基于NUMPY的剂量计算程序改进

刘新建 2023年1月14日

**1.目标需求：**

在已经初步完成的剂量计算程序中，主要采用cupy的GPU加速方式实现，单个源项整年的计算仅需要2-3s，因此并未考虑并行。但在服务器和一些便携笔记本电脑上，由于没有GPU卡，仅采用numpy数组，计算速度大大降低，单个源项的时间可能达到2min，严重影响了整体的计算效率。因此拟对相关程序进行适当的改进优化。

**2.问题分析：**

根据分析，不太适宜采用单次计算作为一个最小并行单元。因为涉及到弥散因子的大量重叠，对于3小时间隔的120h的计算时段，大体重复40次，导致典型弥散因子本身的内存消耗达到20G（每次的弥散因子输入，需要形成独立的变量的列表）。服务器上尚可，一般的电脑可能会带来问题。

**3.解决方法：**

3.1基于计算的特征，拟根据计算平台的核心数（num\_cpu），将整年的弥散因子分为对应段数。典型的笔记本平台有4核心，服务器上可用的大约48-96核心，这样弥散因子重叠量大大降低，典型内存消耗量不会超过弥散因子大小的2倍，也就是一般1G内存即可。这样既能充分利用计算资源，又不大量增加内存消耗。

3.2 考虑到服务器上硬盘读写速度较慢，在计算每个抽样的剂量时，直接将统计结果一并计算。剂量阈值可选用常用值（有效剂量可选100mSv，50mSv，10mSv，1mSv等，并根据用户需要灵活添加，参照nuke map的功能）；并将剂量随距离变化的结果也给出。

3.3 程序要形成两套，一套用于连续释放的场景（一般核电厂事故源项，连续的长时间释放，可以按照单位释放量考虑），一套用于瞬时释放的场景（必须准确的描述释放源项，可得到每个时刻烟团的准确位置）。

3.4 结果可视化

为了描述一次计算过程中的每一个时刻的浓度场和剂量场，需要保存每一个时刻的结果。为了提高效率，节约磁盘空间，可考虑采用函数实时计算的方式。

用户输入开始时刻，程序开始读取该时刻开始的弥散因子，根据源项，计算剂量。结果保存在内存中，可直接在前端展示。根据经验，无论是对于瞬时释放（一次弥散因子的算例单独形成一个文件）或者连续释放场景（逐时的连续结果），读取一段的弥散因子速度应该很快（0.2s），计算大致在0.2s，结果展示画图大约0.5s。1s就可以大致看到结果。用户体验应该可以接受。

**4．程序修改：**

为了保持程序的模块结构，可将一次抽样剂量计算的函数单独包装。将一段时间的抽样计算作为另外一个函数。源项读取和简单的处理作为单独函数？

本次改进中，增加一个将弥散因子按cpu数分配函数即可。（没有必要，将所有弥散因子读入内存，子函数利用i调用对应的数组即可，不用传递xq数组本身）

其他尝试：将met网格设置为120\*120，在服务器上计算时间大约为30min，而瞬时烟团模式，计算时间则为2小时，解压后的con文件，三种核素情况下，大概为200G，能将磁盘全部装满。与网格数量，考虑三种沉积速率有密切关系。后期考虑到效率问题，干湿沉积应尽量简化处理。**不考虑干湿沉积能使影响距离更远，在一般的应急计划区计算中是保守的作法。**后期可根据实际需要开展敏感性对比实验。

为了提升效率，在开始试算过程中，网格数量建议取50-60，可以极大提升效率。整体参数均确定后，可以考虑用80-100的网格，calpuff网格加密处理，计算效率略低，但相关曲线更加平滑，图形更美观。