



基于图形处理器加速数值求解三维含时薛定谔方程

唐富明 刘凯 杨溢 屠倩 王凤 王哲 廖青

Numerical solution of three-dimensional time-dependent Schrödinger equation based on graphic processing unit acceleration

Tang Fu-Ming Liu Kai Yang Yi Tu Qian Wang Feng Wang Zhe Liao Qing

引用信息 Citation: *Acta Physica Sinica*, 69, 234202 (2020) DOI: 10.7498/aps.69.20200700

在线阅读 View online: <https://doi.org/10.7498/aps.69.20200700>

当期内容 View table of contents: <http://wulixb.iphy.ac.cn>

您可能感兴趣的其他文章

Articles you may be interested in

基于多组态含时Hartree-Fock方法研究电子关联对于H₂分子强场电离的影响

Multiconfiguration time-dependent Hartree-Fock treatment of electron correlation in strong-field ionization of H₂ molecules

物理学报. 2018, 67(10): 103301 <https://doi.org/10.7498/aps.67.20172701>

非线性薛定谔方程的高阶分裂改进光滑粒子动力学算法

Numerical study of nonlinear Schrödinger equation with high-order split-step corrected smoothed particle hydrodynamics method

物理学报. 2019, 68(9): 090203 <https://doi.org/10.7498/aps.68.20190169>

基于修正并行光滑粒子动力学方法三维变系数瞬态热传导问题的模拟

Simulation of three-dimensional transient heat conduction problem with variable coefficients based on the improved parallel smoothed particle hydrodynamics method

物理学报. 2017, 66(13): 130201 <https://doi.org/10.7498/aps.66.130201>

四阶色散非线性薛定谔方程的明暗孤立波和怪波的形成机制

Study on the generation mechanism of bright and dark solitary waves and rogue wave for a fourth-order dispersive nonlinear Schrödinger equation

物理学报. 2020, 69(1): 010502 <https://doi.org/10.7498/aps.69.20191384>

强场非次序双电离中再碰撞动力学的强度依赖

Intensity-dependent recollision dynamics in strong-field nonsequential double ionization

物理学报. 2019, 68(3): 033201 <https://doi.org/10.7498/aps.68.20181811>

三维电磁扩散场数值模拟及磁化效应的影响

Three-dimensional numerical simulation of electromagnetic diffusion problem and magnetization effects

物理学报. 2019, 68(3): 030201 <https://doi.org/10.7498/aps.68.20181567>

基于图形处理器加速数值求解三维含时薛定谔方程*

唐富明 刘凯 杨溢 屠倩 王凤 王哲 廖青†

(武汉工程大学, 光学信息与模式识别湖北省重点实验室, 武汉 430205)

(2020年5月11日收到; 2020年7月10日收到修改稿)

量子力学领域中对强激光场与原子分子相互作用的理论研究非常依赖于数值求解含时薛定谔方程. 本文在强场电离的背景下并行求解氢原子的三维含时薛定谔方程. 基于球极坐标系, 采用分裂算符-傅里叶变换方法将含时薛定谔方程进行了离散化. 由此可得到长度规范下的光电子连续态波函数. 图形处理器 (GPU) 可以依托多线程结构充分发挥细粒度并行的优势, 实现整体算法的并行加速. 计算表明, 相对于中央处理器 (CPU), GPU 并行计算有着最高约 60 倍的加速比. 由此可见, 基于 GPU 加速数值求解三维含时薛定谔方程能够显著缩短计算耗费的时间. 这一工作对利用 GPU 快速求解三维含时薛定谔方程有着重要的指导意义.

关键词: 三维含时薛定谔方程, 强场电离, 并行计算

PACS: 42.50.Hz, 32.80.Fb, 02.70.-c

DOI: 10.7498/aps.69.20200700

1 引言

强场激光物理是过去二十多年随着超短强激光技术的发展而快速发展起来的前沿学科, 主要研究超强超短激光与电子的相互作用. 早在 1917 年, 爱因斯坦的辐射理论就提出了受激辐射的基本概念, 预测到光可以产生受激辐射. 直至 1960 年, 世界上第一台激光器诞生, 此后的几十年间激光技术飞速发展, 从自由输出到调 Q 技术 (Q -switch)、锁模技术 (mode locking), 再到啁啾脉冲放大技术, 激光的脉冲宽度越来越短, 功率越来越大. 利用先进激光技术获得的超快强激光脉冲与物质相互作用, 成为了研究物质基本性质的一种重要手段. 其中, 强场中一些非线性现象, 如高次谐波的产生 (HHG)^[1-5]、次序和非次序双电离^[6-8]等, 受到了广泛的关注.

