

Рис. 2

Из рис. 1, 2 очевидно, что построенный закон оптимального управления обеспечивает перевод твердого тела из заданного начального в требуемое конечное угловое положение.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Челноков Ю. Н. Кватернионные и бикватернионные модели и методы механики твердого тела и их приложения: Геометрия и кинематика движения. М.: Физматлит, 2006. 512 с.
2. Бранец В. Н., Шмыглевский И. П. Применение кватернионов в задачах ориентации твердого тела. М.: Наука, 1973. 320 с.
3. Понтрягин Л. С., Мищенко Е. Ф., Болтянский В. Г., Гамкрелидзе Р. В. Математическая теория оптимальных процессов. М.: Наука, 1983. 392 с.
4. Моисеев Н. Н. Численные методы в теории оптимальных систем. М.: Наука, 1971. 424 с.

А. А. Барышев, М. А. Федюкина

О РЕАЛИЗАЦИИ АЛГОРИТМА РЕШЕНИЯ КРАЕВОЙ ЗАДАЧИ ВИБРАЦИОННОГО ИЗГИБА ВЯЗКОУПРУГОЙ ПЛАСТИНЫ – ПОЛОСЫ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ

В статье рассматривается пластинка толщины h и шириной a , изготовленная из вязкоупругого материала. Срединная плоскость пластинки отнесена к декартовой системе координат, как показано на рис. 1.

Считается, что рассматриваемая пластинка испытывает малые деформации под действием распределенной по внешней плоскости $z = -h/2$ поперечной нагрузки:

$$q(x, t) = q_1(x) \cos \omega t + q_2(x) \sin \omega t, \quad (1)$$

где ω – частота внешних воздействий. Аналогично представляются все характеристики НДС.

Для составляющих проекций вектора смещения $u_k(x, z)$ и $w_k(x, z)$ ($k = 1, 2$) в [1] получена система уравнений, которая после перехода к безразмерным величинам преобразуется к виду (тангенциальные силы инерции не учитываются)

$$\frac{\partial^2 U_k}{\partial \zeta^2} = -\frac{h_0^2}{v'} \frac{\partial^2 U_k}{\partial \xi^2} - \frac{h_0}{(1-2\nu)} \frac{\partial^2 W_k}{\partial \xi \partial \zeta},$$

$$\frac{\partial^2 W_k}{\partial \zeta^2} = -2(1+\nu)v'(-1)^{k-1}\rho h^2\omega^2 \sum_{j=1}^2 e_{k+j-1}W_j - v'h_0^2 \frac{\partial^2 W_k}{\partial \xi^2} - \frac{h_0}{2(1-\nu)} \frac{\partial^2 U_k}{\partial \xi \partial \zeta}, \quad (2)$$

где $U_k = u_k/h$, $W_k = w_k/h$ – безразмерные составляющие компонент вектора перемещения, $\zeta = z/h$, $\xi = x/h$, $h_0 = h/a$, $v' = 0.5(1-2\nu)/(1-\nu)$, $\nu = \text{const}$ – коэффициент Пуассона, ρ – плотность.

Для понижения размерности этой системы применим метод сплайн-коллокаций [2]. Будем искать неизвестные функции в виде

$$U_k(\xi, \zeta) = \sum_{j=0}^N \varphi_j(\xi) \tilde{U}_{j+(k-1)(N-1)}(\zeta),$$

$$W_k(\xi, \zeta) = \sum_{j=0}^N \psi_j(\xi) \tilde{W}_{j+(k-1)(N-1)}(\zeta). \quad (3)$$

Здесь N – число точек коллокации, φ_j и ψ_j – линейные комбинации кубических В-сплайнов, подобранные таким образом, чтобы на краях $\xi = 0$, $\xi = 1$ выполнялись условия, например, жесткого закрепления $U_k = W_k = 0$.

Это позволяет свести краевую задачу для системы (2) к одномерной относительно вектора $\bar{Z} = \{z_r(\zeta)\}$. Данная система может быть записана

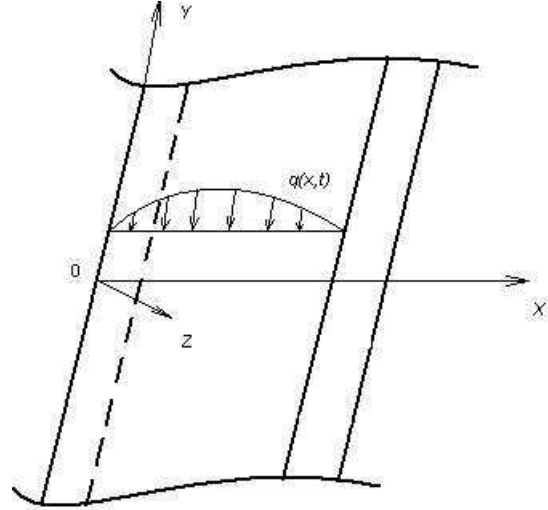


Рис. 1

в нормальной форме Коши:

$$\frac{d\bar{Z}}{d\zeta} = A(\omega)\bar{Z},$$

где $A(\omega)$ – квадратная матрица с известными компонентами,

$$Z_j(\zeta) = \tilde{U}_j(\zeta), Z_{j+2(N+1)}(\zeta) = \tilde{U}'_j(\zeta),$$

$$Z_{j+4(N+1)}(\zeta) = \tilde{W}_j(\zeta), Z_{j+6(N+1)}(\zeta) = \tilde{W}'_j(\zeta), j = 0 \dots 2\bar{N} + 1.$$

Для решения данной задачи использовался метод Виноградова [3], алгоритм которого представлен в виде блок-схемы на рис. 2.



