

## Лекция 8. Жесткие системы обыкновенных дифференциальных уравнений

В данной части курса рассмотрим наиболее популярные методы решения типовых вычислительных задач. Будем рассматривать задачи по мере возрастания их сложности в той же последовательности. Что и в курсах вычислительной математики – от систем обыкновенных дифференциальных уравнений к системам уравнений в частных производных.

Рассматриваем задачу Коши для системы обыкновенных дифференциальных уравнений (ОДУ)

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}),$$

$$\mathbf{x}(0) = \mathbf{x}_0.$$

В 1952 году Кертисом и Хиршфельдером было открыто явление жесткости. Под ЖЕСТКИМИ системами понимались те, которые не решались стандартными явными методами. После того стали появляться специальные ( неявные) методы решения ОДУ. Теория жестких систем и некоторые методы решения описаны в [19].

Основные направления развития численных методов для жестких систем.

1. Конструирование новых численных методов и совершенствование математического аппарата исследования на устойчивость (первые методы исследования устойчивости – Далквист).

2. Совершенствование существующих численных методов с учетом специфики решаемых задач. (Пример – специальные методы для решения АВТОНОМНЫХ систем ОДУ или систем с запаздыванием).

3. Конструирование численных методов, пригодных для параллельной реализации.

Раньше – основной источник жестких систем ОДУ – задачи химической кинетики. Затем – биология. Пример – фотосинтез, система ОДУ примерно 130 уравнений, время счета – десятки часов. Теперь – новые электронные устройства и проектирование СБИС, число уравнений в системе может составлять десятки тысяч.

Пример метода для автономной системы, практически не допускающий параллельной реализации – методы Розенброка с комплексными коэффициентами. Они требуют точного обращения матрицы, а процедура вычисления обратной матрицы практически не распараллеливается.

Рассмотрим методы только для АВТОНОМНОЙ системы, выписанная выше система уравнений может быть легко сведена к автономной с помощью установления следующего соответствия:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}), \Rightarrow \frac{d\mathbf{y}}{dt} = \mathbf{g}(\mathbf{y}), \mathbf{y}^T = (\mathbf{x}^T, y_{N+1}),$$

$$\mathbf{g}^T(\mathbf{y}) = (\mathbf{f}^T(y_{N+1}, \mathbf{x}), 1),$$

$$\mathbf{x}(0) = \mathbf{x}_0. \Rightarrow \mathbf{y}(0) = (\mathbf{x}_0, 1)^T$$

T — символ транспонирования.

Далее будем рассматривать только задачу Коши для автономной системы.

Рассмотрим семейство одношаговых методов Рунге – Кутты

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

где вспомогательные векторы вычисляются по правилам

$$\mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^s a_{ij} \mathbf{k}_j).$$

Тогда получается s-стадийный неявный метод Рунге-Кутты.

Достоинства семейства методов. Путем соответствующего выбора коэффициентов можно построить схемы с хорошими свойствами по аппроксимации и устойчивости. Недостаток. Для вычисления всех вспомогательных векторов необходимо решать нелинейную систему алгебраических уравнений размера  $Ns \times Ns$ ,  $N$  — размерность фазового пространства решаемой системы ОДУ,  $s$  — число стадий метода. Набор неявных методов Рунге-Кутты не обладает потенциалом для распараллеливания, отличается низкой экономичностью.

Упрощение формул. Пусть теперь, как и выше,

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

но  $\mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^i a_{ij} \mathbf{k}_j)$ . Тогда система нелинейных уравнений распадется на несколько систем (столько, сколько стадий метода) размерности  $N \times N$ . В этом случае метод называется диагонально неявным методом. Появляется возможность распараллелить процесс – каждый вектор может определяться как предел последовательности итераций на собственном исполнителе. Мешает только зависимость от предыдущих вспомогательных векторов, появляется значительное количество пересылок векторных

данных и элементы конвейера. Естественно, при таком упрощении мы несколько проигрываем по сравнению с исходным случаем в порядке аппроксимации и в устойчивости.

Запишем последнее равенство в виде  $\mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j + \tau a_{ii} \mathbf{k}_i)$ . Если считать, что шаг дискретизации задачи – малый параметр (а именно так оно и есть!), то можно написать приближенное равенство  $\mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) \cdot \tau a_{ii} \mathbf{k}_i$  или

$$(\mathbf{E} - \tau a_{ii} \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j)) \mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j)$$

Теперь для определения вспомогательных векторов надо решать линейные системы! Мы просто в явном виде ввели в решение один шаг итераций по методу Ньютона, такие методы называются одноитерационными. Правда, в системе выше необходимо на каждой итерации и для каждой стадии вычислять матрицу Якоби, а затем решать систему линейных уравнений!