强场激光物理的主要理论方法是数值求解含时薛定谔方程 (TDSE)^[9-11]、强场近似 (SFA)^[12-14]和半经典模型^[15-17]. 求解三维含时薛定谔方程得到的结果可以认为相当于发生在数值上的实验. 但是求解三维含时薛定谔方程并非一项简单的任务, 无法获得解析解, 只能借助计算机数值求解. 在直角坐标系下哈密顿算符的表示很直观, 但由于正交网格在三个维度上均较为致密, 因此对存储空间和计算量的需求十分巨大. 而在球极坐标系中只有径向网格较为致密, 另外两个维度相对稀疏. 尽管如此, 求解三维含时薛定谔方程的计算量也是十分巨大. 为了缩短计算的时间, 更快地得到计算结果, 就有必要使用并行方法去加速计算.

在过去的一段时间里, 对基于图形处理器 (GPU) 并行求解三维含时薛定谔方程的优化与加速主要体现在两个方面. 一方面, 不断改进 TDSE 算法或者引入其他方法以减小计算量, 如在柱坐标

* 国家自然科学基金 (批准号: 11674257, 11604248, 11874019, 11947096) 和湖北省高等学校优秀中青年科技创新团队计划 (批准号: T201806) 资助的课题.

† 通信作者. E-mail: liaoqing@wit.edu.cn

系下采用的混合数值迭代格式^[18]. 另一方面, 采用不同的并行平台, 比如: 使用 CUDA (compute unified device architecture) 平台编写 TDSE 程序^[19], 使用 OpenCL(open computing language) 语言编写 TDSE 程序^[20–22]. 但在 CUDA 架构或 OpenCL 架构下对算法并行计算进行加速, 使用起来难度较高.

基于硬件的不同, 并行加速有中央处理器 (CPU) 并行和 GPU 并行两种方法. CPU 中多条指令构成指令流水线, 且每个线程都有独立的硬件来操纵整个指令流. 采用复杂的分支预测技术来达到并行计算目的. GPU 是图形处理器, 图形运算的特点是大量同类型数据的密集运算——如图形数据的矩阵运算, 正因为如此, GPU 的微架构就是面向适合于矩阵类型的数值计算而设计的, 这类计算可以分成众多独立的数值计算——大量数值运算的线程, 而且数据之间没有像程序执行的那种逻辑关联性. GPU 并行计算的崛起得益于大数据时代的到来, 而传统的 CPU 并行计算已经远远不能满足大数据的需求. CPU 的核处理器相对较少, 并行的效率较低. GPU 最大的特点是拥有超多计算核心, 往往成千上万核. 而每个核心都可以模拟一个 CPU 核心的计算功能. 因此 GPU 的并行效率比 CPU 并行效率高.

本文基于 GPU 加速数值求解三维含时薛定谔方程, 通过在 GPU 上并行加速求解三维含时薛定谔方程, 并且与 CPU 并行加速做对比, 得到 GPU 相对于 CPU 的加速比. 在 GPU 加速的应用程序中, 工作负载的顺序部分运行在 CPU 上 (这是为单线程性能优化的), 而应用程序的计算密集型部分并行运行在数千个 GPU 内核上. 正是如此, GPU 并行计算可以得到很好的加速效果.

2 理论方法

2.1 三维含时薛定谔方程求解方案

物理系统的时间演化由波函数 $\Psi(t)$ 来描述, 满足含时薛定谔方程

$$i\frac{\partial}{\partial t}\Psi(t) = H(t)\Psi(t). \quad (1)$$

本文以线偏振激光电场作用下的氢原子为例, 在球极坐标中系统的哈密顿量为 (如无特殊说明, 下文一律使用原子单位):

$$H(t) = T + V + W(r, \theta, t) \\ = -\frac{1}{2}\frac{\partial^2}{\partial r^2} + \frac{L^2}{2r^2} - \frac{1}{r} - E(t)r\cos\theta, \quad (2)$$

这里 $H(t)$ 指作用在约化波函数 $\Phi(t) = r\Psi(t)$ 上的算符, 其中 L^2 为系统的总角动量算符, $E(t)$ 为线偏振入射激光电场.

