

2. Kovalev, I. V. Information training technology: learner's memory state model / I. V. Kovalev, M. V. Karaseva, E. A. Souzdaleva // *Advances in Modeling : J. of AMSE Periodicals. Ser. B.* 2002. Vol. 45, № 1. P. 21–32.

3. Kovalev, I. V. Adaptive algorithm of foreign vocabulary training on the basis of lexically related components / I. V. Kovalev, M. V. Karaseva, O. V. Leskov // *Control systems and information technology : Sci. J.* 2008. № 4 (34). P. 78–82.

М. В. Карасева

## МУЛЬТИЛИНГВИСТИЧЕСКАЯ ТЕХНОЛОГИЯ. СИСТЕМНЫЕ АСПЕКТЫ ОРГАНИЗАЦИИ И ПРИМЕНЕНИЯ

*Рассмотрены системные аспекты организации мультилингвистической адаптивно-обучающей технологии и ее применение для изучения многоязычной терминологической лексики. Представлено информационно-терминологическое обеспечение этой технологии, реализованное с учетом ориентации на первичность архитектурного слоя правил, соответствующего функциям адаптивной модели обучения и многоязычному терминологическому базису.*

*Ключевые слова: мультилингвистический подход, адаптивная модель, системная архитектура, частотный терминологический словарь, адаптивно-обучающая технология.*

УДК 519.688

М. П. Варыгина, И. В. Киреев, О. В. Садовская, В. М. Садовский

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АНАЛИЗА ВОЛНОВЫХ ДВИЖЕНИЙ В МОМЕНТНЫХ СРЕДАХ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ<sup>1</sup>

*Для численного исследования динамических задач моментной теории упругости Коссера на многопроцессорных вычислительных системах разработаны параллельные алгоритмы, программная реализация которых выполнена по технологии SPMD на языке Fortran-95 с использованием библиотеки передачи сообщений MPI. Программный комплекс оснащен средствами сжатия больших массивов данных с контролируемой потерей информации, позволяющими многократно снизить сетевой трафик при копировании файлов – результатов счета с удаленного кластера и служащими для компактного хранения численных решений в постоянной памяти компьютера.*

*Ключевые слова: моментный континуум, упругие волны, высокопроизводительные вычисления.*

Моментная теория упругости Коссера [1; 2] служит для описания деформации материалов с микроструктурой: композитов, гранулированных, порошкообразных, микроразрушенных и микрополярных сред. В отличие от обычной теории упругости в ней неявно присутствует малый параметр среды – характерный размер частиц микроструктуры, поэтому при численном решении задач расчеты необходимо выполнять на сетках, размер ячеек которых меньше этого параметра. При решении динамических задач в пространственной постановке эффективными оказываются параллельные алгоритмы, поскольку они позволяют распределять вычислительную нагрузку между большим числом узлов кластера, что дает возможность существенно измельчать расчетные сетки, повышая тем самым точность численного решения.

**Математическая модель.** В модели моментной среды, кроме поступательного движения, которое характеризуется вектором скорости  $v$ , рассматриваются независимые повороты частиц с вектором угловой скорости  $\omega$ , и наряду с тензором напряжений  $u$ , компоненты которого несимметричны, вводится несимметричный тензор моментных напряжений  $m$ . Полную систему уравнений модели образуют уравнения движения, кинематические соотношения и обобщенный закон линейной теории упругости:

$$\begin{aligned} \rho \dot{v} &= \nabla \cdot \sigma + \rho g, \\ j \dot{\omega} &= \nabla \cdot m - 2\sigma^a + j q, \\ \dot{\Lambda} &= \nabla v + \omega, \quad \dot{M} = \nabla \omega, \end{aligned} \quad (1)$$

$$\begin{aligned} \sigma &= \lambda(\delta : \Lambda)\delta + 2\mu\Lambda^s + 2\alpha\Lambda^a, \\ m &= \beta(\delta : M)\delta + 2\gamma M^s + 2\epsilon M^a, \end{aligned}$$

<sup>1</sup> Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 08–01–00148), Комплексной программы фундаментальных исследований Президиума Российской академии наук № 2 «Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация» и Междисциплинарного интеграционного проекта Сибирского отделения Российской академии наук № 40.

