

О. В. Шаповалов

РАЗРАБОТКА МЕТОДОВ РАСЧЕТА ЗАДАЧИ ТЕПЛОПРОВОДНОСТИ В КОНТЕКСТЕ МУЛЬТИФИЗИЧЕСКОГО ПОДХОДА

В статье рассматривается вопрос разработки методов расчета теплопроводности в контексте совместного расчета многотельной модели и теплопроводности. Для использования в расчете задачи теплопроводности модифицирован и реализован метод Ньюмарка. Разработан метод для расчета жестких систем (в частности, задачи теплопроводности), обладающий высокой точностью расчета. Определена максимальная размерность задач, которые могут быть решены данным методом. Для расчета больших СЛАУ определенного вида был разработан алгоритм декомпозиции. *Задача теплопроводности; междисциплинарный расчет; метод Ньюмарка; метод декомпозиции*

В современных транспортных средствах важную роль играет устройство системы подвески и, в частности, гасителей колебаний. Принцип их действия заключается в последовательном перемещении вязкой жидкости поршнем через дроссельные каналы с большим гидравлическим сопротивлением, в результате чего происходит диссипация механической энергии с последующим ее рассеиванием в окружающую среду.

При движении автомобиля по неровному грунту в демпфере управляемой автомобильной подвески энергия столкновения переходит в тепловую, тем самым нагревая жидкость в амортизаторах. Для обеспечения их нормального функционирования должен соблюдаться заданный температурный режим, что обеспечивается особой конструкцией корпуса демпфирующего элемента подвески.

Процесс движения автомобиля моделируется на ЭВМ [6]. Представление уравнений динамики систем тел основано на уравнениях Лагранжа первого рода. Уравнения динамики механической системы слагаются из уравнений свободного движения тел и уравнений связей от кинематических пар. Кроме кинематических пар связи между телами могут задаваться силами от упруго – демпфирующих элементов, соединяющих тела.

Уравнения движения произвольной системы тел при таком подходе записываются в виде [5]:

$$\begin{aligned} M\ddot{x} - D^T \lambda &= f(\dot{x}, x, t), \\ D\ddot{x} &= h(x, t). \end{aligned}$$

Здесь x – вектор переменных системы размерностью n , M – постоянная матрица коэффици-

циентов при вторых производных, $f(\dot{x}, x, t)$ – вектор-функция правых частей дифференциальных уравнений, D – матрица переменных коэффициентов уравнений связи размерностью $k \times n$ (k – число связей), $h(\dot{x}, x)$ – вектор правых частей уравнений связи, λ – вектор множителей Лагранжа.

Уравнение является одной из форм записи уравнений Лагранжа 1 рода.

В ходе работы подвески жидкость в амортизаторе подвергается периодическому сжатию, что служит причиной повышения ее температуры. Совместное решение механической многотельной модели и задачи теплопроводности носит итерационный характер. На каждом шаге расчета модели определяется положение амортизатора, на основе которого рассчитывается работа, совершенная демпфером. Затем, считая, что вся энергия идет на равномерный нагрев жидкости амортизатора, можно определить ее среднюю температуру, которая будет использоваться при решении задачи нестационарной теплопроводности в качестве граничных условий с внутренней стороны амортизатора. С внешней части граничным условием будет служить температура воздуха.

Технология расчета состоит в использовании нестационарной дискретной модели, аппроксимирующей решение уравнения теплопроводности для произвольной геометрии. Особенностью метода является использование кубической сетки для разбиения геометрии.

Для моделирования процесса охлаждения необходима геометрическая модель амортизатора. Технология моделирования включает разбиение исходной геометрии на конечные элементы, расчет полученной конечно-элементной модели и визуализацию поведения модели

на основе данных, полученных при проведении расчетов.

В процессе разбиения поверхность детали, представленная исходной САД-геометрией, разделяется на полигоны. Разбиение поверхности на полигоны происходит с использованием свободной библиотеки функций для работы с геометрией OpenCASCADE. Также на первом этапе происходит разбиение модели машиностроительного изделия секущими плоскостями. Полигоны используются при построении сечения: ищутся линии пересечения полигона и секущей плоскости. Затем происходит разбиение сечения на равномерную ортогональную сетку путем проведения взаимно перпендикулярных лучей и поиска точек пересечения этих лучей.