将波函数 $\Psi(r, t)$ 在球谐函数 $Y_{lm}(\theta, \varphi)$ 下展开,

$$\Psi(r, t) = \frac{1}{r} \sum_{l=0}^{\infty} \sum_{m=-l}^l \Phi_{lm}(t) Y_{lm}(\theta, \varphi). \quad (3)$$

由于 m 是一个好量子数, 通过对方位角 φ 进行积分并把角动量算符数值求解. 含时薛定谔方程的一般步骤是先给定一个初始波函数 $\Phi(t=0)$, 然后将时间演化算符作用在该初始波函数上, 反复迭代直至得到任意时刻的末态波函数. 假设将波函数向前推进 Δt 时间增量, 那么 t 时刻和 $t + \Delta t$ 时刻的波函数关系为

$$\Phi(t + \Delta t) = e^{-iH(t)\Delta t}\Phi(t). \quad (4)$$

利用分裂算符法^[23] 将 (2) 式代入 (4) 式, 可以得到

$$\Phi(t + \Delta t) = e^{-i\frac{\Delta t}{2}T} e^{-i\frac{\Delta t}{2}V} e^{-i\Delta tW(r, \theta, t)} e^{-i\frac{\Delta t}{2}V} \\ \times e^{-i\frac{\Delta t}{2}T}\Phi(t) + O(\Delta t^3), \quad (5)$$

这里产生的变换误差为 Δt 的三阶误差项. 由于 m 是一个好量子数, 通过对方位角 φ 进行积分并把角动量算符 L_z 替换为它的本征值 m , 可以使问题大为简化. 为了方便起见, 下文只讨论 $m=0$ 的情况. 假设波函数可以在有限勒让德多项式下展开:

$$\Phi(r_i, x_j, t) = \sum_{l=0}^L f_l(r_i, t) P_l(x_j), \quad x_j = \cos\theta_j. \quad (6)$$

P_l 为球谐函数 Y_{l0} , $f_l(r_i, t)$ 可以由高斯-勒让德求积法数值计算得出,

$$f_l(r_i, t) = \sum_{j=1}^{L+1} w_j P_l(x_j) \Phi(r_i, x_j, t), \quad (7)$$

这里 x_j 为勒让德多项式 $P_{L+1}(x_j)$ 的 $L+1$ 个零点, w_j 为与之对应的求积权重. 对于 l 的不同取值, 一维径向波函数 $f_l(r_i, t)$ 均定义在 R 点等距径向网格上, 它的网格表示为

$$f_l(t) = \{f_l(r_0, t), f_l(r_1, t), \dots, f_l(r_{R-1}, t)\}^T, \\ r_i - r_{i-1} = \Delta r, \quad i = 1, 2, \dots, R-1. \quad (8)$$

(4) 式中时间演化算符的作用可以分为三步.

1) 径向动能算符 $T = (-1/2)\partial^2/\partial r^2$ 独立作用在每个一维径向波函数 $f_l(r_i, t)$ 上, 利用带边界条件的傅里叶谱方法^[24], 得到

$$e^{-i\frac{\Delta t}{2}T} f_l(r, t) \doteq \mathcal{F}^{\text{sin}} \cdot \text{diag}\left(e^{-i\frac{\Delta t}{2}\frac{k^2}{2}}\right) \cdot \mathcal{F}^{\text{sin}} \cdot f_l(t),$$

$$k_j = \frac{\pi}{R\Delta r} j. \quad (9)$$

上述操作中, 由于对所有径向波函数都进行了一维正弦变换, 应用径向动能算符的计算复杂度为 $O(R\log_2(R)L)$.

2) 等效势能算符 V 分为离心势能项 $L^2/(2r^2)$ 和库仑势能项 $-1/r$, 其中总角动量算符 L^2 在球谐函数表象下是对角化的, 对角元为它的本征值 $l(l+1)$, 因此可以得到

$$e^{-i\frac{\Delta t}{2}V} f_l(r_i, t) \approx e^{i\frac{\Delta t}{2}\left[\frac{l(l+1)}{2r_i^2} - \frac{1}{r_i}\right]} f_l(r_i, t),$$

$$i = 0, 1, 2, \dots, R-1. \quad (10)$$

3) 相互作用势能算符 $W(r, \theta, t)$ 在坐标表象 $\{r, \theta\}$ 下是对角化的, 利用 (5) 式重构波函数 $\Phi(r, x, t)$, 得到

$$e^{-i\frac{\Delta t}{2}W(r, \theta, t)} \Phi(r, x, t) \doteq e^{i\Delta t E(t)r_i x_j} \cdot \Phi(r_i, x_j, t). \quad (11)$$

上述对应算法步骤如表 1 所列.

通过 (6) 式再次展开成径向波函数 $f_l(r_i, t)$, 接下来只需要依次执行步骤 2) 和步骤 1), 就完成了波函数向前推进 Δt 的一次迭代. 在含时演化中, 重复执行上述步骤, 能够得到任意时刻的末态波函数. 步骤 1) 中的快速傅里叶变换操作以及 (5) 式和 (6) 式中的变换操作都可以通过向量化提高计算效率.