Рис. 2

Ясно, что при увеличении числа точек коллокаций растет число уравнений в краевой задаче и, следовательно, увеличивается время вычислений. Пробные расчеты показали, что устойчивость вычислений достигается при $N \geq 80$ (время вычислений – 582,554 с). Поэтому являются актуальными вопросы, связанные с сокращением времени расчетов.

Для этого метод был реализован с использованием технологий параллельного программирования OpenMP [4] и MPI [5]. При использовании технологии OpenMP за основу берётся последовательная программа, а для создания её параллельной версии программисту предоставляется набор директив, функций и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью.

Численные расчеты проводились на персональном компьютере ASUS N52D с двухъядерным процессором AMD Turion II Dual-Core Mobile P520, с тактовой частотой 2.3 ГГц, оперативная память 3 Гб.

Из табл. 1 видно, что скорость вычисления матричной экспоненты увеличилась почти в два раза, скорость решения СЛАУ на 27%, а скорость заполнения данных на 34%. При использовании технологии параллельного программирования OpenMP ресурсы компьютера используются практически полностью, например, загрузка ЦП составляет 95–98%.

Таблица 1

	без OpenMP		с OpenMP	
	k = 100	k = 200	k = 100	k = 200
Заполнение данных, с	0,453	0,548	0,352	0,361
Вычисление матричной экспоненты, с	133,43	267,596	72,712	150,564
Решение СЛАУ, с	17,864	17,095	12,999	13,833

Код программы был собран на сервере 8x Xeon E5335 2GHz с доступной оперативной памятью 32 Гб. На сервере установлена реализация стандарта MPI – MPICH2, а также собрана библиотека PETSc (версия 3.1) (см. [5]). Кластер предоставлен Институтом наноструктур и биомеханики СГУ.

Анализируя значения табл. 2, можно сказать, что в рамках одного сервера программа, реализованная с помощью технологии OpenMP, работает быстрее, чем при использовании MPI. Например, время вычисления матричной экспоненты сократилось примерно в 4 раза (при значении $k = 100$).

Таблица 2

	8x Xeon E5335 (MPI)		8x Xeon E5335 (OpenMP)		Кластер из двух серверов 8x Xeon E5335 (MPI)	
	k = 100	k = 200	k = 100	k = 200	k = 100	k = 200
Заполнение и рассылка данных, с	0,199	0,213	0,167	0,112	1,851	1,931
Вычисление матричной экспоненты, с	13,83	31,82	4,61	31,58	9,57	21,36
Решение СЛАУ, с	1,749	1,743	1,6106	2,73	1,39	1,401

Когда расчет идет на кластере, состоящем из двух серверов, скорость вычислений матричной экспоненты увеличивается примерно на 30%, а решение СЛАУ на 20%. Таким образом, при увеличении вычислитель-

ных действий алгоритма, не связанных с пересылкой данных, время значительно сокращается.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Недорезов П. Ф. Установившиеся поперечные колебания вязкоупругой пластинки – полосы // Теоретическая и прикладная механика : науч.-техн. сб. Харьков, 2002. Вып. 35. С. 139–146.
2. Григоренко Я. М., Крюков Н. Н. Решение задач теории пластин и оболочек с применением сплайн-функций (обзор) // Прикладная механика. 1995. Т. 31, № 6. С. 3–26.
3. URL : <http://alexeivinogradov.narod.ru/>
4. Антонов А. С. Параллельное программирование с использованием технологии OpenMP. М., 2009.
5. Антонов А. С. Параллельное программирование с использованием технологии MPI. М., 2004.

УДК 629

В. Г. Бирюков, В. Ю. Вахлюев

ОПТИМАЛЬНАЯ ОСТАНОВКА ВРАЩАТЕЛЬНОГО ДВИЖЕНИЯ ТВЕРДОГО ТЕЛА

Рассмотрена задача оптимальной остановки вращательного движения твердого тела. Построен закон оптимального управления, обеспечивающий асимптотическую устойчивость неподвижного положения твердого тела и доставляющий минимум интегральному квадратичному функционалу качества переходного процесса.

1. Постановка задачи. Вращательное движение твердого тела описывается системой динамических уравнений Эйлера [1]

$$\begin{aligned} I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 &= M_1, \\ I_2\dot{\omega}_2 + (I_1 - I_3)\omega_1\omega_3 &= M_2, \\ I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 &= M_3, \end{aligned} \quad (1)$$

где I_1, I_2, I_3 – осевые моменты инерции твердого тела, $\omega_1, \omega_2, \omega_3$ – проекции вектора абсолютной угловой скорости твердого тела на оси связанной с телом системы координат, оси которой совпадают с главными центральными осями инерции твердого тела, M_1, M_2, M_3 – проекции момента внешних сил, действующих на твердое тело, на оси связанной системы координат, верхняя точка означает дифференцирование по времени.

Задача заключается в построении управляющих воздействий M_1, M_2, M_3 , переводящих твердое тело, движение которого описывается системой