После подстановки результатов разложения в исходную систему уравнений для расчета полного НДС многослойных оболочек, получаем систему обыкновенных дифференциальных уравнений для пары волновых чисел m и n для каждого шага по времени. Осуществляя интегрирование полученной системы уравнений с использованием метода дискретной ортогонализации [6], позволяющего автоматически удовлетворять условиям идеального механического контакта слоев, а также суммируя тригонометрические ряды разложения напряжений σ_{11}, σ_{12}, σ_{22}, σ_{23}, σ_{33}, получаем решение задачи о трехмерном НДС многослойной оболочки.

Прогнозирование состояния конструкции осуществляется на основании критериев (1), (2) для полученных значений растягивающих напряжений в зонах концентрации напряжений и областях возможного образования трещин.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Бартенев Г.М.* Прочность и механизм разрушения полимеров. – М.: Химия, 1984. – 280 с.
2. *Новожилов В.В.* Основы нелинейной теории упругости. –М.: –Л.: Гостехиздат, 1984. – 212 с.
3. *Григоренко Я.М., Василенко А.Т., Панкратова Н.Д.* Статика анизотропных толстостенных оболочек. – Киев.: Вища школа, 1985. – 190 с.
4. *Бакулин В.Н.* Использование уравнений трехмерной теории упругости для решения задач динамики многослойных оболочек /Известия вузов. Авиационная техника, 1985. № 3. – С. 7-12.
5. *Бартенев Г.М.* Сверхпрочные и высокопрочные неорганические стекла. – М.: Стройиздат, 1974. – 240 с.
6. *Григолюк Э.И., Шалашилин В.И.* Проблемы нелинейного деформирования: Метод продолжения решения по параметру в нелинейных задачах механики твердого деформируемого тела. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 232 с.

В.В. Владимиров, Н. С. Звягинцев, Д.В. Граждан

РАЗВИТИЕ АЛГОРИТМА ДИСКРЕТНОГО КВАТЕРНИОННОГО ПРЕОБРАЗОВАНИЯ ДЛЯ МНОГОМЕРНЫХ ПРОСТРАНСТВ

Рост объема вычислений, свойственный современному уровню развития информационных технологий, неразрывно связан с разработкой новых методов и алгоритмов вычислений. В частности, это актуально для решения задач многомерных преобразований. Одним из перспективных путей решения данного класса задач являются итерационные алгоритмы дискретного вращения вектора [1]. Однако эти алгоритмы, как правило, строятся на основе плоского базового оператора. В работах [2-4] предложен итерационный алгоритм дискретного кватернионного преобразования, выгодно реализующий вращение вектора в трехмерном пространстве. Логика построения этого алгоритма дает возможность его развития для многомерных пространств.

Пусть $\{e_i\}$, $i=\{1...n\}$ – канонический базис n-мерного евклидова пространства, $\overline{x}=\{x_i\}$, $i=\{1...n\}$ – исходный вектор, а $\overline{\omega}=\{\omega_i\}=const$ – угловая скорость вращения вектора $\overline{x}$, определенного своими проекциями в базисе $\{e_i\}$. Рассмотрим определитель

$$F=\begin{vmatrix} e_1 & . & . & . & e_n \\ \xi_{21} & . & . & . & \xi_{2n} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ \xi_{(n-2)1} & . & . & . & \xi_{(n-2)n} \\ \omega_1 & . & . & . & \omega_n \\ x_1 & . & . & . & x_n \end{vmatrix}, \qquad (1)$$

где символы ξ_{ij} принимают значения произвольных чисел, которые, в общем случае, образуют систему из $(n-3)$-х линейно независимых векторов. Условия, при которых символы ξ_{ij} принимают конкретные значения, излагаются ниже.

Выполняя последовательное разложение определителя (1) по минорам первой, n-ой и $(n-1)$-й строки получим выражение

$$F_i(t)=\sum_{i=1}^{n}(-1)^{i+1}e_i\sum_{\substack{j=1\\ j\neq i}}^{n}(-1)^{n-1+j}x_j\sum_{\substack{k=1\\ k\neq j\neq i}}^{n}(-1)^{n-2+k}\omega_k\times$$

$$\times\begin{vmatrix} \xi_{21} & . & . & \xi_{2(k-1)} & \xi_{2(k+1)} & \xi_{2n} \\ . & & & . & . & . \\ . & & & . & . & . \\ \xi_{(n-2)1} & . & . & \xi_{(n-2)(k-1)} & \xi_{(n-2)(k+1)} & \xi_{(n-2)n} \end{vmatrix}. \qquad (2)$$