На этапе расчета происходит интегрирование системы обыкновенных дифференциальных уравнений теплопроводности, соответствующих рассматриваемому изделию. Используется дискретное представление уравнений теплоемкости в форме, ориентированной на решение методом установления.

Дискретная математическая модель соответствует уравнению теплопроводности в частных производных:

$$\gamma\rho\frac{\partial u}{\partial t} - q\nabla^2 u = Q(x, y, z, t), \quad (1)$$

где u – температура, q – коэффициент теплопроводности материала, γ – теплоемкость, ρ – плотность, с начальными условиями

$$u(x, y, z, 0) = f(x, y, z) \quad (2)$$

и граничными условиями

$$u(x, y, z, t)|_r = T_b. \quad (3)$$

В качестве методов решения данной задачи использовались различные подходы. В качестве первого подхода рассматривалась простейшая разностная схема на основе явного метода интегрирования Рунге–Кутты четвертого порядка.

Уравнение (1) можно с заданной точностью представить в виде

$$\alpha\frac{\partial^2 u}{\partial t^2} + \gamma\rho\frac{\partial u}{\partial t} - q\nabla^2 u = Q(x, y, z, t). \quad (4)$$

При $\alpha \rightarrow 0$ решение уравнения (4) стремится к решению уравнения (1). Необходимое условие для величины α :

$$\alpha < \frac{\gamma^2 \rho^2}{4q\lambda_n^2}. \quad (5)$$

В методе Фурье уравнению (4) соответствует уравнение для функции времени

$$\alpha\ddot{T}_n(t) + \gamma\rho\dot{T}_n(t) + q\lambda_n^2 T_n(t) = 0; \quad n = 1, 2, \dots, \quad (6)$$

где λ_n – собственное значение, $T_n(t)$ – некоторая функция времени, соответствующая собственным значениям λ_n . При достаточно малых α решение уравнения (6) стремится к решению уравнения

$$\gamma\rho\dot{T}_n(t) + q\lambda_n^2 T_n(t) = 0. \quad (7)$$

Дискретизация уравнения (4) по методу конечных элементов приводит к уравнению вида

$$M\ddot{X} + C\dot{X} + KX = G(x, t), \quad (8)$$

где X – вектор узловых переменных размерности n , $M_{n \times n}$ – матрица при вторых производных, $C_{n \times n}$ – матрица демпфирования, $K_{n \times n}$ – матрица жесткости, $G(x, t)$ – вектор узловых нагрузок, соответствующий источникам тепла.

Матрица жесткости в предлагаемом методе складывается из элементов, которые для связи двух соседних узлов записываются в виде

$$k = q/l/9, \quad (9)$$

где l – расстояние между соседними узлами.

Матрица демпфирования складывается из элементов вида

$$C_{ij} = \gamma\rho/l/9. \quad (10)$$

В рассматриваемой реализации метода предполагается, что решетка разбиения – кубическая, с кубиками одинакового размера с длиной ребра l . Использование равномерной кубической решетки позволяет интегрировать уравнение (8) без явного вычисления матрицы жесткости. Правая часть уравнения (8) для i -го узла будет вычисляться по формуле

$$\alpha\ddot{x}_i = -\sum_{j=1}^m k(x_i - x_j) - \sum_{j=1}^m c_{ij}(\dot{x}_i - \dot{x}_j) + g(x, t), \quad (11)$$

где m – число контактирующих с узлом элементов.

Таким образом, был реализован модуль расчета задачи теплопроводности с использованием явной схемы интегрирования. Описанная выше технология расчетов позволяет использовать САД-геометрию любой сложности, имеет достаточно высокую точность расчета, предоставляет высокую скорость разбиения геометрии на сетку, генерации и интегрирования дифференциальных уравнений. Выбранные методы разбиения модели и проведения расчетов допускают эффективное распараллеливание на многопроцессорных вычислительных системах. В качестве возможных путей распараллеливания были рассмотрены различные технологии.

При адаптации программы для систем с общей памятью выяснилось, что затраты на порождение параллельных потоков не слишком велики для данной задачи и параллельная программа выигрывает в скорости у последовательной версии даже на небольших сетках. В данном случае имеется в виду использование технологии OpenMP [1]. Идеальным решением для данного алгоритма стало применение технологий GPGPU [7], поскольку коммуникационные затраты в данном случае незначительны и сами потоки более легковесны, так как аппаратная структура графических процессоров рассчитана на мелкозернистый параллелизм. Для эффективной реализации алгоритма на данной платформе необходимо использование большого числа потоков, что не является проблемой для выбранного алгоритма решения. Результаты тесты для разных вариантов расчета представлены в таблице.