где  $\rho$  – плотность среды;  $g$  – вектор массовых сил;  $q$  – вектор моментов;  $\Lambda$  и  $M$  – равные нулю в естественном (ненапряженном) состоянии среды тензоры деформаций и кривизны;  $j$  – инерционный параметр, равный произведению момента инерции частицы относительно оси, проходящей через ее центр тяжести, на число частиц в единице объема;  $\delta$  – символ Кронекера;  $\lambda, \mu, \alpha, \beta, \gamma$  и  $\varepsilon$  – коэффициенты упругости для изотропного материала. В (1) также используются общепринятые обозначения и операции тензорного анализа: двоеточие означает двойную свертку, точка над символом – производную по времени, звездочка – транспонирование, верхними индексами  $s$  и  $a$  отмечены симметричная и антисимметричная составляющие тензоров. С антисимметричной составляющей, где это необходимо, отождествляется соответствующий ей вектор, в частности в уравнения вращательного движения входит вектор тензора  $\sigma^a = (\sigma - \sigma^*)/2$ . Для оценки линейного параметра микроструктуры материала справедлива формула  $r = \sqrt{5j/(2\rho)}$ , основанная на модельном представлении о среде как о плотной упаковке ансамбля шарообразных частиц одинакового радиуса.

В пространственном случае система уравнений (1) включает 24 уравнения относительно 24 неизвестных функций. Можно переписать ее в матричной форме [3]:

$$A\dot{U} = B^1 U_{,1} + B^2 U_{,2} + B^3 U_{,3} + QU + G, \quad (2)$$

где  $U$  – вектор-функция, составленная из компонент векторов скорости и угловой скорости, несимметричных тензоров напряжений и моментных напряжений. В декартовой системе координат  $U$  принимает вид

$$U = (v_1, v_2, v_3, \sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{32}, \sigma_{31}, \sigma_{13}, \sigma_{12}, \sigma_{21}, \omega_1, \omega_2, \omega_3, m_{11}, m_{22}, m_{33}, m_{23}, m_{32}, m_{31}, m_{13}, m_{12}, m_{21}).$$

В вектор  $G$  входят массовые силы и моменты. Матрицы-коэффициенты  $A, B^1, B^2$  и  $B^3$  симметричны, а матрица  $Q$  антисимметрична. Матрица  $A$  положительно определена, если выполняются условия, ограничивающие допустимые значения параметров среды:

$$3\lambda + 2\mu, \mu, \alpha > 0, 3\beta + 2\gamma, \gamma, \varepsilon > 0.$$

При выполнении этих условий потенциальная энергия упругой деформации представляет собой положительную квадратичную форму, а система уравнений (2) является гиперболической по Фридрихсу.

Для гиперболической системы корректно поставлена краевая задача с начальными данными и диссипативными граничными условиями [4; 5], выполнение которых для любых двух векторов-функций  $U$  и  $\bar{U}$  в точках границы обеспечивает выполнение неравенства

$$(\bar{U} - U)(n_1 B^1 + n_2 B^2 + n_3 B^3)(\bar{U} - U) \leq 0.$$

К диссипативным, в частности, относятся условия в напряжениях, которые формулируются в рамках модели моментной среды следующим образом:

$$\sigma_{ik} n_i = p_k, \quad m_{ik} n_i = q_k,$$

где  $n$  – вектор внешней нормали;  $p$  и  $q$  – заданные на границе векторы внешних усилий и моментов (сумми-

рование производится по повторяющимся индексам), а также условия в скоростях:  $v = v_0, \omega = \omega_0$ .

**Вычислительная технология.** Для проведения численного исследования краевых задач для системы (2) разработан параллельный вычислительный алгоритм, основанный на методе двуциклического расщепления по пространственным переменным [3]. Полученные в результате этого расщепления одномерные системы решаются с помощью явной монотонной ENO-схемы с предельной реконструкцией решения.