Для перехода в выражении (2) от угловой скорости к углам вращения выполним интегрирование (2) по параметру t на интервале $\Delta t=t_2-t_1$:

$$\Delta x_i=\sum_{i=1}^{n}(-1)^{i+1}e_i\sum_{\substack{j=1\\ i\neq j}}^{n}(-1)^{n-1+j}x_j\sum_{\substack{k=1\\ k\neq i\neq j}}^{n}(-1)^{n-2+j}\Delta\alpha_k\lambda_{ijk}, \qquad (3)$$

где

$$\alpha_k=\omega_k\Delta t \qquad (4)$$

– угол вращения вокруг оси ω_k, а λ_{ijk} – определитель, полученный из (1) удалением первой, $(n-1)$-й, $(n-2)$-й строки и i-го, j-го, k-го столбца; $i=\{1...n\}$, $j=\{1...n\},\}$, $k=\{1...n\}$, $i\neq j\neq k$, причем индекс i принимает n значений, j – $n-1$ значений, а k – $n-2$ значений.

Значения Δx_i, $i=(1 \ldots n)$ представляют собой приращения вектора $\bar{x}$ на интервале Δt, а вектор $\bar{x}' = \{x_i'\} = \bar{x}(t_1 + \Delta t) = x(t_2)$, с учетом (3) и (4), определяется следующим образом

$$\bar{x}' = \bar{x} + \Delta\bar{x} = A\bar{x}\,, \tag{5}$$

где $\Delta\bar{x} = \{\Delta x_i\}$ – вектор приращений, а матрица A равна

$$A = \begin{pmatrix} 1 & (-1)^{n+2}\sum\limits_{k=3}^{n}\Delta\alpha_k\lambda_{1,2,k} & . & . & (-1)^{2n}\sum\limits_{k=3}^{n-1}\Delta\alpha_k\lambda_{1,n,k} \\ (-1)^{n+3}\sum\limits_{k=3}^{n}\Delta\alpha_k\lambda_{2,1,k} & 1 & . & . & (-1)^{2n+1}\sum\limits_{\substack{k=1\\k\neq 2}}^{n-1}\Delta\alpha_k\lambda_{2,n,k} \\ . & & . & & \\ . & & & . & \\ (-1)^{2n+1}\sum\limits_{j=k}^{n}\Delta\alpha_k\lambda_{n,1,k} & . & . & . & 1 \end{pmatrix}. \tag{6}$$

Из (6) видно, что каждый a_{ij} элемент матрицы A состоит из $k = n–2$ $(k = 1 \ldots n;\ k \neq j \neq i)$ элементов – углов вращения, умноженных на соответствующий определитель λ_{ijk}. Каждая i-я строка матрицы A содержит $j = n–1$ $(k = 1 \ldots n;\ j \neq i)$ углов вращения.

Матрица $A' = A - 1$ кососимметрична, так как $a'_{ij} = -a'_{ji}$ (учитывая, что $\lambda_{i,j,k} = \lambda_{j,i,k}$). Исследование матрицы $A = 1+A'$ показали, что при малых значениях интервала Δt ($\Delta t \to 0$) матрица A является ортогональной [5]. Таким образом, по аналогии с трехмерным пространством выражение (5) с матрицей (6) при бесконечно малых изменениях параметра t в выражении (4) задает элементарное вращение (бесконечно малое ортогональное преобразование) в n-мерном евклидовом пространстве на углы $\Delta\alpha_i$, $i = (1 \ldots n)$ вокруг осей ω_i, $i = (1 \ldots n)$. Отметим, что свойства ортогональности и кососимметричности матрицы A инвариантны относительно значений $\lambda_{i,j,k}$. Значения операторов $\lambda_{i,j,k}$ определяет многообразие вращений в n-мерном пространстве. Поэтому не ограничивая общности представления вращений, зададим $\lambda_{i,j,k} = \{-1,0,1\}$, в зависимости от вида преобразования (6). Очевидно, что для любого из заданных $\lambda_{i,j,k}$ можно найти такие значения ξ_{ij}, чтобы удовлетворялось равенство (3).