连续态波函数在球谐函数表象下可以写成

$$\psi_k(r) = \frac{1}{k\sqrt{2\pi}} \sum_{l,m} e^{i(\sigma_l + \delta_l)} \mathbf{Y}_{lm}^*(\theta, \varphi)$$

$$\times \Phi_{lm}(r, t) \mathbf{Y}_{l'm'}(\theta', \varphi'), \quad (12)$$

其中是 $\sigma_l = \arg \Gamma(1 + l - iZ/k)$ 是库仑散射相移, δ_l 是除去长程库仑势的短程势产生的相移.

电子与电场相互作用后的末态动量分布是

$$\frac{dP}{dk} = |\langle \psi_k(r) | \Psi(r, t) \rangle|^2$$

$$= \left| \frac{1}{k\sqrt{2\pi}} \sum_{l,m} (-i)^l e^{i(\sigma_l + \delta_l)} \mathbf{Y}_{lm}(\theta, \varphi) \right.$$

$$\left. \times \int_0^\infty dr \Phi_{lm}(r, t) r f_l(r, t) \right|^2. \quad (13)$$

表 1 TDSE 算法步骤
Table 1. TDSE algorithm steps.

算法	$\Phi(t + \Delta t) = e^{-iH(t)\Delta t}\Phi(t)$
Input:	$f_l(r_i, t)$
Output:	$f_l(r_i, t)$
1. for	n do
2. for	l do
3.	$f_l(r_i, t) = \text{ifft}\left(\text{diag}\left(e^{-i\frac{\Delta t}{2}\frac{k^2}{2}}\right) \cdot \text{fft}(f_l(r_i, t))\right)$
4. end for	
5. for	i and l do
6.	$f_l(r_i, t) = e^{-i\frac{\Delta t}{2}\left[\frac{l(l+1)}{2r_i^2} - \frac{1}{r_i}\right]} \cdot f_l(r_i, t)$
7. end for	
8. for	i and j do
9.	$\Phi(r_i, x_j, t) = \sum_{l=0}^L f_l(r_i, t) P_l(x_j)$
10. end for	
11. for	i and j do
12.	$ \Phi(r_i, x_j, t) = e^{i\Delta t E(n)r_i x_j} \cdot \Phi(r_i, x_j, t) $
13. end for	
14. for	i and j do
15.	$f_l(r_i, t) = \sum_{j=1}^{L+1} w_j P_l(x_j) \Phi(r_i, x_j, t)$
16. end for	
17. for	i and l do
18.	$f_l(r_i, t) = e^{-i\frac{\Delta t}{2}\left[\frac{l(l+1)}{2r_i^2} - \frac{1}{r_i}\right]} \cdot f_l(r_i, t)$
19. end for	
20. for	l do
21.	$f_l(r_i, t) = \text{ifft}\left(\text{diag}\left(e^{-i\frac{\Delta t}{2}\frac{k^2}{2}}\right) \cdot \text{fft}(f_l(r_i, t))\right)$
22. end for	
23. end for	

2.2 并行计算方案

本文在 Matlab 环境下调用设备端 (GPU) 来实现并行计算, 离散化的初始波函数以及传播算符均保存在主机端 (CPU), 如图 1 所示. GPU 上数组的创建和传输通过 Matlab 并行计算工具箱的相关函数完成, 使用 `gpuArray()` 函数从主机端向设备端发送 Matlab 数组, 即将 Matlab 工作区内的数组传输到设备端内存. 在 GPU 中进行数值计算完毕以后, 再通过 `gather()` 函数从设备端向主机端发送 Matlab 数组, 即将设备端内的 Matlab 数组传输到主机端内存.

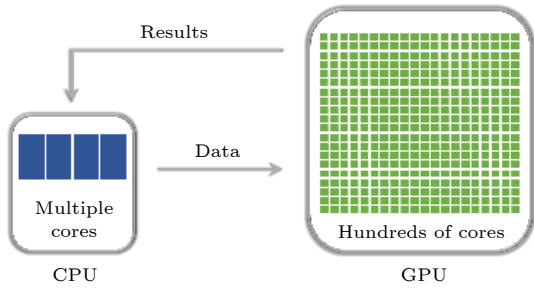


图 1 数据传输流程图

Fig. 1. The flowchart of data transmission.

