 179-184.

[13] SHARMA G, AGARWALA A, BHATTACHARYA B. A Fast Parallel Gauss Jordan Algorithm for Matrix Inversion Using CUDA[J]. *Computers & Structures*, 2013, 128: 31-37.

[14] 张舒, 褚艳利. GPU高性能运算之CUDA[M]. 北京: 中国水利水电出版社, 2009.
ZHANG Shu, CHU Yanli. CUDA for GPU High Performance Computing[M]. Beijing: China Water & Power Press, 2009.

[15] 赵炳杰, 杨光亮, 黎哲君. 移去-恢复法校准重力异常场及其应用[J]. 大地测量与地球动力学, 2019, 39(4): 361-365.
ZHAO Bingjie, YANG Guangliang, LI Zhejun. Study on the Remove-Restore Theory of Refining Gravity Anomaly and Its Application[J]. *Journal of Geodesy and Geodynamics*, 2019, 39(4): 361-365.

[16] 陈涛, 张贵宾. 利用重力异常计算重力梯度的等效源技术[J]. 地球物理学进展, 2019, 34(4): 1398-1410.
CHEN Tao, ZHANG Guibin. Deriving the Full Gravitational Gradient Tensor from Gravity Anomaly: An Equivalent Source Technique[J]. *Progress in Geophysics*, 2019, 34(4): 1398-1410.

[17] 汪凤林, 蔡体菁, 王东霞. 惯性/重力匹配组合导航系统可观测性研究[J]. 安徽大学学报(自然科学版), 2008, 32(6): 27-31.
WANG Fenglin, CAI Tijing, WANG Dongxia. Inertial/Gravity Matching Integrated Navigation System[J]. *Journal of Anhui University (Natural Sciences)*, 2008, 32(6): 27-31.

[18] 杨靖宇, 张永生, 李正国, 等. 遥感影像正射纠正的GPU-CPU协同处理研究[J]. 武汉大学学报(信息科学版), 2011, 36(9): 1043-1046.
YANG Jingyu, ZHANG Yongsheng, LI Zhengguo, et al. GPU-CPU Cooperate Processing of RS Image Ortho-Rectification[J]. *Geomatics and Information Science of Wuhan University*, 2011, 36(9): 1043-1046.

[19] 陈召曦, 孟小红, 郭良辉, 等. 基于GPU并行的重力、重力梯度三维正演快速计算及反演策略[J]. 地球物理学报, 2012, 55(12): 4069-4077.
CHEN Zhaoxi, MENG Xiaohong, GUO Lianghui,

et al. Three-Dimensional Fast Forward Modeling and the Inversion Strategy for Large Scale Gravity and Gravimetry Data Based on GPU[J]. *Chinese Journal of Geophysics*, 2012, 55(12): 4069-4077.

[20] 刘文志. 并行算法设计与性能优化[M]. 北京: 机械工业出版社, 2015.
LIU Wenzhi. Parallel Computing and Performance Optimization[M]. Beijing: China Machine Press, 2015.

[21] 许大欣. 利用重力异常匹配技术实现潜艇导航[J]. 地球物理学报, 2005, 48(4): 812-816.
XU Daxin. Using Gravity Anomaly Matching Techniques to Implement Submarine Navigation[J]. *Chinese Journal of Geophysics*, 2005, 48(4): 812-816.

[22] 曹雪峰, 万刚, 张宗佩. Hilbert曲线层级演进关系[J]. 测绘学报, 2016, 45(S1): 77-84.
CAO Xuefeng, WAN Gang, ZHANG Zongpei. Hilbert Curve Hierarchical Evolution Relution[J]. *Acta Geodaetica et Cartographica Sinica*, 2016, 45(S1): 77-84.

[23] 刘辉, 冷伟, 崔涛. 高维Hilbert曲线的编码与解码算法设计[J]. 数值计算与计算机应用, 2015, 36(1): 42-58.
LIU Hui, LENG Wei, CUI Tao. Development of Encoding and Decoding Algorithms for High Dimensional Hilbert Curves[J]. *Journal on Numerical Methods and Computer Applications*, 2015, 36(1): 42-58.

[24] 贾连印, 陈明鲜, 李孟娟, 等. 基于状态视图的高效Hilbert编码和解码算法[J]. 电子与信息学报, 2020, 42(6): 1494-1501.
JIA Lianyin, CHEN Mingxian, LI Mengjuan, et al. State View Based Efficient Hilbert Encoding and Decoding Algorithms[J]. *Journal of Electronics & Information Technology*, 2020, 42(6): 1494-1501.

[25] 吴宇豪, 曹雪峰. 虚拟战场环境时空数据的Hilbert码索引方法[J]. 武汉大学学报(信息科学版), 2020, 45(9): 1403-1411.
WU Yuhao, CAO Xuefeng. Hilbert Code Index Method for Spatiotemporal Data of Virtual Battlefield Environment[J]. *Geomatics and Information Science of Wuhan University*, 2020, 45(9): 1403-1411.

[26] 孙立伟, 袁昱纬, 周俊芳. 一种采用Hilbert空间排列码的场景数据调度策略[J]. 无线电工程, 2019, 49(9): 775-778.
SUN Liwei, YUAN Yuwei, ZHOU Junfang. A Scene Data Scheduling Strategy Using Hilbert Space Arrangement Codes[J]. *Radio Engineering*, 2019, 49(9): 775-778.

[27] ODERA P A, FUKUDA Y, KUROISHI Y. A High-Resolution Gravimetric Geoid Model for Japan from EGM2008 and Local Gravity Data[J]. *Earth, Planets and Space*, 2012, 64(5): 361-368.

[28] 范宏涛, 郭春喜, 王小华, 等. 超高阶重力场模型EIGEN-6C2适应性分析[J]. 测绘科学, 2015, 40(9): 18-22.
FAN Hongtao, GUO Chunxi, WANG Xiaohua, et al. Adaptability Analysis on Ultra-High-Gravity Model EIGEN-6C2 in Chinese Coastal Zone[J]. *Science of Surveying and Mapping*, 2015, 40(9): 18-22.