Введение в методы параллельных вычислений

Разработчик:
А.В. Старченко, д.ф.-м.н., профессор
E-mail: starch@math.tsu.ru
Томский государственный университет

Направление 010300.68
«Фундаментальная информатика и информационные технологии»

Проект комиссии Президента по модернизации и техническому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»

Разработка курса выполнена в рамках Проекта комиссии Президента РФ по модернизации и техническому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»

Применение потенциала суперкомпьютерных технологий (СКТ) как значимой составляющей инновационного развития страны является задачей государственной важности, относится к приоритетному направлению и находится под постоянным контролем Президента и Правительства России. Одним из сдерживающих факторов развития страны в этом направлении является острая нехватка высококвалифицированных кадров в области СКТ, поскольку подготовка таких специалистов сейчас отсутствует как элемент системы высшего профессионального образования.

Стратегической целью проекта является создание национальной системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения.

http://hpc-education.ru.

Содержание курса

- Введение
- Рекуррентные формулы
- Параллельные вычисления определенных и кратных интегралов
- Умножение матрицы на вектор. Умножение матриц
- Прямые методы решения систем линейных уравнений на многопроцессорных системах Организация межпроцессорных обменов
- Трехдиагональные системы. Параллельная реализация прямых методов решения систем линейных уравнений
- Параллельная реализация итерационных методов решения СЛАУ
- Параллельная реализация быстрого преобразования Фурье

Содержание лекции

- Вычисление определенного интеграла
  - Формула средних прямоугольников
  - Квадратурная формула трапеций
  - Адаптивный алгоритм для формулы трапеций
  - Параллельная реализация
- Вычисление кратных интегралов
  - Метод ячеек
  - Метод последовательного интегрирования
  - Метод статистических испытания (метод Монте-Карло)
  - Параллельная реализация

Вычисление определенных интегралов

- Пусть требуется найти значение определенного интеграла Римана для некоторой функции $f(x)$

$$I = \int_a^b f(x)dx$$

- Простые квадратурные формулы можно вывести непосредственно и определения интеграла

$$I = \lim_{n\to\infty} \sum_{i=1}^{n} f(\xi_i) * (x_i - x_{i-1}), \xi_i \in [x_{i-1}, x_i]$$

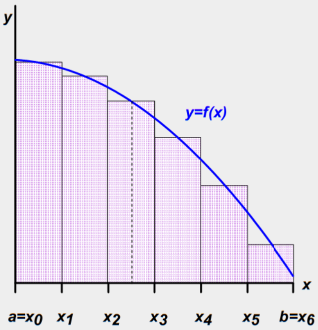
Вычисление определенных интегралов

- Общая формула прямоугольников

$$I \approx \sum_{i=1}^{n} f(\xi_i)(x_i - x_{i-1})$$

- Пусть $x_i = a + h \cdot i, i = 0,...,n; h = (b-a)/n$
- Тогда можно получить формулы:
  - Левых прямоугольников: $\xi_i = x_{i-1}$
  - Правых прямоугольников: $\xi_i = x_i$
  - Средних прямоугольников: $\xi_i = 0.5 \cdot (x_{i-1} + x_i)$

Вычисление определенных интегралов

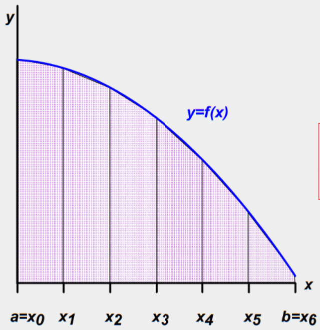


- Наиболее точной из рассмотренных формул прямоугольников является формула средних прямоугольников:

$$I \approx h\sum_{i=1}^{n} f\left(x_{i-1}+h/2\right)$$

- В ней не используются значения функции, вычисленные на концах отрезка интегрирования.

