


这是 Google 对 <http://blog.sciencenet.cn/blog-3102863-971052.html> 的缓存。 这是该网页在 2020年8月3日 11:39:40 GMT 的快照。 [当前页](#)在此期间可能已经更改。 [了解详情](#).

[完整版本](#) [纯文字版本](#) [查看源代码](#)

提示: 要在此页面上快速找到您的搜索字词, 请按 **Ctrl+F** 或者 **⌘-F** (Mac), 然后使用查找栏搜索。

 构建全球华人科学博客圈 | [返回首页](#) | [微博](#) | [RSS订阅](#) | [帮助](#)

[注册](#) | [登录](#)

brucefan1983的个人博客  [分享](#)

<http://blog.sciencenet.cn/u/brucefan1983>

[博客首页](#)

[动态](#)

[博文](#)

[相册](#)

[好友](#)

[留言板](#)

博文

A 100-line matlab code for molecular dynamics (fast version)

已有 8258 次阅读 2016-4-19 05:25 | 个人分类:计算物理基础 | 系统分类:科研笔记 |  分子动力学模拟

改进matlab分子动力学模拟程序的性能

樊哲勇

<brucenju(at)gmail.com> <zheyongfan(at)163.com>

最后修改: 2016年4月24日 (上传了所有代码)

 [find_force_vectorized.m](#)



樊哲勇

 加为好友

 给我留言

 打个招呼

 发送消息

扫一扫, 分享此博文

 find_neighbor.m

 initialize_position.m

 initialize_velocity.m

 md_vectorized.m

 test_md_vectorized.m

一、引言

在前一篇博文中，我介绍了分子动力学模拟的基本技术并给出了一个完整的matlab程序。该程序除去注释之后还不到100行，“体积”上可谓小巧玲珑。可惜的是，该程序在计算性能上并不理想：让一个含有256个氩原子的体系演化2000步，在我的笔记本上用了大概一分钟的时间。本文介绍如何用矢量化编程的技巧提升程序性能。

二、问题分析

用matlab很容易写出很慢的程序，就像上一篇博文中给出的那个程序那样。提高matlab程序的技巧有不少，但这里只关注一点：那就是尽量用矢量化的方式代替循环的方式进行编



作者的精选博文 [全部](#)

- 青年基金要结题了，清点一下
- 一篇Carbon论文的审稿意见和
- 分享一篇Nano Lett.文章的
- 关于论文写作、投稿以及审稿

作者的其他最新博文 [全部](#)

- 在 Windows 编译 GPUMD
- GPUMD 学习笔记
- GPUMD 的官方网站
- GPUMD 的 mailing list
- 准备写一本分子动力学模拟的
- 2019年发表论文合集

精选博文导读 [全部](#)

- 211硕士毕业作，第一篇1区...
- 获取基金的新策略？
- 对即将到来的博士生活的几...
- 疫情下学位论文的问题
- Two stressful weeks and we...
- 2021年科技部出版基金，有...

相关博文

- 西安交通大学第六届丝绸之...

程。在上文中的代码中，我也用了不少矢量编程的技巧。例如，速度-Verlet积分第一部分的代码只有两行，都采用了矢量化方式进行编程：

```
v(:,d)=v(:,d)+(f(:,d)./m)*(dt*0.5);  
r(:,d)=r(:,d)+v(:,d)*dt;
```

如果不用矢量化的方式，而用普通的循环方式，代码可以写为：

```
for n=1:N  
    v(n,d)=v(n,d)+(f(n,d)./m(n))*(dt*0.5);  
    r(n,d)=r(n,d)+v(n,d)*dt;  
end
```

然而，这些部分代码的优化并不能带来程序整体性能上的提升，因为程序最耗时的部分并不在积分这一块。分子动力学模拟中最耗时的部分是求力的函数；它的计算时间可以占总的计算时间的99%以上。要想提升程序性能，就得优化求力的函数。

三、矢量化的求力函数

下面先复制上篇博文中求力的函数中的部分代码（循环部分）：

宣友超群, 张世, 徐中, 问题

- 北京工业大学第四届国际青...
- 直播预告 | 多位锂离子电池...
- 中国学者与国外学者简介的...
- 两类历史学者

```

for n1=1:N-1 % loop over the atoms
    for m=1:NN(n1) % loop over the neighbors of atom n1
        n2=NL(n1,m); % (n1, n2) is a pair of atoms
        if n2<n1;continue;end % Use Newton's 3rd law to speed up
        r12=r(n2,:)-r(n1,:);
        r12=r12-round(r12./L).*L_times_pbc; % minimum image convention
        d12_square=sum(r12.*r12);
        d_6=d12_square^3;d_8=d12_square*d_6;d_12=d_6*d_6;d_14=d_6*d_8;
        f12=(sigma_6/d_8-2.0*sigma_12/d_14)*24.0*EPSILON;
        f(n1,:)=f(n1,:)+f12*r12;
        f(n2,:)=f(n2,:)-f12*r12; % Newton's 3rd law used here
        U=U+4.0*EPSILON*(sigma_12/d_12-sigma_6/d_6); % accumulate energy
    end
end
end