在求解含时薛定谔方程时, 具体需要并行计算的部分是: 快速傅里叶变换、逆快速傅里叶变换、矩阵乘法和数组点乘. 将需要进行变换的数组从 CPU 传输到 GPU, 然后在 GPU 上对数组按列分别进行快速傅里叶变换. 不同的列在进行快速傅里叶变换时是相互独立的, 所以可以在不同的线程内对不同的列同时进行快速傅里叶变换, 这样就达到了并行的目的. 计算得到的数组存储在 GPU 内存中, 将计算结果传输回 CPU 后, 可以释放在 GPU 中占用的内存. 逆快速傅里叶变换与快速傅里叶变换类似. 矩阵乘法的思路是将左乘矩阵的任一行和右乘矩阵的任一列做点积, 得到目标矩阵的任一元素. 每个目标矩阵元素的计算都是相互独立的, 因此可以同时计算多个目标矩阵元素, 以达到并行计算的目的. 数组点乘是 Matlab 中的一种运算方式, 直接对两个数组中相同位置的元素做乘法, 即可得到目标数组的对应元素. 因此, 数组点乘本质上是一种标量运算, 非常便于利用多线程进行并行计算.

3 结果与讨论

3.1 测试环境与测试算例

本文所有计算使用的并行环境包含 1 个 CPU (Intel Xeon E7-8880 v4, 22 核, 主频 2.2 GHz) 和 NVIDIA Tesla P100 卡中的 1 个 GPU (3584 核心, 主频 1.3 GHz). 程序的实现基于 Matlab 环境, GPU 中的矩阵向量操作调用了重载的 Matlab 函数. 本次测试算例以线偏振红外激光电场作用下氢原子的阈上电离为背景. 红外激光电场强度 $I = 9 \times 10^{13} \text{ W/cm}^2$, 波长 $\lambda = 800 \text{ nm}$, 具有正弦平方脉冲包络, 共 8 个光学周期. 其中计算采用的空间范围 $r_{\text{max}} = 1000 \text{ a.u.}$, 时间步长 $\Delta t = 0.0037 \text{ a.u.}$

3.2 实验结果

为获取 GPU 并行的优化表现, 需要给出 CPU 上的并行性能作为基准. 通过改变角量子数 L 和径向网格点 R 计算演化时间结束后的末态波函数, 比较不同参数下 GPU 相对于 CPU 的加速比 (单个 CPU 计算时间与单个 GPU 计算时间的比值):

$$S = \frac{T_{\text{CPU}}}{T_{\text{GPU}}}. \quad (14)$$

首先, 设置径向网格点 $R = 1024 \times 32$ 不变, 选取不同的角量子数 L , 计算结果如表 2 和图 2 所示.

表 2 不同角量子数下 CPU 与 GPU 的计算时间比较
Table 2. Computation time of CPU and GPU under different angular quantum numbers.

角量子数 L	计算时间/s	
	CPU	GPU
4	2164.309	159.368
9	4120.602	164.418
19	7922.537	205.440
39	17682.308	378.104
79	36774.347	757.198

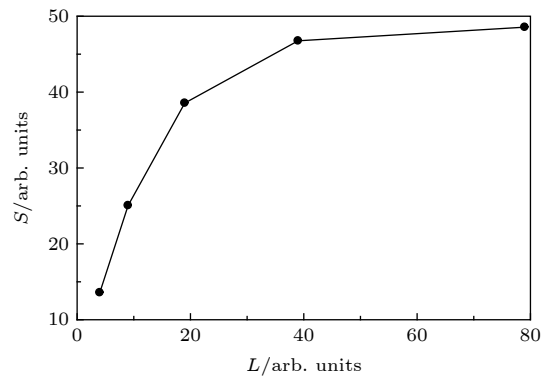


图 2 加速比随着角量子数的变化

Fig. 2. Speedup ratio as a function of angular quantum number.

角量子数 $L = 19$ 保持不变, 改变径向网格点 R 的大小, 然后将 CPU 与 GPU 的计算时间对比, 如表 3 和图 3 所示.

同时改变径向网格点和角量子数的大小, 也就是改变径向波函数集 $\{f_l(r_i, t)\}$ 的矩阵大小, 比较 CPU 和 GPU 的计算时间, 结果如表 4 和图 4 所示.

此外, 为了进一步测试 GPU 的加速性能, 又进行了一组 1600 nm 中红外激光的实验, 中红外激光电场强度 $I = 1 \times 10^{13} \text{ W/cm}^2$, 波长 $\lambda = 1600 \text{ nm}$,

具有正弦平方脉冲包络, 共 8 个光学周期. 其中计算采用的空间范围 $r_{\max} = 1000$ a.u., 时间步长 $\Delta t = 0.0037$ a.u.. 此时与第三组实验相同, 同时改变径向网格点和角量子数的大小, 也就是改变径向波函数集 $\{f_l(r_i, t)\}$ 的矩阵大小, 比较 CPU 和 GPU 的计算时间, 结果如表 5 和图 5 所示.

选取一组 800 nm 激光电场作用下氢原子的阈上电离, 由 CPU 计算得到的末态波函数与 GPU 的计算结果对比, 如图 6 所示, 电子末态动量分布保持一致.

