§4. Неявные схемы и алгоритм прогонки

Как уже было сказано, реализация неявной схемы Эйлера требует решения на каждом временном слое системы уравнений (линейных, если исходное уравнение в частных производных линейно, и нелинейных, если оно нелинейно). На примере расчетов прошлого параграфа мы убедились, что обычно неявные схемы устойчивее явных, поэтому часто идут на существенное усложнение при их реализации, связанное с необходимостью решать систему алгебраических уравнений на каждом слое.

Перепишем неявную разностную схему Эйлера (5) для уравнения теплопроводности

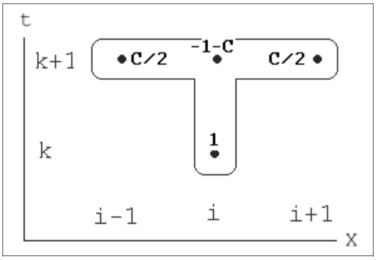

Рис. 8. *Шаблон неявной схемы*

$$\frac{u_i^{j+1}-u_i^j}{\tau}=D\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{\Delta^2} \qquad (16)$$

в следующем виде:

$$A_i u_{i-1}-B_i u_i+C_i u_{i+1}=F_i \quad \cdot \qquad (17)$$

Мы сняли верхний индекс, который относится к (j=1)-му, т.е. верхнему, временному слою, оставив в качестве u_i неизвестные значения сеточной функции с верхнего слоя. Значения с нижнего слоя, которые известны, мы заключили в коэффициент F_i. Остальные коэффициенты очевидным образом выражаются через шаги по времени и пространству и коэффициент диффузии D. Их явный вид, выраженный через число Куранта, приведен на рис. 8 в виде соответствующих меток узлов шаблона.

Мы получили разностные уравнения, которые связаны системой линейных алгебраических уравнений (17). Повторимся, что для завершения формирования замкнутой системы линейных уравнений достаточно дополнить ее соответствующим представлением начальных и граничных условий.

Выпишем теперь матрицу, соответствующую данной системе уравнений:

$$\begin{pmatrix} B_1 & 0 & 0 & 0 & .. & 0 \\ A_2 & B_2 & C_2 & 0 & .. & 0 \\ 0 & A_3 & B_3 & C_3 & .. & 0 \\ ... & ... & ... & ... & .. & ... \\ 0 & 0 & 0 & 0 & .. & B_{M-1} \end{pmatrix} . \qquad (18)$$

Легко заметить, что эта матрица является *трехдиагональной*. Каждое из линейных уравнений содержит коэффициент B_i, находящийся на диагонали, и два элемента по бокам от диагонального элемента – A_i и C_i. Все остальные коэффициенты этой матрицы равны нулю. Как известно, для численного решения систем линейных алгебраических уравнений применяется метод исключения Гаусса.

Важно, что система линейных уравнений (17), аппроксимирующая уравнение в частных производных, является хорошо обусловленной. Для хорошо обусловленных линейных систем метод исключения Гаусса является оптимальным, а специальные алгоритмы (см. главу 2) нужны для плохо обусловленных систем.

Алгоритм исключения Гаусса, как известно, приводит к необходимости проведения порядка N^2 арифметических операций. Между тем, в данном случае при реализации неявной разностной схемы его можно модифицировать таким образом, чтобы существенно сократить количество арифметических операций. Такая возможность появляется из-за того, что матрица систем уравнений является трехдиагональной, и большинство ее элементов – это нули. В этом случае метод Гаусса можно модифицировать таким образом, чтобы количество арифметических операций составляло величину не порядка N^2, а порядка 3N.

Экономичный вариант метода Гаусса можно получить, исключив из него шаги, содержащие умножение на ноль. Однако, мы продемонстрируем вывод данного метода, который называется методом *прогонки*, в несколько другом виде. Алгоритм прогонки был предложен во времена «холодной войны» независимо российскими и американскими учеными.

Предположим, что значения u_{i+1} выражается через u_i следующим образом:

$$u_{i+1}=\alpha_i u_i+\beta_i \tag{19}$$

и подставим его в канонический вид системы линейного уравнения (17). Такая подстановка уберет из уравнения (17) упоминание о u_{i+1} и получит соотношение, связывающее u_{i-1} и u_i. Теперь перепишем получившееся соотношение, как функцию пересчета u_i через u_{i-1}:

$$u_i=\gamma_i A_i u_{i-1}+\gamma_i(C_i\,\beta_i-D_i), \tag{20}$$

где

$$\gamma_i\equiv 1/(B_i+C_i\alpha_i). \tag{21}$$

Если повторить те же самые выкладки уже не для i-й, а для (i-1)-й точки, то получим следующие выражения для α_{i-1} и β_{i-1}:

$$\alpha_{i-1}=A_i\gamma_i, \tag{22}$$

$$\beta_{i-1}=\gamma_i(C_i\,\beta_i-D_i). \tag{23}$$

Эти два соотношения (22-23) являются основными формулами алгоритма прогонки. Пользуясь (20), мы получим очень простой рецепт разрешения всей системы линейных уравнений. Будем исходить из того, что для индекса i=N значение сеточной функции u_N известно. Значение сеточной функции на правой границе интервала даст нам коэффициент $\alpha_{N-1}\equiv 0$, а также коэффициент β_{N-1}, равный граничному условию u_N, что следует непосредственно из основного предположения

$$u_{N+1}=\alpha_N u_N+\beta_N. \tag{24}$$

Значение u_N равно правому граничному условию и не зависит от u_{N-1}. Естественно, в этом случае $\alpha_{N-1}=0$, а $\beta_{N-1}=u_N$. Таким образом, все искомые α_{N-1}, β_{N-1} и γ_{N-1} мы знаем. Коэффициенты α_{N-1}, β_{N-1} мы только что обсудили, а γ_{N-1} пересчитывается через них. В то же время, основные формулы метода прогонки выражают α_{i-1}, β_{i-1} через, соответственно, через α_i, β_i. В итоге, зная α_{N-1}, β_{N-1}, мы по этим соотношениям сразу же пересчитаем α_{N-2}, β_{N-2}.

Далее, зная коэффициенты для (N-2)-й точки мы можем по тем же самым соотношением пересчитать их для (N-3)-й точки, и так далее, двигаясь по рекуррентным соотношениям, дойти до i=1. Всего нам придется сделать порядка 2N арифметических операций. Этот шаг рекуррентного пересчета коэффициентов α и β через соответствующие коэффициенты для более высоких индексов называется *обратным ходом* прогонки.

Второй, и заключительный, шаг заключается в прямом ходе прогонки. Естественно, если мы теперь знаем все α и β для любого индекса i, мы можем, по-просту, пересчитать все искомые значения сеточной функции, двигаясь уже по индексу i от 0 до N. Значения сеточной функции на левой границе интервала мы знаем, т.к. u_0 берется из граничного условия. Согласно (19), мы сразу пересчитаем следующее u_1. Затем, двигаясь вверх по индексу, мы таким простым способом узнаем все искомые значения сеточной функции на следующем слое: все u_{i+1} через u_i.

В результате, при реализации неявной разностной схемы нам придется затратить порядка 3N арифметических действий, вместо N^2 действий, как в обычном алгоритме Гаусса. Для сетки, состоящей из 100 узлов по пространственной координате, нам придется затратить 300 арифметических операций, вместо более 10^4 операций, необходимых для использования метода Гаусса.