

Билет №17 №1 Метод релаксации. Методы Якоби и Гаусса-Зейделя.

Методы релаксации для решения граничных задач.

Как упоминалось выше, методы релаксации разбивают разреженную матрицу на несколько, а затем задача решается при помощи итераций. Можно объяснить метод релаксации с физической точки зрения. Предположим нам надо решить эллиптическое уравнение:

$$Lu = \rho \quad (4.1)$$

Здесь L – эллиптический оператор, а ρ – правая часть уравнения, тогда перепишем уравнение как уравнение диффузии:

$$(4.2)$$

Начальное распределение u релаксирует к равновесному решению при T стремящемся к

$$\frac{\partial u}{\partial t} = Lu - \rho$$

бесконечности. Уравнение диффузии для нашей задачи можно записать в виде:

$$(4.3)$$

Если мы используем разностную схему FTCS, то получим:

$$(4.4)$$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \rho$$

$$u_{j,l}^{n+1} = u_{j,l}^n + \frac{\Delta t}{\Delta^2} (u_{j+1,l}^n + u_{j-1,l}^n + u_{j,l+1}^n + u_{j,l-1}^n - 4u_{j,l}^n) - \rho_{j,l} \Delta t$$

Эта схема устойчива, если $\Delta t/(\Delta^* \Delta) \leq 1/2$, в двумерном случае $\Delta t/(\Delta^* \Delta) \leq 1/4$, возьмем максимально возможный шаг, при котором $\Delta t/(\Delta^* \Delta) = 1/4$, тогда уравнение (4.4) примет вид:

$$u_{j,l}^{n+1} = \frac{1}{4} (u_{j+1,l}^n + u_{j-1,l}^n + u_{j,l+1}^n + u_{j,l-1}^n) - \frac{\Delta^2}{4} \rho_{j,l} \quad (4.5)$$

Эта классическая разностная схема была предложена в конце прошлого века и называется методом Якоби. Этот метод редко используется на практике из-за медленной сходимости, однако он служит основой для понимания многих современных методов.

Второй классический метод называется методом Гаусса-Зейделя; этот метод используется в многосеточных методах решения граничных задач. В этом методе два значения неизвестной функции в правой части (4.5) берутся в момент времени $n+1$, как только они становятся известны.

$$(4.6)$$

j, l

$$u_{j,l}^{n+1} = \frac{1}{4} (u_{j+1,l}^n + u_{j-1,l}^{n+1} + u_{j,l+1}^n + u_{j,l-1}^{n+1}) - \frac{\Delta^2}{4} \rho_{j,l}$$

Этот метод также медленно сходится, однако анализ этого метода может быть полезен.

Рассмотрим методы Якоби и Гаусса-Зейделя с точки зрения представления матриц в виде суммы. Заменяем обозначение u на x , чтобы получить стандартный вид матричного уравнения.

$$(4.7)$$

$$A \cdot x = b$$

Мы можем представить матрицу A в виде

$$A = L + D + U \quad (4.8)$$

Здесь $-D$ – диагональная часть матрицы A , L – нижняя треугольная часть матрицы A , U – верхняя треугольная часть матрицы A , матрицы L , U содержат нули на диагонали. При использовании метода Якоби итерацию на r -м шаге можно записать в виде:

$$D \cdot x^{(r)} = -(L+U) \cdot x^{(r-1)} + b$$

(4.9)

Матрица $-D^{-1}(L+U)$ – итерационная матрица при помощи которой находится следующее итерационное приближение. Была произведена для этого метода оценка числа итераций, необходимых для достижения точности 10^{-p}

$$r \approx \frac{\rho \ln 10}{(-\ln \rho_s)}$$

(4.10)

При увеличении размерности сетки J спектральный радиус ρ_s стремится к единице. Для данного конкретного уравнения, граничных условий и геометрии сетки спектральный радиус, в принципе, можно вычислить аналитически, так для сетки размерности $J \times J$ с условиями Дирихле на всех четырех границах, асимптотическая формула для больших J имеет вид:

$$\rho_s \approx 1 - \frac{\pi^2}{2J^2}$$

(4.11)

При этом необходимое число итераций можно оценить по формуле:

$$r \approx \frac{2 \rho J^2 \ln 10}{\pi^2} \approx \frac{1}{2} \rho J^2$$

(4.12)

Другими словами, число итераций пропорционально числу точек сетки. Методу Гаусса-Зейделя соответствует следующее матричное уравнение:

$$(L+D) \cdot x^{(r)} = -U \cdot x^{(r-1)} + b$$

(4.13)

Для рассматриваемой нами модели спектральный радиус и число итераций можно оценить по формулам:

$$\rho_s \approx 1 - \frac{\pi^2}{J^2}$$

(4.14)

(4.15)

$$r \approx \frac{\rho J^2 \ln 10}{\pi^2} \approx \frac{1}{4} \rho J^2$$

Метод SOR

Мы получим лучший алгоритм – один из самых распространенных до семидесятых годов прошлого века - если мы скорректируем величину ω на r -м шаге итераций Гаусса-Зейделя. Из метода Гаусса-Зейделя следует:

$$x^{(r)} = x^{(r-1)} - (L+D)^{-1} \cdot [(L+D+U) \cdot x^{(r-1)} - b]$$

(4.16)

Член в квадратных скобках – вектор невязки $\xi^{(r-1)}$ т.е.

$$x^{(r)} = x^{(r-1)} - (L+D)^{-1} \cdot \xi^{(r-1)}$$

(4.17)

Для улучшения сходимости введем так называемый параметр «сверхрелаксации» ω :

$$x^{(r)} = x^{(r-1)} - \omega(L+D)^{-1} \cdot \xi^{(r-1)} \quad (4.18)$$

Метод, использующий эту схему, назвали методом SOR (successive overrelaxation). Можно доказать следующие теоремы:

Метод сходится только для $0 < \omega < 2$, если $0 < \omega < 1$, то говорят о недостаточно быстрой релаксации

При определенных математических ограничениях, которым удовлетворяют матрицы, получающиеся в методах конечных разностей только при $1 < \omega < 2$ этот метод сходится быстрее метода Гаусса-Зейделя.

Если ρ_{Jacobi} спектральный радиус итерационной схемы Якоби, квадрат его – спектральный радиус метода Гаусса-Зейделя), то оптимальное значение ω имеет вид:

$$\omega = \frac{2}{1 + \sqrt{1 - \rho_{\text{Jacobi}}^2}} \quad (4.19)$$

Спектральный радиус при этом равен

$$\rho_{\text{SOR}} = \left(\frac{\rho_{\text{Jacobi}}}{1 + \sqrt{1 - \rho_{\text{Jacobi}}^2}} \right)^2$$

(4.20)

Если использовать выражение для радиуса Якоби из формулы (4.11), то получим:

$$\omega \approx \frac{2}{1 + \pi/J}$$

(4.21)

(4.22)

$$\rho_{\text{SOR}} \approx 1 - \frac{2\pi}{J}$$

Для достижения точности 10^{-p} необходимо следующее число итераций:

$$r \approx \frac{pJ \ln 10}{2\pi} \approx \frac{1}{3} pJ$$

(4.23)

Отсюда следует, что метод SOR для достижения точности 10^{-p} требует количества итераций, пропорционального J , а не J^2 .

При помощи этого численного метода можно с успехом решать граничные задачи для уравнений в частных производных.

Диаграммы классов

Детали проектного решения можно отобразить средствами UML на статических структурных диаграммах — диаграммах классов. Диаграммы классов строятся на стадии проектирования каждого цикла разработки. До начала их создания необходимо построить следующее.

- Диаграммы взаимодействий, по которым разработчик определяет, какие классы должны быть задействованы в проектном решении, а также методы этих классов

- Концептуальную модель, на основе которой разработчик детализирует определения классов

Хотя диаграммы классов создаются после диаграмм взаимодействия, на самом деле они зачастую разрабатываются параллельно. Имена многих классов, методов и типы отношения можно определить уже на начальной стадии этапа проектирования, до построения диаграмм взаимодействия, с помощью шаблонов распределения обязанностей.

Диаграмма классов (design class diagram) иллюстрирует спецификации программных классов и интерфейсов в приложении. Обычно на такую диаграмму выносятся следующая информация.

- Классы, ассоциации и атрибуты
- Интерфейсы со своими операциями и константами
- Методы
- Информация о типах атрибутов
- Способы навигации
- Зависимости

В отличие от концептуальной модели диаграммы классов отображают определения программных сущностей, а не понятия предметной области. В языке UML существуют специальные обозначения для диаграммы классов.

Для построения диаграммы классов используется следующая стратегия.

1. Определите все классы, задействованные в программном решении. Для этого проанализируйте диаграммы взаимодействий.
2. Отобразите их на диаграмме классов.
3. Перенесите на диаграмму атрибуты соотв. понятий из концептуальной модели.
4. Добавьте имена методов на основе анализа диаграмм взаимодействия.
5. Добавьте информацию о типах атрибутов и методов.
6. Добавьте ассоциации, необходимые для поддержки обеспечения видимости посредством атрибутов.
7. Добавьте стрелки, определяющие направление навигации для ассоциаций.
8. Добавьте линии зависимостей, определяющие другие способы обеспечения видимости, отличные от видимости посредством атрибутов.