

Метод SOR (метод последовательной верхней релаксации)

Мы получим лучший алгоритм – один из самых распространенных до семидесятых годов прошлого века - если мы скорректируем величину $\mathbf{x}^{(r)}$ на r -м шаге итераций Гаусса-Зейделя. Из метода Гаусса-Зейделя следует:

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

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

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

Для улучшения сходимости введем так называемый параметр «сверх релаксации» 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 over relaxation).

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:

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

Спектральный радиус при этом равен

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

Если использовать выражение для радиуса Якоби, то получим:

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

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

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.

Рассмотрим эллиптическое уравнение второго порядка относительно двух переменных x и y и напомним разностную схему для квадратной области. Каждой строке матрицы A в матричном уравнении $A\mathbf{x}=\mathbf{b}$ соответствует выражение

$$a_{j,l}u_{j+1,l} + b_{j,l}u_{j-1,l} + c_{j,l}u_{j,l+1} + d_{j,l}u_{j,l-1} + e_{j,l}u_{j,l} = f_{j,l}$$

Для рассматриваемой нами задачи $a = b = c = d = 1$, $e = 4$.

Итерационную формулу можно написать в виде

$$u_{j,l}^* = \frac{1}{e_{j,l}} (f_{j,l} - a_{j,l}u_{j+1,l} - b_{j,l}u_{j-1,l} - c_{j,l}u_{j,l+1} - d_{j,l}u_{j,l-1})$$

Средневзвешенное значение имеет вид

$$u_{j,l}^{\text{new}} = \omega u_{j,l}^* + (1 - \omega)u_{j,l}^{\text{old}}$$

Невязку для нашей задачи можно записать:

$$\xi_{j,l} = a_{j,l}u_{j+1,l} + b_{j,l}u_{j-1,l} + c_{j,l}u_{j,l+1} + d_{j,l}u_{j,l-1} + e_{j,l}u_{j,l} - f_{j,l}$$

Используя формулу метода SOR, можно записать

$$u_{j,l}^{\text{new}} = u_{j,l}^{\text{old}} - \omega \frac{\xi_{j,l}}{e_{j,l}}$$

Эту формулу можно очень легко запрограммировать и норма невязки может служить критерием окончания итерационного процесса.

Узлы сетки можно разбить на четные и нечетные, при этом окажется, что значения в четных узлах зависят только от значений в нечетных узлах и наоборот. В программе, приведенной ниже, такое разбиение используется.

Асимптотическая оценка скорости сходимости метода SOR может служить лишь для оценки порядка количества итераций; в реальности количество итераций может быть в 20 раз больше асимптотической оценки.

В методе SOR с ускорением Чебышева используется динамическое вычисление коэффициента релаксации

$$\omega^{(0)} = 1$$

$$\omega^{(1/2)} = 1/(1 - \rho_{\text{Jacobi}}^2/2)$$

$$\omega^{(n+1/2)} = 1/(1 - \rho_{\text{Jacobi}}^2 \omega^{(n)}/4), \quad n = 1/2, 1, \dots, \infty$$

$$\omega^{(\infty)} \rightarrow \omega_{\text{optimal}}$$

Важным достоинством этой реализации метода SOR является уменьшение нормы ошибки на каждой итерации (имеется в виду ошибка при вычислении решения)

Текст программы на языке C приведен ниже

```
#include <math.h>
#define MAXITS 1000
#define EPS 1.0e-5
void sor(double **a, double **b, double **c, double
**d, double **e,
double **f, double **u, int jmax, double rjac)
Successive overrelaxation solution with Chebyshev
acceleration. a, b, c
d, e, and f are input as the coefficients of the equation,
each dimensioned to the grid size
[1..jmax][1..jmax]. u is input as the initial guess to the
solution, usually zero, and returns
with the final value. rjac is input as the spectral radius of
the Jacobi iteration, or an estimate
of it.
{
void nrrerror(char error_text[]);
int ipass,j,jsw,l,lsw,n;
double anorm,anormf=0.0,omega=1.0,resid;
Double precision is a good idea for jmax bigger than
about 25.
for (j=2;j<jmax;j++)
```

Compute initial norm of residual and terminate iteration when norm has been reduced a factor EPS.

```
for (l=2;l<jmax;l++)
anormf += fabs(f[j][l]); Assumes initial u is zero.
for (n=1;n<=MAXITS;n++) {
anorm=0.0;
jsw=1;
for (ipass=1;ipass<=2;ipass++) { Odd-even ordering.
lsw=jsw;
for (j=2;j<jmax;j++) {
for (l=lsw+1;l<jmax;l+=2) {
resid=a[j][l]*u[j+1][l]
+b[j][l]*u[j-1][l]
+c[j][l]*u[j][l+1]
+d[j][l]*u[j][l-1]
+e[j][l]*u[j][l]
-f[j][l];
anorm += fabs(resid);
u[j][l] -= omega*resid/e[j][l];
}
lsw=3-lsw;
}
jsw=3-jsw;
omega=(n == 1 && ipass == 1 ? 1.0/(1.0-
0.5*rjac*rjac) :
1.0/(1.0-0.25*rjac*rjac*omega));
}
if (anorm < EPS*anormf) return;
}
nrerror("MAXITS exceeded");
}
```

Главным достоинством метода является простота использования, а главным недостатком — недостаточно быстрая сходимость при решении больших задач

Схема Кранка – Николсона.