Вычисление определенных интегралов



- Квадратурная формула трапеций получается с использованием формул левых и правых прямоугольников.

$$I \approx h\sum_{i=1}^{n-1} f(x_i)+h\frac{f(a)+f(b)}{2}$$

- И хотя ее точность при заданном ***n*** хуже чем у метода средних прямоугольников, она требует сравнительно меньшего объема вычислений.

Вычисление определенных интегралов

- Точность получаемых результатов в методах численного интегрирования зависит как от выбранной квадратурной формулы и характера изменения подынтегральной функции, так и от шага интегрирования.
- При интегрировании медленно меняющейся функции шаг можно выбирать большим, чем при интегрировании резко меняющихся функций.
- Адаптивные алгоритмы позволяют вводить разные значения шага интегрирования на отдельных участках отрезка интегрирования, что дает возможность уменьшить машинное время без потери точности результатов расчета.

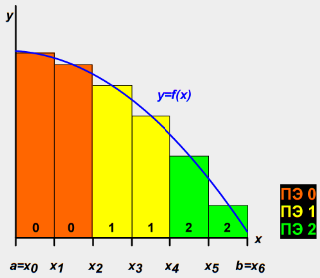
Вычисление определенных интегралов

- Принцип работы адаптивного алгоритма:
  1. Отрезок ***[a,b]*** разбиваем на ***n*** равных частей. ***k=0***
  2. Для каждого отрезка ***[x***$[x_i,x_i+h]$ ($i=0,\dots,n-1$) применяем формулы численного интегрирования для разбиений отрезка с шагами $h/2^k$ и $h/2^{k+1}$.
  3. Полученные значения интеграла сравниваем и проводим оценку погрешности.
  4. При неудовлетворении требования точности расчетов производим дробление таких элементарных отрезков (***k=k+1***) и переход на пункт 2.

Параллельная реализация

- Декомпозиция: Область интегрирования ***[a,b]*** разобьем на подобласти, а исходный интеграл представим в виде суммы интегралов по таким частичным отрезкам. На каждом таком отрезке интеграл вычисляется по выбранной квадратурной формуле.
- Проектирование коммуникаций: Полученные результаты с каждой подобласти должны быть собраны вместе для получения приближенного значения интеграла.
- Укрупнение: Объединение подобластей проводится с учетом планируемого к использованию числа процессоров ***p***, минимизации межпроцессорной передачи данных и равномерной загрузки процессоров.

Параллельная реализация



- Формула средних прямоугольников (блочный способ распределения подобластей по процессорным элементам).
- Такой способ распределения подобластей по процессорным элементам может дать неравномерную загрузку, например для кусочно-гладких функций.

Параллельная реализация

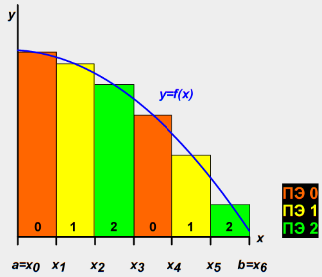
- Формула средних прямоугольников:

$$I \approx h\sum_{i=1}^{n} f\left(0.5(x_{i-1}+x_i)\right) =$$

$$= h\sum_{\mu=0}^{p-1}\left[\sum_{i=1}^{n/p} f\left(0.5(x_{i-1+\mu n/p}+x_{i+\mu n/p})\right)\right]$$

- Суммы в квадратных скобках вычисляются одновременно и независимо.

Параллельная реализация



- Формула средних прямоугольников (циклический способ способ распределения подобластей по процессорным элементам).
- При использовании адаптивного алгоритма также следует уделять внимание равномерному распределению вычислительной нагрузки между ПЭ.

Кратные интегралы

- Метод ячеек
- Метод повторного интегрирования
- Метод статистических испытаний

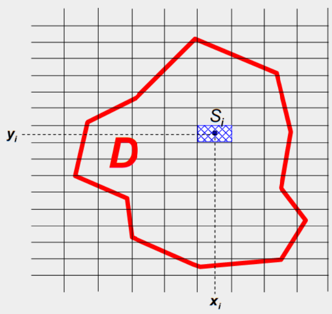
Метод ячеек

- Область ***D*** покрывается сеткой с прямоугольными ячейками.
- Для приближенного вычисления интеграла в каждой ячейке используется формула средних

$$I = \iint_D f(x,y)\,dxdy \approx \sum_{i=1}^{N} f(\overline{x}_i,\overline{y}_i)\cdot S_i$$

Здесь $\overline{x}_i,\overline{y}_i$ - координаты центра масс ячейки;
S_i - площадь ячейки.

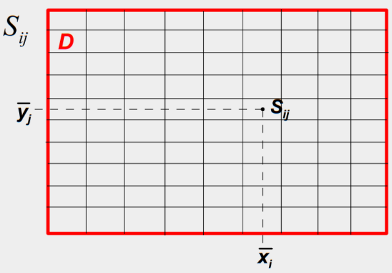
Метод ячеек



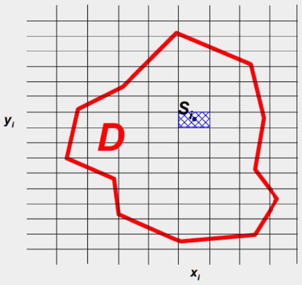
Метод ячеек

- Если область ***D*** имеет прямолинейные границы, то формула приобретает вид:

$$I \approx \sum_{i=1}^{N}\sum_{j=1}^{M} f(\overline{x}_i,\overline{y}_j)\cdot S_{ij}$$



Метод ячеек



- Метод ячеек переносится на большее число измерений.
- Он эффективен для областей с прямолинейными границами.
- Параллельные вычисления кратных интегралов по этому методу организуются с использованием геометрической декомпозиции, причем, возможно, неоднородной.

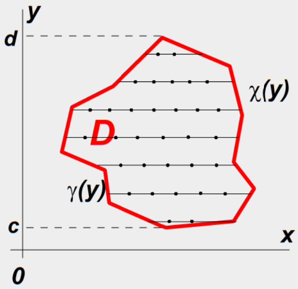
Последовательное интегрирование

- Через область ***D*** проводятся хорды, параллельные оси ***Ox***, и на них вводятся определенным образом узлы.
- Интеграл представляется в виде

$$I=\iint_D f(x,y)dxdy=\int_c^d F(y)dy; F(y)=\int_{\gamma(y)}^{\chi(y)} f(x,y)dx$$

- Сначала вычисляется интеграл по ***x*** вдоль каждой хорды по одномерной квадратурной формуле, затем интеграл по ***y*** – здесь в качестве узлов можно взять проекции хорд на ось ординат.

Последовательное интегрирование



- Сведение кратного интеграла к повторному интегрированию по одномерным квадратурным формулам позволяет использовать все представленные ранее результаты.
- Существуют другие методы, использующие кубатурную формулу

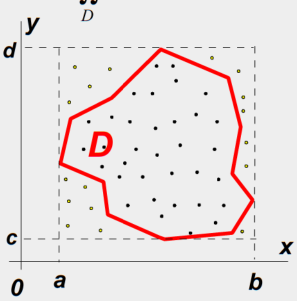
$$I \approx \sum_i C_i f(r_i)$$

Метод статистических испытаний

- Метод Монте-Карло (статистических испытаний) для вычисления многомерных интегралов впервые был введен Э.Ферми, а затем использовался Дж. фон Нейманом и Ст.Уламом в Лос-Аламосе в связи с моделированием нейтронной диффузии в расщепляемом материале.
- Используется для решения задач в различных областях физики, химии, математики, экономики, оптимизации, теории управления и др.

Метод Монте-Карло

$$I=\iint_D f(x,y)dxdy \approx \frac{(b-a)(d-c)}{N}\sum_{i=1}^{N_{in}} f(\xi_i,\eta_i)$$



N – это количество вброшенных точек с координатами ***(ξ***(ξ_i, η_i);

N_{in} – это количество точек, попавших в область ***D***;

ξ_i – случайные числа, равномерно распределенные на отрезке ***[a,b]***;

η_i – случайные числа, равномерно распределенные на отрезке ***[c,d]***

Метод Монте-Карло

- Чем больше ***N***, тем более точным будет результат. Погрешность метода оценивается как ***$O(N^{-1/2})$***. Т.е., для увеличения точности в 10 раз нужно в 100 раз увеличить количество вбрасываемых точек.
- Однако независимость вычислений в методе Монте-Карло позволяет рассматривать его как метод практически с идеальным параллелизмом.
- Располагая независимыми генераторами псевдослучайных чисел, вычисление интеграла можно проводить одновременно и независимо несколькими процессорами.
- Коммуникации потребуются только при получении итогового результата.

Заключение

- Рассмотрены квадратурные формулы для вычисления определенных и кратных интегралов.
- Показано, что для параллельной реализации методов необходимо при распределении вычислений по процессорам обеспечивать равномерную загрузку процессоров и минимальные затраты на межпроцессорную пересылку данных.
- Этим требованиям удовлетворяет метод Монте-Карло, однако имеет небольшую точность.

Вопросы для обсуждения

- В чем заключается преимущество обобщенной формулы средних прямоугольников перед формулой трапеций?
- Докажите, что метод средних прямоугольников имеет больший объем вычислений по сравнению с формулой трапеций.
- Для чего нужны адаптивные алгоритмы при вычислении интегралов? Опишите принципы их работы.
- Как построить параллельный алгоритм расчета по квадратурной формуле? Перечислите основные этапы.
- Чем опасен блочный способ распределения вычислительной нагрузки между процессорами при вычислении интегралов?
- Как лучше распределять подобласти вычисления интегралов между процессорами при использовании адаптивного алгоритма?
- Перечислите основные способы вычисления кратных интегралов. Дайте их сравнительный анализ.
- Какие методы параллельного программирования используются при вычислении кратных интегралов?

Темы заданий для самостоятельной работы

- Написать MPI-программу вычисления определенного интеграла с точностью $\varepsilon>0$, используя обобщенную квадратурную формулу трапеций. Для оценки точности использовать правило Рунге.
- Написать MPI-программу вычисления определенного интеграла с точностью $\varepsilon>0$, используя обобщенную квадратурную формулу Симпсона. Для оценки точности использовать правило Рунге.
- Написать MPI-программу вычисления двойного интеграла методом ячеек.
- Написать MPI-программу вычисления двойного интеграла методом Монте-Карло.

Основная литература

1. *Старченко А.В., Есаулов А.О.* Параллельные вычисления на многопроцессорных вычислительных системах. – Томск: Изд-во Том. ун-та, 2002.
2. *Калиткин Н.Н.* Численные методы. – М.: Наука, 1978.
3. *Каханер Д., Моулер К., Нэш С.* Численные методы и программное обеспечениею – М.: Мир, 2001.
4. *Самарский А.А., Гулин А.В.* Численные методы. – М.: Наука, 1989.
5. *Миллер Р., Боксер Л.* Последовательные и параллельные алгоритмы. – М.: БИНОМ. Лаборатория знаний, 2006.
6. *Самарский А.А.* Введение в численные методы. – М.: Изд-во «Лань», 2005.

Следующая тема

- Введение
- Рекуррентные формулы
- Параллельные вычисления определенных и кратных интегралов
- Умножение матрицы на вектор. Умножение матриц
- Прямые методы решения систем линейных уравнений на многопроцессорных системах Организация межпроцессорных обменов
- Трехдиагональные системы. Параллельная реализация прямых методов решения систем линейных уравнений
- Параллельная реализация итерационных методов решения СЛАУ
- Параллельная реализация быстрого преобразования Фурье