```

其中，第一层循环对所有原子进行遍历（去掉一个编号最大的；因为会利用牛顿第三定律），第二层循环对某个中心原子的所有“邻居”原子进行遍历（忽略编号比中心原子小的“邻居”；因为会利用牛顿第三定律）。如果不利用牛顿第三定律的话，那么上述代码中的第二层循环就很容易“矢量化”（vectorized）。具体如何矢量化，我感到难以用语言表达，还是直接上代码吧。下面就是将第二层循环矢量化之后的新的求力函数的代码：

```

for n1=1:N % loop over the atoms
    m=1:NN(n1);
    n2=NL(n1,m);% all the neighbors of n1
    r12=zeros(NN(n1),D);
    for d=1:D
        r12(:,d)=r(n2,d)-r(n1,d);
        r12(:,d)=r12(:,d)-round(r12(:,d)/L(d)).*L_times_pbc(d); %minimum image convention
    end
end
end

```

```

end
d12_square=sum(r12.*r12,2);
d_6=d12_square.^3;d_8=d12_square.*d_6;d_12=d_6.*d_6;d_14=d_6.*d_8;
f12=(sigma_6./d_8-2.0*sigma_12./d_14)*(24.0*EPSILON);
f(n1,:)=f(n1,:)+sum([f12,f12,f12].*r12);
U=U+2.0*EPSILON*sum(sigma_12./d_12-sigma_6./d_6); % avoid double counting
end

```

将这部分代码替换原来的部分就得到了一个新的求力的函数（当然，为了比较这两个函数，最好将这两个函数起不同的名字，并保存于不同的文件中）。将上一篇博文中的主函数复制一份（假设取个新名字md_new），并将其中的求力函数名替换为新的函数名，运行与上文同样的例子，即：

```
tic;[E]=md_new(3,4,[4,4,4],5.45*[1,1,1],[1,1,1],1000,1000,20,10,5,80);toc
```

该计算在我的笔记本电脑中耗时为15.099908秒，计算速度是之前的版本（63.758959秒）的4倍！

新的求力函数没有利用牛顿第三定律。在普通的编程方式中，利用牛顿第三定律会让计算量减少一半，从而将性能提升为两倍。然而，在矢量化编程方式中，不宜利用牛顿第三定律，因为利用该定律必将破坏建立好的矢量化（即不得不使用一些for循环来处理与牛顿第三定律相关的计算）。我尝试过在矢量化方式中运用牛顿第三定律，得到的性能比不用该定律还要低一倍。

最后，有兴趣的读者可以验证新的函数与老的函数在计算结果上是否等价。

四、小结

合理利用矢量化编程可以有效地改善matlab程序的性能。将含有256个氩原子的体系演化2000步，矢量化的代码耗时约15秒，非矢量化的代码耗时约一分钟，性能相差4倍。也许该代码还有不少优化的空间，但无论如何优化也赶不上高级语言（例如C或者Fortran）的性能。

我写过C语言的分子动力学模拟程序，作上述同样的计算只要半秒钟，计算速度是上述矢量化matlab程序计算速度的30倍。所以，真正做科学研究的时候几乎不会用matlab写的分子动力学模拟程序。但是，在开发一个新的算法或者实现一个新的功能的时候，先用matlab试试是很不错的。用C或者Fortran需要一个星期才能实现的功能，用matlab也许一天就搞定了。以后有空的话，我会继续以上述优化过的matlab代码为基础，介绍一些分子动力学模拟的技术与应用。

转载本文请联系原作者获取授权，同时请注明本文来自樊哲勇科学网博客。

链接地址：<http://blog.sciencenet.cn/blog-3102863-971052.html>

上一篇：[A 100-line matlab code for molecular dynamics \(slow version\)](#)

下一篇：[Calculating phonon density of states with molecular dynamics](#)

★收藏

当前推荐数：**0**

推荐到博客首页

评论 (2 个评论)

该博文允许注册用户评论 请点击登录

数据加载中...

返回顶部