Следующее упрощение – давайте считать матрицу Якоби только один раз, линеаризуем систему в окрестности значения на текущем временном шаге. «Зато» введем еще один набор аппроксимационных коэффициентов. Теперь метод будет выглядеть как

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

$$(\mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}^n)) \mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \tau \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}^n) \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j$$

Такая конструкция называется методом Розенброка. Мы не останавливаемся на способах определения коэффициентов метода, данный вопрос описан в специальной литературе и многочисленных журнальных публикациях. Отметим только, что очередное упрощение конструкции численного метода снова приводит к ухудшению (правда, не очень значительному) аппроксимационных свойств и сужению области устойчивости.

Следующее упрощение конструкции— W методы (см. [19]).

Теперь считаем, что

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

$$\mathbf{W} \mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \tau \mathbf{A} \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j$$

$$\mathbf{W} = (\mathbf{E} - \tau d_{ii} \mathbf{A})$$

Здесь уже  $\mathbf{A}$  — произвольная матрица, такая, что  $\mathbf{W}$  — обратимая матрица (то есть невырожденная матрица, обладающая обратной матрицей с «разумной» нормой). Конечно, желательно, чтобы  $\mathbf{A} \approx \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{y}^n)$ , и чем лучше выбрано приближение якобиана, тем лучше метод. Но это только «практическое» требование, не входящее в определение метода.

Основная предпосылка к созданию таких методов — желание обращать матрицу Якоби системы не на каждой итерации, а один раз в несколько временных шагов, затем запоминать эту матрицу и использовать ее до нового пересчета.

Первая идея параллельности в использовании  $\mathbf{W}$  методов — использование аппроксимации матрицы Якоби в виде разложения  $\mathbf{A} = \mathbf{Q}\mathbf{S}^T$  где  $\mathbf{Q}, \mathbf{S} \in \mathbb{R}^{n \times k}$ .

Вслед за [20,21] введем понятие *достаточно хорошей оценки* обратной матрицы.

Определение. Пусть  $\mathbf{B} \in \mathbb{R}^{n \times n}$  и невязка приближения  $\mathbf{R}(\mathbf{B}) = \mathbf{E} - \mathbf{B}\mathbf{A}$  такова, что в выбранной норме  $\|\mathbf{R}(\mathbf{B})\| < 1$ . Тогда матрица  $\mathbf{B}$  называется *достаточно хорошей оценкой* для матрицы  $\mathbf{A}^{-1}$

Справедлива следующая теорема [2]. Пусть  $\mathbf{B}$  — достаточно хорошая оценка для матрицы  $\mathbf{A}^{-1}$ . Если  $\mathbf{S}_k = (\mathbf{E} + \mathbf{R}(\mathbf{B}) + \mathbf{R}^2(\mathbf{B}) + \dots + \mathbf{R}^k(\mathbf{B}))\mathbf{B}$ , то при  $k \rightarrow \infty$

$$\|\mathbf{A}^{-1} - \mathbf{S}_k\| \leq \frac{\|\mathbf{B}\| \cdot \|\mathbf{R}(\mathbf{B})\|^k}{1 - \|\mathbf{R}(\mathbf{B})\|} \rightarrow 0.$$

Заметим, что при доказательстве теорем в [20] используется норма, согласованная с нормой вектора  $\|\mathbf{x}\|^2 = (\mathbf{x}, \mathbf{x})$ .

На основе этой теоремы строится быстросходящийся итерационный метод обращения матриц (метод Шульца). Для вычисления обратной матрицы приходим к итерационному процессу

$$\mathbf{R}_m = \mathbf{E} - \mathbf{B}_m \mathbf{A}$$

$$\mathbf{B}_{m+1} = (\mathbf{E} + \mathbf{R}_m + \mathbf{R}_m^2 + \dots + \mathbf{R}_m^k) \mathbf{B}_m$$

Заметим, что в данном итерационном процессе используются только перемножения матриц, то есть операции, по сути своей параллельные.

Рассмотрим теперь построение w-метода, обладающего существенным потенциалом для распараллеливания. Пусть, как и выше,

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

$$\mathbf{W}_i \mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \tau \mathbf{A} \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j$$

$$\mathbf{W}_i = (\mathbf{E} - \tau d_{ii} \mathbf{A})$$

Здесь учтено, что для разных вспомогательных векторов матрица  $\mathbf{W}$ , вообще говоря, может отличаться. Конечно, самыми экономичными будут ОДНОКРАТНО-ДИАГОНАЛЬНО неявные методы Розенброка, для которых выполнено  $d_{11} = \dots = d_{kk} = \dots = d_{ss} = d$ .

Учтем теперь, что сами матрицы, приближающие матрицы Якоби, могут меняться от шага к шагу.