Таблица 1
Результаты распараллеливания задачи теплопроводности

Количество узлов разбиения, млн.	время расчета, с		
	GpuTesla c1060	1 ядро corei7 920 2,667ГГц	4 ядра corei7 920 2,667ГГц
0,67	1,38	4,13	2,74
1,02	1,59	5,7	3,69
1,3	1,7	7	4,39
1,7	2	8,7	5,3
2,2	2,36	11,4	6,65
3,05	2,8	15	8,5
4,33	3,72	19,8	11,8
6,44	4,76	28,3	15,9

Визуализация результатов интегрирования дифференциальных уравнений происходит с использованием библиотек OpenCASCADE и VTK.

На рис. 1 представлен результат расчета гидropневматической рессоры автомобиля.

При увеличении размерности задачи, а также при решении смешанных задач (механической подсистемы и задачи теплопроводности, механической подсистемы и задачи напряженно-деформированного состояния) проявились недостатки явных алгоритмов интегрирования.

Используемые в пакете ФРУНД явные методы интегрирования – метод Эйлера и метод Рунге-Кутты четвертого порядка – хорошо работают, когда скорости происходящих в системе реакций соизмеримы. В случае, когда скорость одного процесса значительно превосходит скорость другого, явные схемы перестают работать, адекватно определяя поведение только одного из происходящих процессов. Системы обыкновенных дифференциальных уравнений, описывающие совокупности таких процессов, называются жесткими. Известно, что жесткие СОДУ эффективно решаются только с помощью неявных схем интегрирования.

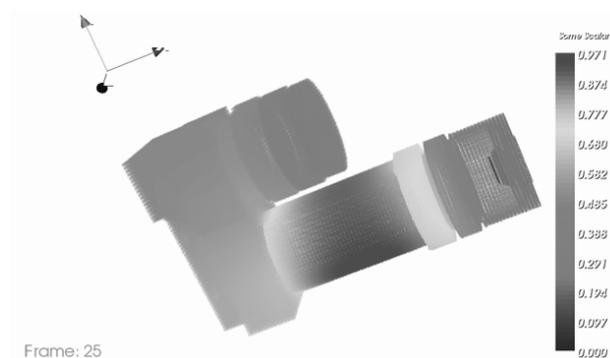


Рис. 1. Тепловой расчет гидropневматической рессоры

В качестве одного из неявных методов для реализации был выбран алгоритм Ньюмарка, который, правда, используется только в МКЭ-системах, но может быть использован после определенной модификации. Чтобы не отходить от терминологии метода будем обозначать изменение температуры перемещением, скорость изменения – скоростью, а ускорение изменения температуры – просто ускорением. Метод основан на предположении о линейном законе изменения ускорения $\ddot{w}(t)$ на интервале $[t_i, t_i + t]$ (рис. 2). Функции перемещения $w(t)$ и скорости $\dot{w}(t)$ произвольной узловой точки, совершающей колебания, представляются в виде отрезков степенных рядов

$$w(t_i + t) = w_i + t \cdot \dot{w}_i + \frac{t^2}{2} \ddot{w}_i + a_1 t^3 \ddot{\ddot{w}}_i,$$

$$\dot{w}(t_i + t) = \dot{w}_i + t \cdot \ddot{w}_i + a_2 t^2 \ddot{\ddot{w}}_i,$$

где a_1, a_2 – числовые параметры, определяющие остаточные члены степенных разложений функций $w(t)$ и $\dot{w}(t)$ на интервале $[t_i, t_i + t]$.

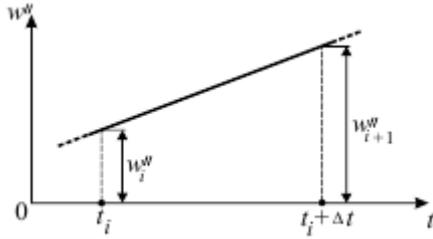


Рис. 2. Изменение ускорения на интервале $[t_i, t_i + t]$

Заменяя \ddot{w} приближенным разностным отношением $(\ddot{w}_{i+1} - \ddot{w}_i)/t$, преобразуем выражения к виду

$$w_{i+1} = w_i + t \cdot \dot{w}_i + \frac{t^2}{2} \ddot{w}_i + a_1 t^2 (\ddot{w}_{i+1} - \ddot{w}_i),$$

$$\dot{w}_{i+1} = \dot{w}_i + t \cdot \ddot{w}_i + a_2 t (\ddot{w}_{i+1} - \ddot{w}_i),$$