Алгоритм реализован в виде комплекса программ на языке Fortran-90 с использованием библиотеки MPI (*Message Passing Interface*) по технологии SPMD (*Single Program – Multiple Data*). Распараллеливание выполнено на основе блочного разбиения области решения задачи. Программный комплекс позволяет проводить расчеты распространения волн, вызванных внешними механическими воздействиями, в массиве среды, составленном из произвольного числа разнородных блоков с криволинейными границами. Внутренние границы раздела блоков считаются согласованными, что в общем случае достигается с помощью фиктивного разбиения расчетной области. На внешних границах допускается постановка основных типов краевых условий в напряжениях и скоростях, а также неотражающих условий, моделирующих беспрепятственное прохождение волн. Универсальность программ обеспечивается за счет специальной упаковки переменных, используемых на каждом из вычислительных узлов кластера, в одномерные массивы большой размерности.

Необходимые для расчета исходные данные должны представляться в виде текстовых файлов, организованных по типу реляционных таблиц. Один из таких файлов содержит механические параметры материалов в блоках, другой – условия нагружения, в третьем хранится информация о блочной структуре массива: количество слоев по переменной  $x_1$ , количество полос в слое по переменной  $x_2$  и количество блоков в полосе по переменной  $x_3$ , координаты вершин блоков, а также идентификационные номера материалов и пространственные размерности сеток. Расчетные сетки, вообще говоря, не стыкующиеся на межблочных границах, в простейшем варианте строятся с помощью кубических эрмитовых сплайнов. Склеивка решений на границах раздела выполняется специальной процедурой, в которой решение на измельченной сетке, получаемой пересечением граничных ячеек соседних блоков, находится с помощью уравнений на характеристиках, а затем переносится на исходные сетки методом осреднения.

Каждый из узлов кластера при выполнении программы-препроцессора упаковывает исходные данные в двоичный файл прямого доступа – файл вещественных чисел, в который поблочно записываются параметры материала, часть сетки, приходящаяся на этот узел, и начальные значения решения, и файл целых чисел, содержащий соответствующие им адреса (указатели) – порядковые номера первых элементов. Вещественные файлы такой же структуры в дальнейшем будут создаваться основной программой для организации контрольных точек при счете и для последующего анализа полученных результатов. Суммарный размер таких файлов может значитель-

но превышать объем оперативной памяти отдельного процессора.

Каждый процессор при старте основной программы считывает целочисленный файл и соответствующий ему вещественный файл. Далее целочисленный массив (образ целочисленного файла) редуцируется, т. е. параметры сетки и указатели принимают в нем индивидуальные значения для данного узла. Распределение расчетной области между процессорами производится по принципу равномерной загрузки: если размерность сетки какого-либо блока больше средней размерности в расчете на один узел кластера, то этот блок обслуживается несколькими узлами, и наоборот, один и тот же узел обслуживает несколько блоков, если их суммарная размерность не превосходит среднюю размерность.

Для примера рассмотрим расчетную область в виде куба для двухслойной разнородной среды (слои разделены между собой поверхностью типа гиперболического параболоида), разбитую на 8 криволинейных блоков, с грубой разностной сеткой, построенной в каждом из блоков автономно (рис. 1, а), а также выделенную четверть расчетной области (два блока) (рис. 1, б) с учетом симметрии задачи, равномерно распределенной программой-препроцессором между 72 процессорами:  $4^3 = 64$  процессора обслуживают верхний блок и  $2^3 = 8$  – нижний.

Основная программа на каждом узле кластера выполняется одни и те же вычисления, которые сводятся к взаимно согласованной поэтапной реализации метода расщепления по пространственным переменным на каждом шаге по времени. Исключение составляют процессоры, производящие помимо этого еще и склейку решений на внутренних границах. Условия склейки реализуются по следующей схеме. Процессоры, обслуживающие приграничные блоки (блоки, расположенные по обе стороны от границы раздела), передают необходимую информацию одному из процессоров, который производит расчет всей границы в целом и рассылает результаты в обратном направлении. Из соображений минимизации количества пересылок используются различные варианты разбиений расчетной области: 1D, 2D или 3D. Если

разделяющая блоки граница лежит внутри области, обслуживаемой одним процессором, то такая склейка выполняется автономно.