表 3 不同径向网格点下 CPU 与 GPU 的计算时间比较

Table 3. Computation time of CPU and GPU under different radial grid points.

径向网格点数 R	计算时间/s	
	CPU	GPU
2^{12}	1118.348	148.302
2^{13}	1871.128	154.614
2^{14}	3846.120	160.763
2^{15}	7922.537	205.440
2^{16}	16862.467	354.554

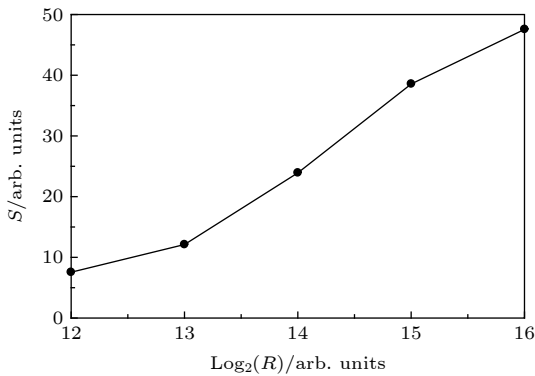


图 3 加速比随着径向网格点的变化

Fig. 3. Speedup ratio as a function of radial grid point.

表 4 不同矩阵大小下 CPU 与 GPU 的计算时间比较

Table 4. Computation time of CPU and GPU under different matrix sizes.

矩阵大小	计算时间/s	
	CPU	GPU
5×2^{12}	199.158	149.895
10×2^{13}	965.276	166.039
20×2^{14}	3846.120	160.763
40×2^{15}	17682.308	378.104
80×2^{16}	74761.695	1524.669

同时计算两种方法得到的约化波函数的误差为 5×10^{-21} , 动量分布误差为 2×10^{-18} . 其中波函数误差计算公式为

$$\sum_{l=0}^L \sum_{r=0}^r |(f_l(r_i, t)_{\text{CPU}} - f_l(r_i, t)_{\text{GPU}})|^2,$$

动能谱的误差计算公式为

$$\sum_{jx=0}^{200} \sum_{jz=0}^{\pi} |(YYS(jx, jz)_{\text{CPU}} - YYS(jx, jz)_{\text{GPU}})|^2.$$

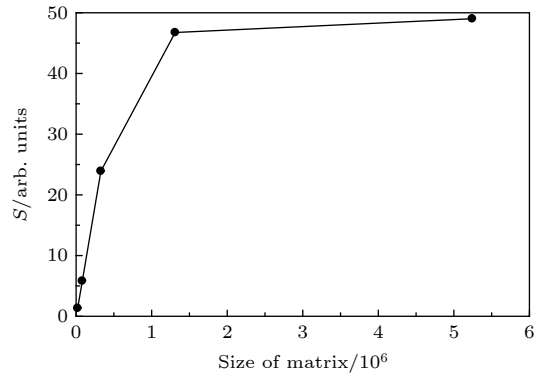


图 4 加速比随着矩阵大小的变化

Fig. 4. Speedup ratio as a function of the size of matrix.

表 5 不同矩阵大小下 CPU 与 GPU 的计算时间比较

Table 5. Computation time of CPU and GPU under different matrix sizes.

矩阵大小	计算时间/s	
	CPU	GPU
5×2^{12}	437.584	315.448
10×2^{13}	2075.667	463.183
20×2^{14}	9252.539	629.088
40×2^{15}	40617.723	814.985
80×2^{16}	182135.643	3024.669

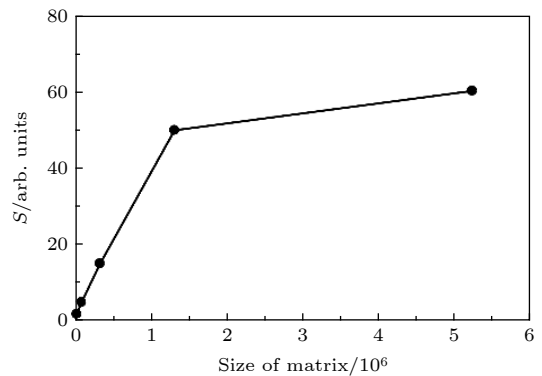


图 5 加速比随着矩阵大小的变化

Fig. 5. Speedup ratio as a function of the size of matrix.