Найдем из первого равенства величину \ddot{w}_{i+1} :

$$\ddot{w}_{i+1} = \frac{1}{a_1 t^2} (w_{i+1} - w_i) - \frac{1}{a_1 t} \dot{w}_i + \left(1 - \frac{1}{2a_1}\right) \ddot{w}_i. \quad (12)$$

После подстановки во второе равенство, получим

$$\dot{w}_{i+1} = \frac{a_2}{a_1 t} (w_{i+1} - w_i) + \left(1 - \frac{a_2}{a_1}\right) \dot{w}_i + \frac{t}{2} \left(2 - \frac{a_2}{a_1}\right) \ddot{w}_i. \quad (13)$$

Последние два выражения являются рекуррентными соотношениями метода Ньюмарка. Для вычисления векторов-столбцов перемещений $\{W\}_{i+1}$, скоростей $\{\dot{W}\}_{i+1}$ и ускорений $\{\ddot{W}\}_{i+1}$ в момент $t_i + t = t_{i+1}$ уравнение движения записываем в виде

$$M \{\ddot{W}\}_{i+1} + C \{\dot{W}\}_{i+1} + K \{W\}_{i+1} = \{P(t)\}_{i+1}. \quad (14)$$

После подстановки в него выражений (12) и (13) и преобразования к компактной форме, получим матричное уравнение вида

$$A \{W\}_{i+1} = \{F\}_{i+1}, \quad (15)$$

где

$$A = K + \frac{a_2}{a_1 t} C + \frac{1}{a_1 t^2} M,$$

$$\{F\}_{i+1} = \{P(t)\}_{i+1} +$$

$$+ M \left[\frac{1}{a_1 t^2} \{W\}_i + \frac{1}{a_1 t} \{\dot{W}\}_i + \left(\frac{1}{2a_1} - 1 \right) \{\ddot{W}\}_i \right] +$$

$$+ C \left[\frac{a_2}{a_1 t} \{W\}_i + \left(\frac{a_2}{a_1} - 1 \right) \{\dot{W}\}_i \right] + \frac{Ct}{2} \left(\frac{a_2}{a_1} - 2 \right) \{\ddot{W}\}_i.$$

В начале алгоритма Ньюмарка, исходя из заданного шага интегрирования t , формируется матрица A , после чего она приводится к треугольному виду и до завершения вычислительного процесса не изменяется. В данном случае существует особенность

$$M \{\ddot{W}\}_{i+1} = \{B(t)\}_{i+1}. \quad (16)$$

Сопоставляя формулы (15) и (16), получим

$$\{B(t)\}_{i+1} = \{P(t)\}_{i+1} - C \{\dot{W}\}_{i+1} - K \{W\}_{i+1},$$

$$\{P(t)\}_{i+1} = \{B(t)\}_{i+1} + C \{\dot{W}\}_{i+1} + K \{W\}_{i+1}.$$

Таким образом, на каждом шаге интегрирования необходимо помимо, собственно, вычисления правых частей, проводить два дополнительных умножения разреженной матрицы на вектор. Это, вместе с необходимостью решения СЛАУ, создает существенные по времени вычислительные сложности, и метод существенно проигрывает по скорости методам явного интегрирования.

Векторы-столбцы узловых ускорений и скоростей вычисляем по формулам:

$$\{\ddot{W}\}_{i+1} = \frac{1}{a_1 t^2} (\{W\}_{i+1} - \{W\}_i) - \frac{1}{a_1 t} \{\dot{W}\}_i + \left(1 - \frac{1}{2a_1}\right) \{\ddot{W}\}_i,$$

$$\{\dot{W}\}_{i+1} = \{\dot{W}\}_i + [(1 - a_2) \{\ddot{W}\}_i + a_2 \{\dot{W}\}_{i+1}] t.$$

Для обеспечения устойчивости алгоритма прямого интегрирования параметры a_1 и a_2 должны удовлетворять следующим условиям: $a_2 \geq 0$ и $a_1 \geq 0,25(a_2 + 0,5)^2$. Без учета искусственного рассеивания энергии максимальная точность метода Ньюмарка достигается при $a_1 = 0,25$ и $a_2 = 0,5$.