Выражение (5) с матрицей (6) позволяет организовать итерационный алгоритм, реализующий вращение вектора $\bar{x}$ в n-мерном пространстве вокруг оси вращения $\bar{\omega} = \{\omega_i\}$ на углы α_i, $i = (1 \ldots n)$ с шагами

$$\Delta\alpha_i = \omega_i\Delta t = M_i 2^{-s}\,, \tag{7}$$

где $M_i = \dfrac{\Delta\alpha_i}{max\{\Delta\alpha_i\}}$ – масштабы, обеспечивающие одновременное по всем углам приведение вектора $\bar{x}$ в конечную точку.

Алгоритм вращения имеет вид

$$\begin{gathered} \bar{x}_{h+1} = A\bar{x}_h\,; \\ \Delta\alpha_{i,h+1} = \Delta\alpha_{i,h} + \lambda_{ijk(h+1)}\Delta\alpha_i\,; \\ \bar{x}_o = \bar{x}\,; \\ \Delta\alpha_{i,0} = 0\,; \\ \lambda_{ijk} = \begin{cases} sign(\alpha_i - \Delta\alpha_{i,h}), \ldots при \ldots \alpha_i - \Delta\alpha_{i,h} \neq 0, \\ 0, \ldots при \ldots \Delta\alpha_{i,h} - \alpha_i = 0, \end{cases} \end{gathered} \tag{8}$$

где h – номер итерации.

При $\lambda_{ijk} = 0$ получим координаты преобразованного вектора $\bar{x}_{h+1}$.

Покажем вычислительные возможности алгоритма (8) многомерного вращения вектора на примере решения системы n линейных уравнений

$$C\bar{x} = \bar{b}\,, \tag{9}$$

где $C = \begin{pmatrix} c_{11} & . & . & c_{1n} \\ . & & & . \\ . & & & . \\ c_{n1} & . & . & c_{nn} \end{pmatrix}$ – невырожденная матрица и $\bar{b} = \begin{pmatrix} b_1 \\ . \\ . \\ b_n \end{pmatrix}$ – вектор свободных членов.

Система (9) решается последовательным приведением матрицы C к верхнему треугольному виду. Для этого введем индекс $m=1,2\ldots n-1$ и зададим условия формирования значений оператора вращения:

$$\lambda_{i,j,k} = \begin{cases} 1, \ldots при \ldots i \cup j = n - m + 1, \\ 0 \ldots при \ldots i \cup j \neq n - m + 1. \end{cases} \tag{10}$$

Направление вращения определяется знаком соответствующего угла из (14). Далее, дополним (10) условием

$$k = max\{i, j\} + 1\,, \tag{11}$$

определяющим один из углов в суммах (6), и, если $max\{i, j\} = n$, то $k = 2$ (в этом случае индекс k принимает значение следующего «свободного» индекса $\{i,j\}$). С учетом (10) (11), а также переиндексации $m = k-1$, матрица вращения (6) сводится к виду

$$T_m = \begin{pmatrix} 1 & 0 & . & . & 0 & 0 & . & . & 0 \\ 0 & 1 & . & . & 0 & 0 & . & . & 0 \\ . & . & . & & & & & & \\ . & . & & . & & & & & \\ 0 & 0 & . & . & 1 & \Delta\alpha_{m(m+1)} & . & . & \Delta\alpha_{mn} \\ 0 & 0 & . & . & -\Delta\alpha_{(m+1)m} & 1 & .. & . & 0 \\ . & . & & & . & & . & & \\ . & . & & & . & & & . & \\ 0 & 0 & . & . & -\Delta\alpha_{(m+1)n} & 0 & . & . & 1 \end{pmatrix}. \tag{12}$$

Для каждого значения *m*, *m = 1…n–1* выполняется обратные преобразование [6] вида

$$C_m = T_m C_{m-1} \text{ и } \bar{b}_{m+1} = T_m \bar{b}_m \ , \qquad (13)$$

где $C_0 = C$, $\bar{b}_0 = \bar{b}$, а элементарные углы вращения матрицы T_m определяются следующим образом

$$\Delta\alpha_{mr} = -\Delta\alpha_{rm} = M_{mr} 2^{-s} = -M_{rm} 2^{-s} = c_{rm}^{(m-1)}\ , \ r = m+1, m+2 \ldots n \ . \qquad (14)$$

В условии (14) верхний индекс элемента $c_{rm}^{(m-1)}$ отмечает его принадлежность к матрице C_{m-1}.