Если в схеме FTCS для одномерного уравнения диффузии в правой части взять значения функции в момент времени $n+1$, то получим полностью неявную схему.

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D \left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2} \right]$$

Параметр ξ для этой схемы имеет вид:

$$\xi = \frac{1}{1 + 4\alpha \sin^2 \left(\frac{k\Delta x}{2} \right)}$$

Эта схема устойчива при любых шагах по времени.

При решении одномерного уравнения диффузии можно сочетать устойчивость неявной схемы с точностью второго порядка по пространству и времени. Представим одномерное уравнение диффузии в разностном виде следующим образом:

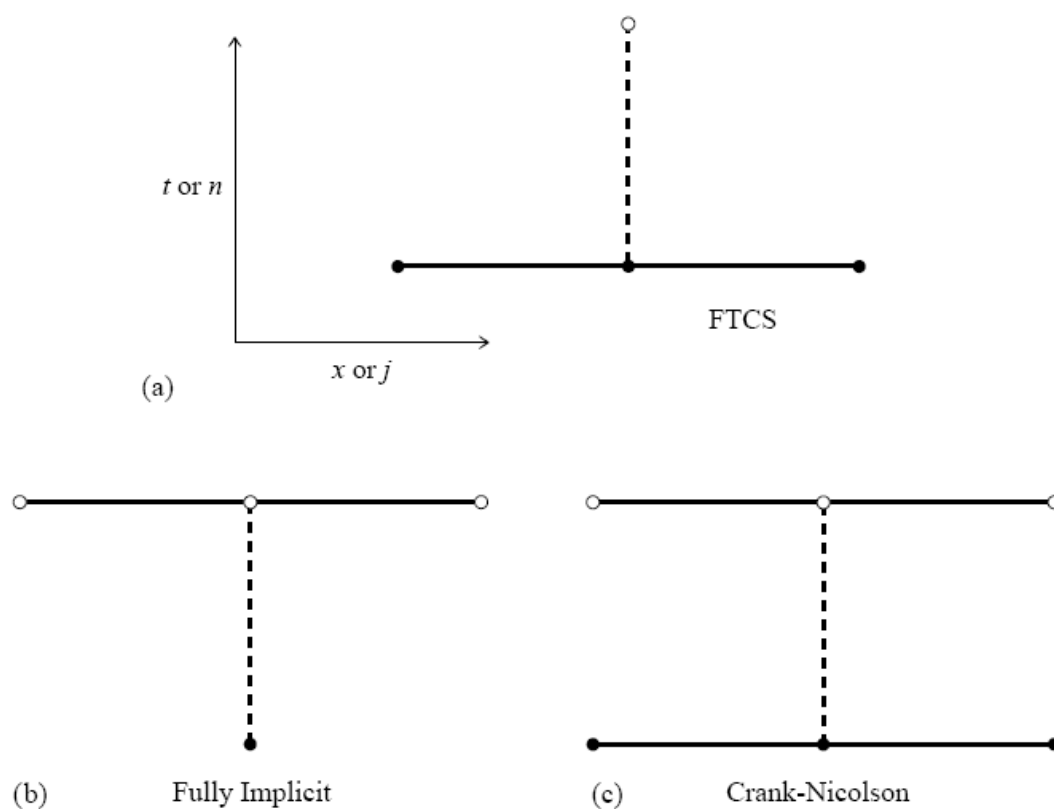
$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D}{2} \left[\frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{(\Delta x)^2} \right]$$

Параметр ξ для исследования устойчивости по критерию Неймана имеет вид:

$$\xi = \frac{1 - 2\alpha \sin^2 \left(\frac{k\Delta x}{2} \right)}{1 + 2\alpha \sin^2 \left(\frac{k\Delta x}{2} \right)}$$

Отсюда следует, что схема стабильна при любом шаге по времени. Эта схема называется схемой Кранка

– Николсона и часто применяется для решения диффузионных задач.



На этом рисунке представлены три схемы для решения уравнения диффузии. (a) – FTCS – первого порядка точности, но стабильна только при малых шагах по времени, (b) – стабильная полностью неявная схема первого порядка точности, (c) – схема Кранка - Николсона второго порядка точности стабильная при больших шагах по времени.