Для запуска алгоритма метода Ньюмарка требуется задать в момент времени $t = 0$ векторы-столбцы $\{W\}_0$, $\{\dot{W}\}_0$, $\{\ddot{W}\}_0$ (начальные условия). В начальный момент времени будем считать скорости и ускорения нулевыми, а перемещения (температуры) должны задаваться. По умолчанию температура будет равна температуре окружающей среды.

Ключевым моментом решения является решение СЛАУ, после сравнения различных реализаций решение СЛАУ было реализовано с использованием библиотеки MKL, что, с одной стороны, позволило сделать расчет быстрым и точным, но в то же время ограничило максимальный размер решаемых задач. Для реализации потребовалось вычислять матрицы жесткости и демпфирования и хранить в явном виде.

К безусловным плюсам данного метода следует отнести то, что схема прямого интегрирования, основанная на методе Ньюмарка, является безусловно устойчивой и неявной. К минусам – данная схема имеет только второй порядок точности, поэтому не всегда обеспечивает необходимую точность. Также, помимо уже упомянутой большой вычислительной сложности, существует ограничение на размер решаемых задач, примерно, в 500 тыс. узлов разбиения.

Для преодоления ограничения на размерность было принято решение разработать алгоритм декомпозиции [3]. Задача теплопроводности относится к задачам с одной степенью свободы, так как рассчитываемая система тел является жесткой, и структура не меняется со временем. Таким образом, было необходимо разработать новый метод расчета жестких систем. В общем случае, система тел описывается следующей системой Лагранжа 1 рода:

$$\begin{aligned} M\ddot{x} - D^T\lambda &= f(\dot{x}, x, t), \\ D\ddot{x} &= h(\dot{x}, x). \end{aligned}$$

Для жесткой системы связи имеют более простой вид, и записывается следующим образом:

$$\begin{aligned} M\ddot{x} - D^T\lambda &= f(\dot{x}, x, t), \\ D\ddot{x} &= 0. \end{aligned}$$

Особенностью жесткой системы является ее полносвязность, в связи с чем, решая систему относительно $\begin{pmatrix} \ddot{x} \\ \lambda \end{pmatrix}$ все x оказываются одинаковыми, т. е. $\forall i \ddot{x}_i = f / N$, где \ddot{x}_i – вектор ускорений, а N – число узлов разбиения. Это знание можно использовать для более эффективного решения системы и решать ее не относительно $\begin{pmatrix} \ddot{x} \\ \lambda \end{pmatrix}$, а относительно λ , т. е. подставляя найденные \ddot{x} в первое уравнение, получаем $D^T\lambda = M\ddot{x} - f(\dot{x}, x, t)$. Данная система для абсолютного большинства случаев является вырожденной, т. е. не имеет решения классическими методами Гаусса или Крамера. Для таких систем существуют алгоритмы решения, наиболее подходящим для данного случая является алгоритм svd-разложения.

Найдя λ , возможно определить распределение нагрузки по узлам рассматриваемой системы. При этом нагрузка может соответствовать как механической силе, так и тепловому потоку.

В данной работе метод рассматривался применимо к тепловой задаче, \ddot{x}_i соответствует средней скорости нагревания тела, а λ – тепловому потоку между соседними узлами.

У предлагаемого метода есть свои достоинства и недостатки. К достоинствам однозначно можно отнести большую точность и скорость расчета, поскольку метод является абсолютно устойчивым, так как погрешность не накапливается в связи с тем, что распределение температуры считается только в момент записи результатов. Этим же объясняется и высокая скорость расчета, поскольку время расчета не зависит от параметров расчета (времени интегрирования, шага расчета), а зависит от количества кадров времени, которые нужно записать. Но у метода также есть существенные недостатки: svd-разложение работает только для матриц небольших размеров, и для реальных задач метод становится неприменимым.

Для устранения этого недостатка было необходимо разработать методику расчета без использования svd-разложения. Для этого имеющуюся систему уравнений требовалось каким-либо способом обусловить, чтобы определитель системы стал отличен от нуля. Рассмотрим исходную систему:

$$D^T\lambda = M\ddot{x} - f(\dot{x}, x, t) = B.$$

Примем $\lambda = Dy$, тогда система примет следующий вид:

$$D^T Dy = B,$$

где $D^T D$ – матрица жесткости системы. Теперь необходимо преобусловить матрицу, для этого достаточно добавить ненулевой элемент (от 1 до 100) к одному из диагональных элементов матрицы $D^T D$. После этого система будет хорошо обусловленной и будет решаться стандартным методом Гаусса.