При решении одномерных систем уравнений в блоке используется обычная технология законтурных ячеек (рис. 2): каждый процессор сначала обменивается с соседними процессорами граничными значениями своих данных, а затем пересчитывает искомые величины в соответствии с разностной схемой и с учетом данных, полученных от соседних процессов. Обмен данными осуществляется с помощью функции `Sendrecv` библиотеки MPI. Для минимизации времени счета объем блока передаваемых данных (и, следовательно, число передач) варьируется путем совместного решения определенного количества одномерных систем.

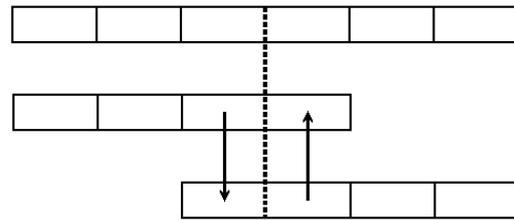


Рис. 2. Схема обмена между процессорами с законтурными ячейками

Процедуру сжатия файлов, содержащих результаты счета, выполняет программа-постпроцессор, поскольку размер таких файлов может быть очень большим и для их транспортировки по сети потребуются значительные ресурсы. Графический вывод результатов осуществляется с помощью специальных программ, предназначенных для обычного персонального компьютера.

**Алгоритм сжатия файлов.** Идея сжатия файлов числовых данных для компактного хранения и передачи решения заимствована в методе деления переменных, в соответствии с которым заданная скалярная функция  $u$ , зависящая от вектора  $x = (x_1, x_2, \dots, x_n)$ , разлагается в бесконечный ряд. Элементы этого ряда представляют собой произведения одномерных функций:

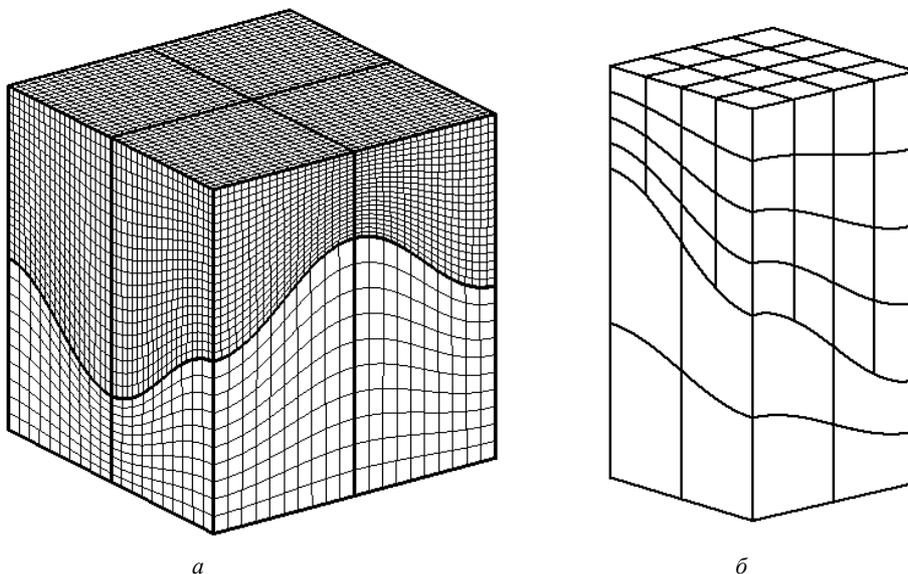


Рис. 1. Сетка и четверть расчетной области с разбиением на процессы

$$u(x) = \sum_{k=1}^{\infty} \lambda_k X^{k,1}(x_1) X^{k,2}(x_2) \dots X^{k,n}(x_n).$$

Дискретный аналог ряда с разделенными переменными выглядит следующим образом:

$$u_{i_1 i_2 \dots i_n} = \sum_{k=1}^{\infty} \lambda_k X_{i_1}^{k,1} X_{i_2}^{k,2} \dots X_{i_n}^{k,n}. \quad (3)$$