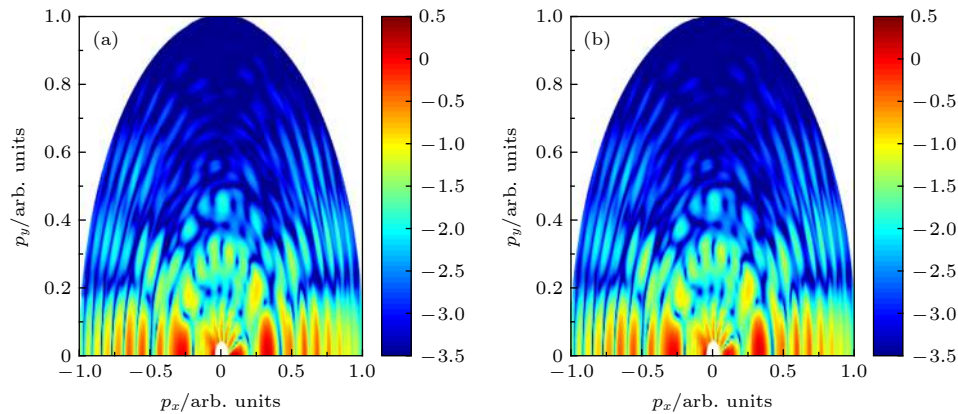


图 6 氢原子的光电子末态动量分布 (a) CPU 计算结果; (b) GPU 计算结果

Fig. 6. Photoelectron final-state momentum distributions of hydrogen atom: (a) Calculation results of CPU; (b) calculation results of GPU.

3.3 实验结果讨论

通过上述的计算可以得出: 1) 相同计算量下 CPU 与 GPU 的计算时间有很大的差距, 当计算量较小时加速比急剧升高, 随着计算量的增大加速比趋于一个稳定值, 最高达到了约 60 倍的加速提升, 加速效果十分明显; 2) 从不同的计算量对比也可以看出, 当计算量越大时, 加速的效果也就越好. 并不是所有的计算都要选取 GPU 来计算, CPU 将数据传输到 GPU 需要一定的时间, 当数据比较大时, 采用 GPU 来并行计算, 能够获得更大的加速比; 3) 如果只是改变一个维度的大小, 所得到的实际加速效果有时候并不理想, 这和 CPU 以及 GPU 存储数据、读取数据的方式有关. 所以在计算的时候为了获得最好的加速比, 需要同时在至少两个维度上调整矩阵的大小; 4) 无论采用 CPU 计算还是 GPU 计算, 都能得到相同的计算结果, 计算误差也在可接受的范围内, 并且该结果也符合现有的阈上电离物理规律.

4 结 论

本文详细分析了在强场电离的背景下数值求解氢原子三维含时薛定谔方程基于不同硬件的并行速度. 借助于分裂算符-傅里叶变换方法, 在球极坐标下得到了三维含时薛定谔方程的末态解. 同时, 依托于 GPU 的多线程结构, 使得 GPU 发挥细粒度方面的并行优势, 实现整体算法的并行加速. 采用了 CPU 并行和 GPU 并行两种加速计算模式, 探讨了两者并行加速的性能. 通过与现有的物理规

律相对比, 验证了程序的正确性. 计算结果表明, 当计算量较小时 GPU 相对于 CPU 的加速效果不突出, 随着计算量的增大加速比迅速增加, 然后趋于一个稳定值. GPU 并行对于数值求解三维含时薛定谔方程有着相对于 CPU 最高约 60 倍的加速. 可见, 计算量越大, 采用 GPU 并行获得的加速比越大. 这一工作对利用 GPU 高效数值求解含时薛定谔方程有着重要的指导意义.

参考文献

- [1] Corkum P 1993 *Phys. Rev. Lett.* **71** 1994
- [2] Liu K L, Luo S Q, Li M, Li Y, Feng Y D, Du B J, Zhou Y M, Lu P X, Barth I 2019 *Phys. Rev. Lett.* **122** 053202
- [3] Lewenstein M, Balcou P, Ivanov M Y, L'Huillier A, Corkum P B 1994 *Phys. Rev. A* **49** 2117
- [4] Zhang X F, Zhu X S, Wang D, Li L, Liu X, Liao Q, Lan P F, Lu P X 2019 *Phys. Rev. A* **99** 013414
- [5] Gaarde M B, Tate J L, Schafer K J 2008 *J. Phys. B* **41** 132001
- [6] Liu K, Qin M Y, Li Q G, Liao Q 2018 *Opt. Quantum Electron.* **50** 364
- [7] Liao Q, Li Y, Qin M Y, Lu P X 2017 *Phys. Rev. A* **96** 063408
- [8] Liu K, Wang F, Wang Z, Qin M Y, Liao Q 2019 *J. Opt. Soc. Am. B* **36** 2624
- [9] Muller H G 1999 *Laser Phys.* **9** 138
- [10] Bauer D, Koval P 2006 *Comput. Phys. Commun.* **174** 396
- [11] Madsen L B, Nikolopoulos L A A, Kjeldsen T K, Fernández J 2007 *Phys. Rev. A* **76** 063407
- [12] Keldysh L V 1964 *Sov. Phys. JETP* **20** 1307
- [13] Faisal F H M 1973 *J. Phys. B: At. Mol. Opt. Phys.* **6** L89
- [14] Reiss H R 1980 *Phys. Rev. A* **22** 1786
- [15] Gallagher T F 1988 *Phys. Rev. Lett.* **61** 2304
- [16] Corkum P B, Burnett N H, Brunel F 1989 *Phys. Rev. Lett.* **62** 1259
- [17] Xiao X R, Wang M X, Li M, Geng J W, Liu Y Q, Peng L Y 2016 *Acta Phys. Sin.* **65** 220203 (in Chinese) [肖相如, 王慕雪, 黎敏, 耿基伟, 刘运全, 彭良友 2016 物理学报 **65** 220203]
- [18] Gainullin I 2017 *Comput. Phys. Commun.* **72** 210
- [19] Liu Q, Liu F, Hou C 2020 *Procedia Computer Sci.* **171** 312