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau \sum_{k=1}^s b_k \mathbf{k}_k,$$

$$\mathbf{W}_i^n \mathbf{k}_i = \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \tau \mathbf{A}^n \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j,$$

$$\mathbf{W}_i^n = (\mathbf{E} - \tau d_{ii} \mathbf{A}^n).$$

Пусть при вычислении значений решения на  $n$  временном слое уже известна матрица  $\mathbf{B}_i^n = (\mathbf{E} - \tau d_{ii} \mathbf{A}^n)^{-1}$ . Разрешим систему линейных уравнений для каждого вспомогательного вектора, мы имеем

$$\mathbf{k}_i = \mathbf{B}_i^n \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \tau \mathbf{B}_i^n \mathbf{A}^n \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j,$$

причем это выражение можно существенно упростить. Действительно, так как  $\mathbf{B}_i^n (\mathbf{E} - \tau d_{ii} \mathbf{A}^n) = \mathbf{E}$ , то  $\tau d_{ii} \mathbf{B}_i^n \mathbf{A}^n = \mathbf{B}_i^n - \mathbf{E}$

и

$$\mathbf{k}_i = \mathbf{B}_i^n \left( \mathbf{f}(\mathbf{y}^n + \tau \sum_{j=1}^{i-1} a_{ij} \mathbf{k}_j) + \frac{1}{d_{ij}} \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j \right) - \frac{1}{d_{ij}} \sum_{j=1}^{i-1} d_{ij} \mathbf{k}_j.$$

Теперь матрица  $\mathbf{B}_i^{n+1} = (\mathbf{E} - \tau d_{ii} \mathbf{A}^{n+1})^{-1}$ , необходимая на следующем шаге по времени, должна быть определена по методу Шульца.

Пусть шаг по времени мал и есть основания полагать, что

$$\|\mathbf{R}_i^{n+1}\| = \left\| \mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right) \right\| < 1. \text{ Если это условие не выполнено, то следует}$$

уменьшить шаг по времени. В случае, если условие выполнено, построим итерацию по методу Шульца, используя минимальную частичную сумму матричного ряда. Тогда

$$\mathbf{R}_i^{n+1} = \mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right),$$

$$\mathbf{B}_i^{n+1} = (\mathbf{E} + \mathbf{R}_i^{n+1})\mathbf{B}_i^n = \left[ 2\mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right) \right] \mathbf{B}_i^n,$$

причем можно уточнить последовательность приближений, введя дополнительные итерации

$$\mathbf{R}_i^{n+1} = \mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right),$$

$$\mathbf{B}_i^{1^{n+1}} = (\mathbf{E} + \mathbf{R}_i^{n+1})\mathbf{B}_i^n = \left[ 2\mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right) \right] \mathbf{B}_i^n,$$

$$\mathbf{B}_i^{s+1^{n+1}} = (\mathbf{E} + \mathbf{R}_i^{n+1})\mathbf{B}_i^{s^{n+1}} = \left[ 2\mathbf{E} - \mathbf{B}_i^n \left( \mathbf{E} - \tau d_{ii} \frac{\partial \mathbf{f}(\mathbf{y}^{n+1})}{\partial \mathbf{y}} \right) \right] \mathbf{B}_i^{s^{n+1}}.$$

Таким образом, процедура обращения матрицы (затратная и плохо распараллеливаемая) заменена итерационной процедурой (быстро сходящейся, как показано в [20]). При этом на каждой итерации требуется лишь два перемножения матриц. Такая операция обладает значительным потенциалом для распараллеливания.

Изложение в данной главе основано на материалах [21].

Краткая справка об алгоритмах перемножения матриц

Пусть даны квадратные матрицы большой размерности. Их произведение вычисляется как

$$\mathbf{C} = \mathbf{A}\mathbf{B}$$

или покомпонентно

$$c_{ij} = \sum_{k=1}^N a_{ik} b_{kj}$$

(Учет при распараллеливании специфики организации языка) Например, в Фортране элементы матриц размещены по столбцам  $a_{11}, a_{21}, a_{31}, \dots, a_{n1}, a_{12} \dots$

Кэш используется наиболее эффективно, если при изменении параметра цикла переход осуществляется в соседнюю ячейку. Поэтому в Фортране цикл  $jki$  будет эффективнее  $ijk$ .

Количество операций при перемножении матриц будет составлять  $n$  умножений и  $n$  сложений при вычислении каждого элемента, всего элементов есть  $N^2$ , тогда асимптотический анализ немедленно дает  $O(N^3)$ .

Без ограничения общности считаем, что число строк в матрице четное. Тогда можно записать

$$\mathbf{A} = \begin{matrix} D & E \\ F & G \end{matrix}, \quad \mathbf{B} = \begin{matrix} H & P \\ Q & R \end{matrix}, \quad \mathbf{C} = \begin{matrix} DH + EQ & DP + ER \\ FH + GQ & FP + GR \end{matrix}$$

(в свою очередь каждый блок можно тоже разделить на блоки)

Эффективность возрастает за счет использования верхних (быстрых) уровней памяти. (раскладываем на всех уровнях – алгоритм Штрассена, асимптотика есть  $N^{2,81}$ )  
Есть ли за что бороться? Да, это существенный выигрыш!

Распараллеливание (декомпозиция матрицы – аналог геометрического распараллеливания в задачах матфизики).

1. Нарезка матриц на полосы.
2. Нарезка на блоки (Штрассена)
3. Нарезка в трехмерной области – труднее реализуется, приводит к еще более благоприятным асимптотическим оценкам (2,3) (практически не используется – все портит большая константа в асимптотической оценке алгоритма).