Эта методика была реализована, для решения СЛАУ использовалась функция Pardiso для решения СЛАУ для разреженных систем из библиотеки MathKernelLibrary. Полученный метод обладает безусловной устойчивостью, высокой скоростью и точностью. Недостатком метода является ограничение на размерность решаемой системы, поскольку Pardiso не решает системы с количеством ненулевых элементов больше 10 млн.

Проведенные исследования показали, что Pardiso позволяет обрабатывать с достаточной точностью матрицы с не более чем 10 миллионами ненулевых элементов (невязка составляет

менее 1 %), при этом требуется порядка 8 Гб оперативной памяти. Увеличение размерности до 15 миллионов ненулевых элементов дает погрешность 7 %, что уже неприемлемо для большинства задач моделирования. Поскольку существует необходимость работы с большими массивами данных, надо каким-либо образом оптимизировать процесс решения СЛАУ с целью уменьшения требуемого объема памяти и времени решения. Одним из таких методов является замена одного решения СЛАУ сверхбольшой размерности [2] на решение нескольких СЛАУ меньшей размерности, основанное на знании структуры портрета матрицы.

Решаемая матрица представляет собой уравнения Эйлера для условного экстремума функционала, включающего в себя функции одной независимой переменной и их производные, в форме системы дифференциально-алгебраических уравнений:

$$\begin{aligned} M\ddot{x} - D^T \lambda &= f(\dot{x}, x, t), \\ D\ddot{x} &= h(x, t). \end{aligned}$$

Зная, что M – диагональная матрица, рассмотрим случай цепной системы (рис. 3) тел, когда все тела последовательно жестко связаны между собой и имеют единичную массу.

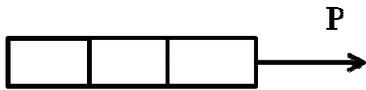


Рис. 3. Цепная система тел со связями

В простейшем случае узлы будут обладать одной степенью свободы, т. е. могут перемещаться только вдоль основного вектора тела. Данной ситуации будет соответствовать граф (рис. 4), где точки соответствуют узлам тела, а λ – коэффициенты взаимодействия между узлами.

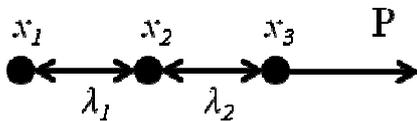


Рис. 4. Граф взаимодействия узлов в теле

Данному графу будет соответствовать следующая система дифференциально-алгебраических уравнений.

$$\begin{aligned} \ddot{x}_1 + \lambda_1 &= 0, \\ \ddot{x}_2 - \lambda_1 + \lambda_2 &= 0, \\ \ddot{x}_3 - \lambda_2 &= P, \\ \dot{x}_1 - \dot{x}_2 &= 0, \\ \dot{x}_2 - \dot{x}_3 &= 0. \end{aligned}$$

Та же система в матричной форме будет иметь следующий вид.

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ P \\ 0 \\ 0 \end{pmatrix}.$$

Из графа видно, что каждый узел связан максимум с двумя соседними, а крайние узлы связаны только с одним телом. Если начать рассчитывать поведение модели с одного из крайних узлов, можно последовательно исключить все неизвестные системы. Таким образом, общее решение системы будет состоять из прямого хода, где на каждом шаге решение будет выражаться через одну переменную λ_i , и обратного хода. Обратный ход начинается после того, как будет решена последняя СЛАУ прямого хода, решение которого уже не будет зависеть ни от каких переменных. Во время обратного хода происходит последовательная постановка уже рассчитанных значений, данный этап полностью соответствует соответствующему этапу алгоритма Гаусса.

Для осуществления одного шага прямого хода необходимо решить систему из двух уравнений.

$$\begin{aligned} \ddot{x}_i - \lambda_{i-1} + \lambda_i &= 0, \\ \dot{x}_{i-1} - \dot{x}_i &= 0. \end{aligned}$$

Поскольку зависимость для переменных x_{i-1} и λ_{i-1} установлена на предыдущем шаге (для первого шага $x_1 = f(\lambda_1) = -\lambda_1$), подставляя выражение для x_{i-1} , получаем систему из двух уравнений и трех неизвестных, которое можно записать следующим образом:

$$A \cdot \begin{pmatrix} \ddot{x}_i \\ \lambda_{i-1} \end{pmatrix} = \begin{pmatrix} -\lambda_i \\ C \end{pmatrix},$$

где A – матрица подсистемы, C – константа, полученная сложением силы, приложенной к данному узлу, и постоянной части решения из предыдущего шага.