- [20] Penfold T J 2017 *Phys. Chem. Chem. Phys.* **19** 19601 185 1791
[21] Broin C Ó 2015 *Ph. D. Dissertation* (Dublin: Dublin City University)
[22] Broin C Ó, Nikolopoulos L A A 2014 *Comput. Phys. Commun.*
[23] Feit M D, Fleck J A, Steiger A 1982 *J. Comput. Phys.* **47** 412
[24] Kjeldsen T K 2007 *Ph. D. Dissertation* (Aarhus: University of Aarhus)

Numerical solution of three-dimensional time-dependent Schrödinger equation based on graphic processing unit acceleration^{*}

Tang Fu-Ming Liu Kai Yang Yi Tu Qian Wang Feng
Wang Zhe Liao Qing[†]

(*Hubei Key Laboratory of Optical Information and Pattern Recognition, Wuhan Institute of Technology, Wuhan 430205, China*)

(Received 11 May 2020; revised manuscript received 10 July 2020)

Abstract

In the field of quantum mechanics, the theoretical study of the interaction between intense laser field and atoms and molecules depends very much on the numerical solution of the time-dependent Schrödinger equation. However, solving the three-dimensional time-dependent Schrödinger equation is not a simple task, and the analytical solution cannot be obtained, so it can only be solved numerically with the help of computer. In order to shorten the computing time and obtain the results quickly, it is necessary to use parallel methods to speed up computing. In this paper, under the background of strong field ionization, the three-dimensional time-dependent Schrödinger equation of hydrogen atom is solved in parallel, and the suprathreshold ionization of hydrogen atom under the action of linearly polarized infrared laser electric field is taken for example. Based on the spherical polar coordinate system, the time-dependent Schrödinger equation is discretized by the splitting operator-Fourier transform method, and the photoelectron continuous state wave function under the length gauge can be obtained. In Graphics processing unit (GPU) accelerated applications, the sequential portion of the workload runs on central processing unit (CPU) (which is optimized for single-threaded performance), while the compute-intensive part of the application runs in parallel on thousands of GPU cores. The GPU can make full use of the advantage of fine-grained parallelism based on multi-thread structure to realize parallel acceleration of the whole algorithm. Two accelerated computing modes of CPU parallel and GPU parallel are adopted, and their parallel acceleration performance is discussed. Compared with the results from the existing physical laws, the calculation error is also within an acceptable range, and the result is also consistent with the result from the existing physical laws of suprathreshold ionization, which also verifies the correctness of the program. In order to obtain a relatively accurate acceleration ratio, many different experiments are carried out. Computational experiments show that under the condition of ensuring accuracy, the GPU parallel computing speeds by up to about 60 times maximally based on the computational performance of CPU. It can be seen that the accelerated numerical solution of three-dimensional time-dependent Schrödinger equation based on GPU can significantly shorten the computational time. This work has important guiding significance for rapidly solving the three-dimensional time-dependent Schrödinger equation by using GPU.

Keywords: three-dimensional time-dependent Schrödinger equation, strong-field ionization, parallel computing

PACS: 42.50.Hz, 32.80.Fb, 02.70.-c

DOI: 10.7498/aps.69.20200700

^{*} Project supported by the National Natural Science Foundation of China (Grant Nos. 11674257, 11604248, 11874019, 11947096) and the Program for Distinguished Middle-aged and Young Innovative Research Team in Higher Education of Hubei Province, China (Grant No. T201806).

[†] Corresponding author. E-mail: liaoqing@wit.edu.cn