В результате преобразований (13) *n–m* элементов *m*-го столбца матрицы C_{m-1} сводятся к нулю (*n–m* элементов считаются снизу). При *m = n–1* матрица C_m является треугольной. С учетом (12) – (14) алгоритм приведения матрицы *C* к треугольному виду заключается в следующем (итерационные индексы *h* записаны сверху):

$$\begin{aligned}
&m = 1 \ldots n-1; \\
&C_0 = C; \\
&\bar{b}_0 = \bar{b}; \\
&C_{m-1}^{h} = T_m C_{m-1}^{h-1}; \\
&C_{m-1}^{0} = C_{m-1}. \\
&\text{При } c_{n,m}^{(m-1)(h-1)} = 0, \ \mathrm{C}_{\mathrm{m\text{-}1}}^{\mathrm{h}} = C_m.
\end{aligned} \qquad (15)$$

При *m = n – 1, C_m* – верхняя треугольная, вычисления окончены.

Согласно приведенного выше метода для решения системы из линейных уравнений необходимо выполнить *n–1* вращений (преобразование вектора $\bar{b}$ по алгоритму (15) реализуется параллельно), в каждом из которых участвуют все компоненты матрицы *C* и вектора $\bar{b}$. Для решения систем линейных уравнений известными методами вращения вектора необходимо выполнить $\frac{n}{2}(n-1)$ преобразований.

Таким образом, методы многомерного вращения вектора, предложенные в настоящей работе, представляют интерес с точки зрения сокращения и оптимизации вычислений в *n*-мерных пространствах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Hsiao S.-F., Delosme J.-M.* Parallel Singular Value Decomposition of Complex Matrices Using Multidimensional CORDIC Algorithms /IEEE Trans. On Signal Processing. 1996. – Vol. (3).
2. *Владимиров В.В.* Алгоритм сопроцессора дискретного кватернионного преобразования. /Многопроцессорные вычислительные структуры. – Таганрог: Изд-во ТРТИ, 1989. – Вып. 11.
3. *Владимиров В.В. Звягинцев Н.С.* Анализ и синтез алгоритмов дискретного вращения вектора для решения задач морской навигации //Проблемы вод-

ного транспорта. Известия вузов, Северо-Кавказский регион. – Ростов-на-Дону: РГУ, 2004.
4. *Владимиров В.В. Звягинцев Н.С.* Граждан Д. В. Вычисление синусно-косинусных сочетаний алгоритмами дискретных кватернионных преобразований //Перспективные информационные технологии и интеллектуальные системы №1 (21). – Таганрог: Изд-во ТРТУ, 2005.
5. *Звягинцев Н.С. Граждан Д.В.* Об ортогональности бесконечно малого преобразования вектора в n-мерном евклидовом пространстве // Сб. научных трудов. Вып. 12. – Новороссийск: МГА им. адм. Ф.Ф. Ушакова, 2007.
6. *Владимиров В.В. Звягинцев Н.С.* Вычислительные возможности алгоритма трехмерного дискретного вращения вектора //Проблемы водного транспорта. Известия вузов, Северо-Кавказский регион. – Ростов-на-Дону: РГУ, 2004.

Я.Е. Ромм, А.А. Лабинцева

ПОИСК КОРНЕЙ ПОЛИНОМА С ПЕРЕМЕННЫМИ КОМПЛЕКСНЫМИ КОЭФФИЦИЕНТАМИ

В [1, 2] изложен метод применения распараллеливаемой внутренней сортировки по ключу (ниже – сортировки) для локализации и приближенного вычисления нулей функций. В частности, в произвольно фиксированной области метод позволяет программно локализовать и с высокой точностью вычислить все нули многочлена, включая случай их плохой отделенности. Степень многочлена и его коэффициенты могут быть произвольными в границах числового диапазона языка программирования. Метод обладает параллелизмом, имеет практические приложения.

Ниже ставится задача быстрой программной локализации области всех нулей многочлена с одновременным их вычислением без использования априорной информации о границах области. При этом задача относится к случаю, когда коэффициенты многочлена являются переменными величинами и могут иметь комплексные значения.

Решение задачи конструируется на основе сортировки слиянием. Известные схемы последней [3, 4] для этой цели модифицируются. На основе матриц сравнения (МС) [5] при выполнении модифицированного слияния достигается максимальный параллелизм, устойчивость, прямая и обратная адресация к входным и выходным индексам. Представленная ниже сортировка не перемещает ключи, используя перемещение их индексов, не ограничена целой степенью двойки для числа элементов.

Конструируемая сортировка и реализующая ее программная процедура именуются *sort*. Сортировка основана на адресном слиянии двух упорядоченных массивов по МС, пример которой представляет таблица: