ПОЛИТЕХ
Санкт-Петербургский Политехнический Университет Петра Великого

Курс "Вычисления на многопроцессорных компьютерах" Лабораторный практикум

<u>Курсовой проект:</u>
Разработка параллельных программ численного решения двумерного уравнения Пуассона на основе технологий OpenMP и MPI

Цель работы - практическое применение освоенных в рамках учебного курса технологий параллельного программирования OpenMP и MPI для разработки программы, распараллеливающей традиционные численные алгоритмы решения двумерного уравнения Пуассона.

Задание к работе - разработать программы, параллелизующие с помощью технологий OpenMP и MPI методы Якоби (простой итерации) и Гаусса-Зейделя численного решения двумерного уравнения Пуассона.

Методические указания

Рассмотрим численное решение дифференциального уравнения в частных производных на примере двумерного уравнения Пуассона:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

Применим постановку задачи Дирихле: вычисления будем выполнять для некоторой заданной формы расчетной области с заданными же граничными условиями. Используем ортогональную равномерную расчетную сетку размером N_xxN_y элементов. Пространственную дискретизацию уравнения Пуассона проведем по методу конечных разностей со вторым порядком точности. К реализации предлагаются два численных алгоритма: метод Якоби (простой итерации) и Гаусса-Зейделя.

Организация итерационного процесса по методу Якоби:

$$u_{i,j}^{(k+1)} = \frac{1}{4}\left(u_{i-1,k}^{(k)} + u_{i+1,k}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)} + \Delta^2 f_{i,j}\right)$$

Организация итерационного процесса по методу Гаусса-Зейделя:

$$u_{i,j}^{(k+1)} = \frac{1}{4}\left(u_{i-1,k}^{(k+1)} + u_{i+1,k}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} + \Delta^2 f_{i,j}\right)$$

Как видно, в методе Якоби все новые значения искомого решения U вычисляются с использованием значений с предыдущей итерации. Старые значения берутся из одного массива, а записываются в другой, что требует выделения двух независимых буферов памяти. После обновления массива в соответствии с приведенной выше формулой его содержимое перед следующей итерацией копируется во второй массив.

В методе Гаусса-Зейделя каждое новое значение массива вычисляется на основании значений соседних элементов, как в методе Якоби, но затем сразу же записываются обратно в исходный массив. Это означает, что если элементы обновляются в порядке их расположения в памяти, следующий вычисляемый элемент массива будет зависеть от уже обновленных значений U в узлах (i-1, j) и (i, j-1), в то время как два других соседних элемента в этот момент еще не обновлены. Как известно, метод Гаусса-Зейделя сходится в два раза быстрее в сравнении с количеством итераций метода Якоби и не требует выделения дополнительной памяти.

Предполагается, что решение будет найдено, когда выполнится условие $||U - U_{old}||_F < err_{max}$, то есть норма Фробениуса (разница между текущим полем и полем с предыдущей итерации) меньше максимальной ошибки $err_{max} = 10^{-7}$.

Параллелизация с помощью технологии OpenMP

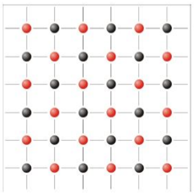
Параллелизацию метода Якоби предлагается провести при задействовании использовании двух различных массивов для хранения результатов вычислений на текущей и предыдущей итерациях:

```
do n=1,maxiter
!$omp do
   do j=2, nx - 1
      do i=2, ny - 1
         U(i, j) = 0.25 * (T(i-1, j) + T(i+1, j) + T(i, j-1)&
            + T(i, j+1) + F(i, j))
      enddo
   enddo
!$omp end do
enddo
```

Параллелизацию метода Гаусса-Зейделя предлагается выполнить с использованием красно-черного упорядочивания (Red-black Gauss-Seidel):

```
!$omp parallel shared(T, F, nx, ny, maxiter, interval, maxerr) &
!$omp   & private(n, i, j, istart, istop)
  do n=1, 2 * maxiter
!$omp do
     do j=2, nx - 1
        istart = 2 + mod(j + n, 2)
        istop = ny - 1
        do i=istart, istop, 2
           T(i, j) = 0.25 * (T(i-1, j) + T(i+1, j) + T(i, j-1) &
              + T(i, j+1) + F(i, j))
        enddo
     enddo
!$omp end do
  enddo
!$omp end parallel
```

Данный метод предполагает разделение элементов массива на 2 типа – "красные" и "черные", располагаемые в шахматном порядке. Такой выбор способа разделения обусловлен видом используемого шаблона, в соответствии с которым для расчета в "красных" узлах нужны значения в "черных" узлах и наоборот. По своей сути, красно-черное упорядочивание обеспечивает перевод рекуррентных формул метода Гаусса-Зейделя в двухшаговое использование формул Якоби. Таким образом, обновление элементов массива выполняется в два этапа: на первом этапе каждое обновление следует той же формуле, что и в методе Якоби, а на втором этапе для обновления значений второй половины элементов массива используются значения, рассчитанные на первом этапе. Первый этап использует только "старые" значения, а второй этап – только "новые" значения, так что в результате элемент обновляется на основании двух новых и двух старых значений.



[Программа численного решения двумерного уравнения Пуассона по методу Якоби (OpenMP-параллелизация)»]

Параллелизация с помощью технологии MPI

Предлагается выполнить средствами MPI параллелизацию метода Якоби с использованием одномерной (1D) или двумерной (2D) декомпозиции расчетной области с использованием блокирующих и неблокирующих обменов между процессами.

Варианты индивидуальных заданий:

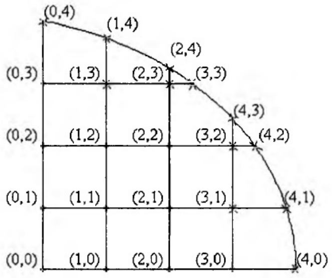
1. 1D декомпозиция, использование для обмена данными между соседними процессами процедур MPI_BSEND (передача данных через выделенный буфер) и MPI_RECV
2. 1D декомпозиция, использование для обмена данными между соседними процессами процедур MPI_SEND и MPI_RECV (парные блокирующие обмены)
3. 1D декомпозиция, использование для обмена данными между соседними процессами процедуры MPI_SENDRECV (совместные прием-передача)
4. 1D декомпозиция, использование для обмена данными между соседними процессами процедур MPI_ISEND и MPI_IRECV (неблокирующие обмены), а таже процедуры MPI_WAITALL для блокирующего ожидания завершения всех обменов

5. 1D декомпозиция, использование для обмена данными между соседними процессами процедур MPI_SEND и MPI_RECV (блокирующие обмены приграничными данными)
6. 2D декомпозиция, использование для обмена данными между соседними процессами процедур MPI_SSEND и MPI_RECV (обмен данными с синхронизацией)
7. 2D декомпозиция, использование для обмена данными между соседними процессами процедуры MPI_SENDRECV (совместные прием-передача)

Варианты программы численного решения двумерного уравнения Пуассона по методу Якоби (параллелизация с помощью MPI, язык Fortran)

Описание алгоритма численного решения уравнения Пуассона для криволинейной области: Меркулова Н.Н. Методы приближенных вычислений (с. 21) »

Пример расчетной сетки для криволинейной области (с разделением узлов на внутренние и граничные):



Варианты индивидуальных заданий

Вариант № 1

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,2 \cdot \ln|x+y|,$$

$$u|_{\Gamma} = 0,5(x+y)^2,$$

где $\Gamma = \left\{(x,y):\ y = -0,5x^2 + 3,\ \ 0 \le x \le \sqrt{6},\ \ 0 \le y \le 3\right\}$.

Вариант № 2

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,2,$$

$$u|_{\Gamma} = e^{-1,5x^2+2y^2},$$

где $\Gamma = \left\{(x,y):\ y = -0,15x^2 + 0,6,\ \ 0 \le x \le 2,\ \ -0,6 \le y \le 0,6\right\}$.

Вариант № 3

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,3,$$

$$u|_{\Gamma} = \sqrt{x^2 y^2 + 1},$$

где $\Gamma = \left\{(x,y):\ y = x^2 - 1,\ \ 0 \le x \le 1,\ \ -1 \le y \le 1\right\}$.

Вариант № 4

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,25,$$

$$u|_{\Gamma} = \sqrt{0,25x^2 + y^2},$$

где $\Gamma = \left\{(x,y):\ y = 0,25x^2,\ y \le 2\right\}$.

Вариант № 5

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = e^{-x},$$

$$u|_{\Gamma} = e^{-0,5(x+y)^2},$$

где $\Gamma = \left\{(x,y):\ y = -0,5x^2 + 3,\ \ 0 \le x \le \sqrt{6},\ \ 0 \le y \le 3\right\}$.

Вариант № 6

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,5,$$

$$u|_\Gamma = e^{-xy},$$

где $\Gamma=\left\{(x,y):\ \frac{x^2}{9}+\frac{y^2}{4}=1,\ \ 0\le x\le 3,\ \ 0\le y\le 2\right\}$.

Вариант № 7

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,1,$$

$$u|_\Gamma = -0,5x^2+y^2,$$

где $\Gamma=\left\{(x,y):\ y=-0,5x^2+1,\ \ 0\le x\le \sqrt{2},\ \ -1\le y\le 1\right\}$.

Вариант № 8

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,6,$$

$$u|_\Gamma = \sin(xy),$$

где $\Gamma=\left\{(x,y):\ x^2+y^2=1,\ \ 0\le x\le 1,\ \ 0\le y\le 1\right\}$.

Вариант № 9

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=e^{-(x+y)},$$

$$u|_\Gamma = e^{-(x+y)^2},$$

где $\Gamma=\left\{(x,y):\ y=-0,5x^2+2,\ \ 0\le x\le 2,\ \ 0\le y\le 2\right\}$.

Вариант № 10

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,4,$$

$$u|_\Gamma = e^{-0,4xy},$$

где $\Gamma=\left\{(x,y):\ y=x^2-4,\ \ 0\le x\le 2,\ \ -4\le y\le 0\right\}$.

Вариант № 11

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,3,$$

$$u|_\Gamma = \sin(0,1xy),$$

где $\Gamma=\left\{(x,y):\ x^2+(y^2-9)=0,\ \ 1\le x\le 3,\ \ 0\le y\le 3\right\}$.

Вариант № 12

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,6,$$

$$u|_\Gamma = e^{-x+y},$$

где $\Gamma=\left\{(x,y):\ y=x^2-4,\ \ 0\le x\le 2,\ \ -4\le y\le -1\right\}$.

Вариант № 13

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\cos(0,1xy),$$

$$u|_\Gamma = e^{2x^2-y^2},$$

где $\Gamma=\left\{(x,y):\ y=0,5x^2-1,\ \ 0\le x\le \sqrt{2},\ \ -1\le y\le 1\right\}$.

Вариант № 14

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,7,$$

$$u|_\Gamma = |\sin xy|,$$

где $\Gamma=\left\{(x,y):\ x^2+y^2=2,\ \ 0\le x\le \sqrt{2},\ \ -1\le y\le \sqrt{2}\right\}$.

Вариант № 15

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,5,$$

$$u|_\Gamma = e^{xy},$$

где $\Gamma=\left\{(x,y):\ y=-x^2+0,2,\ \ 0\le x\le \sqrt{0,2},\ \ -1\le y\le 0,2\right\}$.

Вариант № 16

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0{,}1,$$

$$u|_{\Gamma}=e^{-0{,}3(x+y)},$$

где $\Gamma=\left\{(x,y):\ y=0{,}5x^2-2,\ 0\le x\le 2,\ -2\le y\le 0\right\}$.

Вариант № 17

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0{,}4,$$

$$u|_{\Gamma}=\cos(xy),$$

где $\Gamma=\left\{(x,y):\ x^2+y^2=1,\ 0\le x\le 1,\ -1\le y\le 1\right\}$.

Вариант № 18

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$$

$$u|_{\Gamma}=2x^2-4,$$

где $\Gamma=\left\{(x,y):\ x^2+y^2=4\right\}$.

Вариант № 19

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$$

$$u|_{\Gamma}=8-\frac{26}{9}x^2,$$

где $\Gamma=\left\{(x,y):\ \frac{x^2}{9}+\frac{y^2}{4}=1\right\}$.

Вариант № 20

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$$

$$u|_{\Gamma}=2x^2+2x-2y-4,$$

где $\Gamma=\left\{(x,y):\ x^2+y^2=4\right\}$.

Ход выполнения работы над проектом

1. Выбрать индивидуальный вид уравнения Пуассона, расчетной области и граничных условий.
2. Разработать и отладить последовательные версии программ, выполняющих численное решение уравнения Пуассона методами Якоби и Гаусса-Зейделя. Провести верификацию работы программ и исследовать сходимость алгоритмов, задавая различные размерности расчетных сеток (зависимость от числа узлов сетки количества итераций, времени выполнения программы, а также количества итераций, выполняемых в секунду, для обоих методов численного решения уравнения).
3. Визуализировать результаты численного решения по двум методам в виде 3D-поверхности.
4. Выполнить параллелизацию программ для обоих методов с использованием технологии OpenMP.
5. Выполнить параллелизацию программы для метода Якоби с использованием технологии MPI (выбрав индивидуальный вариант задания).
6. Проверить правильность выполнения параллелизации, сопоставив результаты с полученными для последовательной версии программы.
7. Провести исследования сходимости распараллеленных с помощью OpenMP методов для набора расчетных сеток разной размерности и для постоянного числа нитей, например, четырех (зависимость от числа узлов сетки количества итераций, времени выполнения программы, а также количества итераций, выполняемых в секунду, для обоих методов численного решения уравнения).
8. Провести исследования сходимости распараллеленных с помощью MPI метода Якоби для набора расчетных сеток разной размерности и для постоянного числа процессов, например, четырех (зависимость от числа узлов сетки количества итераций, времени выполнения программы, а также количества итераций, выполняемых в секунду).
9. Провести для различных размерностей расчетных сеток исследование зависимости времени работы программы, ускорения и эффективности распараллеливания, задавая при запуске разное число нитей (для OpenMP) и процессов (для MPI).
10. Сопоставить и проанализировать результаты для двух технологий параллельного программирования.

Структура отчета по курсовому проекту

По результатам выполнения курсового проекта необходимо подготовить развернутый отчет, который должен включать в себя следующие разделы:

- титульный лист, оформленный в соответствии с установленными нормами
- формулировка целей и задач проекта, а также индивидуального задания по проекту

- формулировка базовых численных алгоритмов для методов Якоби и Гаусса-Зейделя
- результаты верификации разработанных последовательных программ
- визуализированные результаты численного решения в виде 3D-поверхности
- результаты исследования сходимости последовательных алгоритмов для набора расчетных сеток разной размерности (в табличном и графическом виде как зависимость от числа узлов сетки количества итераций, времени выполнения программы, а также количества итераций, выполняемых в секунду, для обоих методов численного решения уравнения)
- описание реализованных параллельных алгоритмов с детальными комментариями к использованным методикам распараллеливания
- исходные коды разработанных программ с содержательными комментариями к использованным OpenMP и MPI конструкциям
- результаты верификации программ при параллельном выполнении
- результаты исследования сходимости параллельных алгоритмов для набора расчетных сеток разной размерности и заданного постоянного числа нитей/процессов (в табличном и графическом виде как зависимость от числа узлов сетки количества итераций, времени выполнения программы, а также количества итераций, выполняемых в секунду)
- результаты рассчитанного времени работы программы, ускорения и эффективности распараллеливания при задании разного числа нитей/процессов (в табличном и графическом виде, для набора расчетных сеток разной размерности)
- сопоставление и совместный анализ результаты для двух технологий параллельного программирования
- общие выводы по выполненному курсовому проекту

Учебные материалы:

Лекции по технологии OpenMP
Лекции по технологии MPI
Parallel Poisson Solver in Fortran (описание методик распараллеливания алгоритма численного решения уравнения Пуассона с помощью OpenMP)

Книги и учебники:

Абрамов А.Г. Вычисления на многопроцессорных компьютерах на основе технологии OpenMP
Антонов А.С. Параллельное программирование с использованием технологии OpenMP
Вержбицкий В.М. Основы численных методов
Меркулова Н.Н. Методы приближенных вычислений
Гергель В.П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем
Немнюгин С.А. Средства программирования для многопроцессорных вычислительных систем
Официальные спецификации OpenMP

Антонов А.С. Параллельное программирование с использованием технологии MPI
Баканов В.М., Осипов Д.В. Введение в практику разработки параллельных программ в стандарте MPI
Шпаковский Г.И., Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI
MPI Tutorial (примеры программ на Fortran)
MPI: The Complete Reference (руководство по MPI с примерами)

Бартеньев О.В. Современный Фортран
Немнюгин М.А., Стесик О.Л. Современный Фортран. Самоучитель

Примеры программ:

- Программа численного решения двумерного уравнения Пуассона по методу Якоби
- Вариант последовательной программы численного решения двумерного уравнения Пуассона по методу Якоби
- Вариант программы численного решения двумерного уравнения Пуассона по методу Якоби (параллелизация с помощью OpenMP, язык Fortran)
- Вариант программы численного решения двумерного уравнения Пуассона по методу Якоби (параллелизация с помощью OpenMP, язык C)
- Варианты программы численного решения двумерного уравнения Пуассона по методу Якоби (параллелизация с помощью MPI, язык Fortran)

- Вариант программы численного решения двумерного уравнения Пуассона по методу Якоби (параллелизация с помощью MPI, язык C)