Обычно в методе разделения переменных базисные функции  $X^{k,2}, \dots, X^{k,n}$  являются собственными функциями некоторого дифференциального оператора. В данном случае они определяются из соображений наилучшего приближения  $u$ . При вычислении первого слагаемого ряда решается задача условной минимизации квадратичной функции

$$\sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \dots \sum_{i_n=1}^{M_n} (u_{i_1 i_2 \dots i_n} - \lambda X_{i_1}^1 X_{i_2}^2 \dots X_{i_n}^n)^2 \quad (4)$$

относительно переменных  $\lambda, X_{i_1}^1, X_{i_2}^2, \dots, X_{i_n}^n$ . При этом предполагается, что выполнены следующие условия нормировки:

$$\begin{aligned} \sum_{i_1=1}^{M_1} (X_{i_1}^1)^2 &= \sum_{i_2=1}^{M_2} (X_{i_2}^2)^2 = \\ &= \dots = \sum_{i_n=1}^{M_n} (X_{i_n}^n)^2 = 1. \end{aligned}$$

Точка минимума (4) удовлетворяет системе уравнений, которая может быть получена с учетом условий нормировки методом множителей Лагранжа:

$$\lambda = \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \dots \sum_{i_n=1}^{M_n} u_{i_1 i_2 \dots i_n} X_{i_1}^1 X_{i_2}^2 \dots X_{i_n}^n, \quad (5)$$

$$X_{i_k}^k =$$

$$= \frac{\sum_{i_1=1}^{M_1} \dots \sum_{i_{k-1}=1}^{M_{k-1}} \sum_{i_{k+1}=1}^{M_{k+1}} \dots \sum_{i_n=1}^{M_n} u_{i_1 i_2 \dots i_n} X_{i_1}^1 \dots X_{i_{k-1}}^{k-1} X_{i_{k+1}}^{k+1} \dots X_{i_n}^n}{\sqrt{\sum_{i_1=1}^{M_1} \left( \sum_{i_2=1}^{M_2} \dots \sum_{i_{k-1}=1}^{M_{k-1}} \sum_{i_{k+1}=1}^{M_{k+1}} \dots \sum_{i_n=1}^{M_n} u_{i_1 i_2 \dots i_n} X_{i_1}^1 \dots X_{i_{k-1}}^{k-1} X_{i_{k+1}}^{k+1} \dots X_{i_n}^n \right)^2}}. \quad (6)$$

Эта система используется при численной реализации алгоритма. Параметр  $\lambda$  определяется по формуле (5) в завершение расчета, когда базисные функции уже найдены.

Для получения базисных функций применяется один из вариантов метода последовательных приближений. Учитывая условие нормировки, начальное приближение для  $X_{i_k}^k$  принимается равным  $1/\sqrt{M_k}$ . Все последующие приближения рассчитываются по формуле (6), в которой значения  $X_{i_{k+1}}^{k+1}, \dots, X_{i_n}^n$  берутся с предыдущего шага, а в качестве  $X_{i_1}^1, \dots, X_{i_{k-1}}^{k-1}$  участвуют вновь вычисленные значения. Таким образом, реализуется полуневная процедура вычислений, аналогичная методу Зейделя для решения линейных систем алгебраических уравнений. Второе и все оставшиеся слагаемые ряда (3) находятся так же. При определении второго слагаемого дискретная функция  $u_{i_1 i_2 \dots i_n}$  заменяется погрешностью  $u_{i_1 i_2 \dots i_n} - \lambda X_{i_1}^1 X_{i_2}^2 \dots X_{i_n}^n$ , полученной после вычисления на предыдущем этапе, и т. д. Условие окончания счета формулируется в терминах относительной среднеквадратичной погрешности приближения.

Необходимо отметить, что для функции двух независимых переменных, определенной на прямоугольнике, алгоритм сжатия файлов полностью совпадает с алгорит-

мом вычисления наибольшего собственного значения и соответствующего собственного вектора симметричной неотрицательно определенной матрицы, полученной степенным методом Люстерника.

