План работы в рамках курса по обучению основам параллельного программирования.

Дергачёв А. А., Ефимов О. В., Сметанина Е. О. ☺
Московский государственный университет имени М. В. Ломоносова,
Физический факультет, кафедра Общей физики и волновых процессов;
Международный учебно-научный лазерный центр;
НОЦ «Суперкомпьютерные технологии»

Версия 0.5beta1 от 11 января 2015 г.

1. Общая информация.

Работа предусматривает реализацию различных алгоритмов для решения уравнения дифракции лазерного пучка в среде с кубической по полю нелинейностью и сравнение их эффективности. Основная цель — нахождение скорости работы каждого алгоритма в зависимости от таких параметров, как размер матрицы поперечного сечения, количество процессов, точность метода. Все реализации должны включать Makefile для компиляции и файлы run_skif.sh и run_bluegene.sh для запуска расчета и сохранять результаты в определённые папки. Подробности описаны в части 5.

2. Явная схема.

$$2i\frac{\partial E}{\partial z} = \triangle_\perp E + R\left|E\right|^2 E$$

$$\longrightarrow \quad E^{(n+1)}_{i,j} = E^{(n)}_{i,j} + \frac{\Delta z}{2i}\left(\frac{E^{(n)}_{i,j+1} + E^{(n)}_{i,j-1} + E^{(n)}_{i+1,j} + E^{(n)}_{i-1,j} - 4E^{(n)}_{i,j}}{\Delta x^2} + R\left|E^{(n)}_{ij,}\right|^2 E^{(n)}_{i,j}\right)$$

- Не используя метод расщепления по физическим факторам, реализовать приведенную выше явную схему.
- Для численного интегрирования уравнения использовать метод Рунге-Кутта четвертого порядка точности.
- Использовать блочное распределение матрицы между процессами.

3. Метод Фурье.

3.1. Освоение *FFTW*.

- Разобраться с подключением *FFTW* в проект Visual Studio. Мануал по *FFTW* 2.1.3 есть на сайте fftw.org.
- Разобраться с распределением матрицы между процессами, необходимым для *FFTW*.
- Разобраться с расположением результата одного фурье-преобразования и нормировкой.
- Реализовать fftw_mpi(...) туда-обратно при гауссовых начальных условиях в двумерном случае.

3.2. Дифракция в линейной среде.

$$\begin{cases} 2i\dfrac{\partial E}{\partial z} &= \dfrac{\partial^2 E}{\partial x^2} + \dfrac{\partial^2 E}{\partial y^2} \\ E(x,y,0) &= \exp\left\{-\dfrac{x^2+y^2}{2}\right\}, \quad (x,y) \in [-l,l]^2 \end{cases} \longrightarrow$$

$$\longrightarrow \quad \left[E(x,y,z) = \sum_{j,k} \tilde{E}_{jk}(z) \exp\left\{\frac{2\pi i j x}{N}\right\} \exp\left\{\frac{2\pi i k y}{N}\right\} \right] \quad \longrightarrow$$

$$\longrightarrow \quad \begin{cases} 2i\dfrac{\partial \tilde{E}}{\partial z} &= \left(\dfrac{2\pi i}{N}\right)^2 (j^2+k^2)\tilde{E}(z) \\ \tilde{E}_{jk}(0) &= \tilde{E}^{(0)} \end{cases} \quad \longrightarrow$$

$$\longrightarrow \quad \tilde{E}_{jk}(z) = \tilde{E}^{(0)} \exp\left\{ i\frac{2\pi^2}{N^2}(j^2+k^2)z \right\}$$

Итого: $E(x,y,0) \xrightarrow{\text{2D FFT}} \tilde{E}_{jk}(0) \longrightarrow$

$$\longrightarrow \quad \tilde{E}_{jk}(0)\exp\left\{ i\frac{2\pi^2}{N^2}(j^2+k^2)z \right\} \xrightarrow{\text{2D FFT}^{-1}} E(x,y,z)$$

- Реализовать параллельный расчет дифракции гауссового пучка.
- Сравнить с аналитическим решением.
- Исследовать зависимость времени счёта от следующих параметров:
  - с буфером и без него;

  – FFTW_NORMAL_ORDER vs FFTW_TRANSPOSED_ORDER;
  – (опционально) FFTW_ESTIMATE vs FFTW_MEASURE;
  – (опционально) wisdom, OpenMP, ...

- Выбрать оптимальные параметры для использования в следующем пункте.

3.3. Дифракция в нелинейной среде.

$$\left\{\begin{array}{rcl} 2i\dfrac{\partial E}{\partial z} & = & R\left|E\right|^2 E \\ E(x,y,0) & = & E_0(x,y) \end{array}\right. \to E(x,y,z) = E_0(x,y)\exp\left\{-i\frac{R\left|E_0(x,y)\right|^2}{2}z\right\}$$

- Выполнять шаг дифракции и шаг нелинейности в разном порядке:
  1. «дифракция — нелинейность»
  2. «нелинейность — дифракция»
  3. чередовать
- Изменять шаг интегрирования для выполнения условия $\Delta\varphi_{\text{нл}} = \dfrac{R\left|E_{max}\right|^2}{2}\Delta z < 0.1$.
- Использовать оптимальные параметры ***FFTW***, найденные в предыдущем пункте.
- Получить формулу Марбургера для гауссова пучка и ее аналог для ch^{-1} пучка.

4. Неявные схемы.

- Использовать метод расщепления по физическим факторам — на нелинейность и дифракцию с дальнейшим расщеплением дифракции на «дифракцию по x» и «дифракцию по y».
- Для расчета дифракции использовать консервативную схему.
- Расчет проводить, использую метод прогонки для трехдиагональной матрицы.
- Использовать блочное распределение матрицы между процессами.

5. Входные параметры и формат выходных файлов.

5.1. Начальные условия.

Начальное распределение должно иметь вид гауссова пучка. Размер счётной области – 10 радиусов пучкая. Сгенерировать такое распределение можно с помощью create_2d_func (стабильная версия 1.0) со следующими аргументами:

```
$> mpirun -np 8 ./create_2d_func -n 1024 -f gauss \
      -l 5 --a0 1 --r0 1 ./gauss_n1024_l5.cpl
```

5.2. Проводимые расчёты.

1. Время выполнения одного шага дифракции с использованием различных флагов ***FFTW***. Параметры:

   - $N = 1024, 8192$.
   - $np = 1, 8, 32$ для СКИФ «Чебышёв»
     и $np = 128, 1024$ для IBM Bluegene/P.

   Сравнить скорость работы ***FFTW*** при использовании дополнительного буфера и без него, степень зависимости времени выполнения от использования флагов FFTW_NORMAL_ORDER и FFTW_TRANSPOSED_ORDER.

2. Время расчёта распространения пучка в нелинейной среде на одну дифракционную длину(до $z = 1$). Параметры:

   - $R = 5,\ \ \Delta\varphi < 0.1$.
   - $N = 512, 1024, 2048, 4096, 8192, 16384, 32768$.
   - $np = 1, 2, 4, 8, 16, 32, 64, 128$ для СКИФ «Чебышёв»
     и $np = 128, 256, 512, 1024$ для IBM Bluegene/P.

   Провести замер времени выполнения расчётной части программы в друх вариантах: только распространение и распространение с сохранением результирующего поля и максимальной интенсивности пучка каждые 10 шагов. После расчёта с сохранением бинарные данные можно удалить или использовать для оценки точности метода, чтобы сократить число запусков программы. Столбец полной можности пучка в выходной таблице можно заполнить нулями.

3. Точность алгоритмов для случая линейного распространения на одну дифракционную длину(до $z = 1$). Параметры:

   - $R = 0$.

- $N = 512, 2048, 8192$.
- $\Delta z = 0.001$, соответственно 1000 шагов.