Пользуясь свойством линейности,

$$\begin{pmatrix} \ddot{x}_i \\ \lambda_{i-1} \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} -\lambda_i \\ C \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} 0 \\ C \end{pmatrix} - \lambda_i A^{-1} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

т. е. каждый раз надо решать одну систему уравнений для двух правых частей. На последнем шаге прямого хода в системе не будет λ_N и решение для $\begin{pmatrix} \ddot{x}_N \\ \lambda_{N-1} \end{pmatrix}$ будет определено однозначно. Обратный ход представляет собой подстановку и достаточно тривиален.

Был проведен ряд тестов для цепочек разной длины, результаты тестов приведены в табл. 2.

Таблица 2
Расчет протяженного тела

Размерность	Погрешность, %	Время расчета, с
1,0E+05	0	0,09
1,0E+06	1,0E-04	0,89
1,0E+07	1,0E-02	8,9
1,0E+08	3,42	89,6
1,0E+09	12,73	894

Для сравнения была выбрана самая быстрая на сегодняшний момент библиотека для решения СЛАУ для разреженных матриц IntelMKLPardiso [4]. Для решения СЛАУ сверхбольшой размерности ей требуется значительный объем памяти, поэтому тесты проводились на вычислительном сервере кафедры ЭВМ и систем Волгоградского государственного технического университета, обладающем 16Гб оперативной памяти и процессором IntelCorei7 720. Данного объема памяти достаточно для решения системы размерностью пять миллионов.

Таблица 3
Расчет цепной системы в MKLPardiso

Размерность	Погрешность, %	Время расчета, с
1,0E+05	0	0,964
1,0E+06	1,0E-04	9,46
3,0E+06	0,3	29,27
5,0E+06	6,01	51,006

Предложенный метод расчета имеет более высокую скорость расчета по сравнению со

стандартными библиотеками, обладает более высокой точностью и требует меньшего количества памяти. Данный метод может быть обобщен на случай любой полносвязной структуры тела, если описывающий его граф не имеет колец (в таком случае определитель матрицы будет равен нулю, и система не будет иметь однозначного решения). Сейчас реализован расчетный модуль для решения любой системы с одной степенью свободы.

Для использования в расчете задачи теплопроводности модифицирован и реализован метод Ньюмарка, обладающий безусловной устойчивостью. Определены границы применения этого метода. Разработан метод для расчета жестких систем (в частности, задачи теплопроводности). Метод имеет безусловную устойчивость и высокую точность расчета. Определена максимальная размерность задач, которые могут быть решены данным методом. Для расчета больших систем определенного вида был разработан алгоритм декомпозиции, позволяющий существенно ускорить расчет, также увеличилась точность расчета, и максимальная размерность решаемых задач.

СПИСОК ЛИТЕРАТУРЫ

1. Антонов А. С. Параллельное программирование с использованием технологии OpenMP: учеб. пособие. М.: Изд-во МГУ, 2009. 77 с.
2. Баладин М. Ю., Шурина Э. П. Методы решения СЛАУ большой размерности: учеб. пособие. Новосибирск: НГТУ, 2000. 76 с.
3. Банах Л. Я., Горобцов А. С., Чесноков О. К. Условия разбиения системы дифференциально-алгебраических уравнений на слабосвязанные подсистемы / Л. Я. Банах // Журнал вычислительной математики и математической физики. 2006. Т. 6, № 12. С. 2223–2227.
4. Библиотека PARDISO. [Электронный ресурс]. Режим доступа: <http://www.pardiso-project.org>. 2011.
5. Виттенбург Й. Динамика систем твердых тел. М.: Мир, 1980. 294 с.
6. Официальный сайт пакета ФРУНД [Электронный ресурс]. 2012. Режим доступа: <http://frund.vstu.ru>.
7. Центр разработки NVIDIA CUDA. [Электронный ресурс]. Режим доступа: http://www.nvidia.com/object/cuda_home_new.html.

ОБ АВТОРЕ

Шаповалов Олег Владимирович, асп. каф. ЭВМ и систем Волгоградск. гос. техн. ун-та.