Рассмотрим двумерную задачу диффузии

$$\frac{\partial u}{\partial t} = D \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Применим схему Кранка – Николсона к двумерной задаче

$$u_{j,l}^{n+1} = u_{j,l}^n + \frac{1}{2}\alpha \left(\delta_x^2 u_{j,l}^{n+1} + \delta_x^2 u_{j,l}^n + \delta_y^2 u_{j,l}^{n+1} + \delta_y^2 u_{j,l}^n \right)$$

где

$$\alpha \equiv \frac{D \Delta t}{\Delta^2} \quad \Delta \equiv \Delta x = \Delta y$$

$$\delta_x^2 u_{j,l}^n \equiv u_{j+1,l}^n - 2u_{j,l}^n + u_{j-1,l}^n$$

Аналогично определяется

$$\delta_y^2 u_{j,l}^n$$

Далее надо решить систему линейных уравнений. Система уже не является трехдиагональной, как в одномерном случае, но является сильно разреженной. Ее можно решать, используя специальную технику применимую для разреженных матриц.

Существует другой способ обобщения одномерного метода Кранка - Николсона на двумерную задачу. При этом также используется схема второго порядка точности по времени и пространству и безусловно стабильная, однако в этом случае получаются алгебраические уравнения, которые решаются проще.

При этом каждый шаг по времени делится на два шага длиной $\Delta t/2$ и во время одного шага изменяется одна независимая переменная, например x , а во время другого – переменная y

$$u_{j,l}^{n+1/2} = u_{j,l}^n + \frac{1}{2}\alpha \left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^n \right)$$
$$u_{j,l}^{n+1} = u_{j,l}^{n+1/2} + \frac{1}{2}\alpha \left(\delta_x^2 u_{j,l}^{n+1/2} + \delta_y^2 u_{j,l}^{n+1} \right)$$

При этом на каждом шаге надо решать трехдиагональную систему

Методы Фурье и методы «Циклического продолжения» (Cyclic reduction) для решения граничных задач.

Метод преобразования Фурье - быстрый метод решения уравнений в частных производных применим только к уравнениям с постоянными коэффициентами. «Cyclic reduction» методы применимы к уравнениям с разделяющимися переменными. Кроме того, в обоих случаях, границы области моделирования должны совпадать с линиями координатной сетки. Для ряда задач возможно применение комбинации этих методов – метода FACR. Рассмотрим эти методы.

Дискретное обратное преобразование Фурье для x и y имеет вид:

$$u_{jl} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \hat{u}_{mn} e^{-2\pi i j m / J_e - 2\pi i l n / L}$$

Правую часть уравнения также можно разложить в ряд Фурье:

$$\rho_{jl} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \hat{\rho}_{mn} e^{-2\pi i j m / J_e - 2\pi i l n / L}$$

Если мы подставим эти выражения в разностное уравнение для теплопроводности, то получим:

$$\hat{u}_{mn} (e^{2\pi i m / J} + e^{-2\pi i m / J} + e^{2\pi i n / L} + e^{-2\pi i n / L} - 4) = \hat{\rho}_{mn} \Delta^2$$

или

$$\hat{u}_{mn} = \frac{\hat{\rho}_{mn} \Delta^2}{2(\cos \frac{2\pi m}{J} + \cos \frac{2\pi n}{L} - 2)}$$

Стратегия решения уравнения методом Фурье состоит в следующем:

1. Вычисляем коэффициенты Фурье правой части

$$\hat{\rho}_{mn} = \sum_{j=0}^{J-1} \sum_{l=0}^{L-1} \rho_{jl} e^{2\pi i m j / J} e^{2\pi i n l / L}$$

2. Вычисляем коэффициенты разложения u_{mn}

3. Вычисляем u_{ij} из обратного преобразования Фурье

Изложенная выше процедура справедлива для периодических граничных условий

$$u_{jl} = u_{j+J, L} = u_{j, l+L} \quad (3.7)$$

Рассмотрим сначала граничные условия Дирихле $u=0$ на прямоугольной области, тогда вместо (3.2) получим:

$$u_{jl} = \frac{2}{J} \frac{2}{L} \sum_{m=1}^{J-1} \sum_{n=1}^{L-1} \hat{u}_{mn} \sin \frac{\pi j m}{J} \sin \frac{\pi l n}{L} \quad (3.8)$$

Вычислим коэффициенты Фурье правой части

$$\hat{\rho}_{mn} = \sum_{j=1}^{J-1} \sum_{l=1}^{L-1} \rho_{jl} \sin \frac{\pi j m}{J} \sin \frac{\pi l n}{L}$$

Затем коэффициенты Фурье функции u :

$$\hat{u}_{mn} = \frac{\Delta^2 \hat{\rho}_{mn}}{2(\cos \frac{\pi m}{J} + \cos \frac{\pi n}{L} - 2)}$$

При помощи обратного преобразования Фурье вычисляем решение.

Если граничные условия неоднородны, то находится решение однородного уравнения, удовлетворяющее граничным условиям. При этом, граничное условие также раскладывается в ряд Фурье, а затем решение уравнения с неоднородными граничными условиями представляется в виде суммы решений уравнения с однородными граничными условиями и решения однородного уравнения с неоднородными граничными условиями.