Результатом работы программы должна быть таблица с данными об изменении максимальной интенсивности и полной мощности пучка при распространении. Формат таблицы будет приведён ниже. Также необходимо сохранить в файл конечное распределение поля при $z = 1$.

4. Точность алгоритмов для случая нелинейного распространения на одну дифракционную длину(до $z = 1$). Параметры:

   - $R = 5,\ \Delta\varphi < 0.1$.
   - $N = 512, 2048, 8192$.

   Результатом работы программы должна быть таблица с данными об изменении максимальной интенсивности и полной мощности пучка при распространении. Формат таблицы будет приведён ниже. Также необходимо сохранить в файл конечное распределение поля при $z = 1$.

5.3. Правила именования файлов и папок с результатами.

Результаты должны располагаться в папках следующего вида(относительно папки запускаемой программы):

- Для сравнения скорости работы ***FFTW*** с разными флагами:
  `./results/Skif/FFTW_compare/(no|with)buffer_(no|with)transpose/N1024_NP16=4x4/`
- Для замеров времени без сохранения:
  `./results/Skif/Time_no_save/N1024_NP16=4x4/`
- Для замеров времени с сохранением:
  `./results/Skif/Time_with_save/N1024_NP16=4x4/`
- Для оценки точности:
  `./results/Skif/Accuracy_r0/N1024_NP16=4x4/` и
  `./results/Skif/Accuracy_r5/N1024_NP16=4x4/`

В каждой папке должен располагаться файл log.txt с входными данными и результатами работы программы. Бинарные файлы с распределением поля на отдельных шагах должны располагаться в файлах вида out_00070.cpl. Распределение поля в конце трассы (при $z = 1$) должно быть сохранено под именем out_z1.cpl.

5.4. Формат выходного лога программы.

В начале вывода программы должны присутствовать значения параметров сетки и расчётных параметров. Далее должна следовать таблица, содержащая колонки со значениями текущего шага, координаты, максимальной интенсивности и полной мощности пучка. В последней строке нужно вывести время выполнения расчётной части программы.
Образец:

```
    =========================================
    === Propagation: Diffraction and Kerr ===
    =========================================


MPI grid size: 1 (1x1)

N: 256
L: 5.000000
Impulse file: ../../data/gauss_n256_l5.cpl

R: 0.000000
dz: 0.001000 (dphi < 0.1)
Steps: 1000

n        dz          z           I_max(z)        P(z)
00000    0.010000    0.000000    1.000000000000  0.999999999997
00001    0.010000    0.010000    0.999900162406  0.999999999997
00002    0.010000    0.020000    0.999600769015  0.999999999997

...

Execution time(sec):
144.044038
```

5.5. Обработка результатов.

- Для оценки точности алгоритмов необходимо произвести сравнение формы импульсов после распространения на одинаковую длину. В случае линейного распространения результат работы каждого алгоритма можно сравнить с аналитическим решением, в случае нелинейного — только между собой. Для сравнения используется программа bindiff (стабильная версия 0.1).

- Построение графиков осуществить в автоматическом режиме с использованием bash скриптов и gnuplot.

- Визуализировать бинарные данные можно с использованием программы bin2gif (стабильная версия 0.3).

5.6. Отчётные данные.

- Времена работы алгоритма ***FFTW*** при различных параметрах. Выбранные оптимальные параметры.
- Времена работы всех перечисленных алгоритмов без сохранения и с учётом сохранения. Ускорение работы программ в зависимости от количества процессов.
- Интегральная среднеквадратичная ошибка в распределении поля при дифракции гауссова пучка (линейный случай).
- Интегральная среднеквадратичная ошибка в соответствии с формулой Марбургера для расстояния филаментации (нелинейный случай).

6. Исполнители.

Дергачёв А. А. <dergachev88@yandex.ru>: Метод прогонки.
Ефимов О. В. <efimovov@yandex.ru>: Метод Рунге-Кутта
Сметанина Е. О. <jannes-2002@yandex.ru>: Метод БПФ.