Описанный выше алгоритм сжатия и вопросы сходимости итерационного процесса и ряда (3) с разделенными переменными в двумерном случае детально исследованы в [6–8], где показано, что скорость сходимости ряда (3) определяется гладкостью аппроксимируемой функции  $u$ . Однако для  $n$  независимых переменных эти вопросы еще далеки от своего полного решения.

В качестве примера, иллюстрирующего работоспособность алгоритма сжатия, на рис. 3 приведено графическое изображение исходного файла данных, моделирующего сферическую ударную волну на сетке из  $50 \times 50 \times 50$  ячеек (слева), и результата сжатия (справа). Были вычислены 7 членов ряда. Коэффициент сжатия файла –  $1/120$ . Относительная погрешность, характеризующая потерянную информацию, равна 1,5 %.

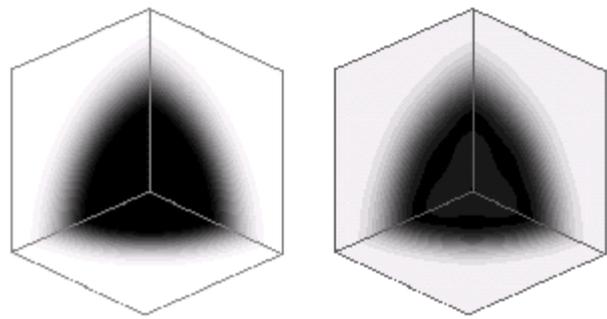


Рис. 3. Сферическая ударная волна и результат сжатия

Также приведем исходную картину и результат сжатия файла для функции

$$u(x_1, x_2, x_3) = e^{-k(r-r_0)^2/r_0^2} (r^2 = x_1^2 + x_2^2 + x_3^2),$$

описывающей локализацию решения на сфере радиуса  $r_0$  (рис. 4). Здесь безразмерный параметр локализации  $k = 20$ , а радиус сферы равен наибольшей диагонали параллелепипеда. В данном случае число членов ряда равно 12, относительная погрешность вычислений – 2,2 %, коэффициент сжатия –  $1/70$ . Число членов ряда, необходимое для достижения заданной погрешности, растет с увеличением степени локализации, т. е. коэффициента  $k$ . Таким образом, рассматриваемый алгоритм сжатия при наличии локализаций решения в весьма узких областях становится малоэффективным.

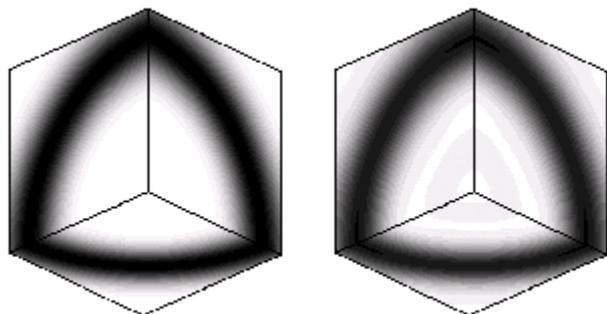


Рис. 4. Локализация решения на сфере и результат сжатия

Проведенная серия расчетов показала, что итерационный процесс на основе формул (5) и (6) является экономичным в том смысле, что требуемое для заданной точности число итераций метода последовательных приближений практически не зависит от размерности дискретной задачи.

#### Библиографический список

1. Cosserat, E. *Theorie des Corps Deformables* / E. Cosserat, F. Cosserat // *Chwolson's Trait  Physique*. 2nd ed. Paris, 1909. P. 953–1173.
2. Пальмов, В. А. Основные уравнения теории несимметричной упругости / В. А. Пальмов // *Прикл. математика и механика*. 1964. Т. 28, вып. 3. С. 401–408.
3. Садовская, О. В. Математическое моделирование в задачах механики сыпучих сред / О. В. Садовская, В. М. Садовский. М. : Физматлит, 2008.

4. Годунов, С. К. Уравнения математической физики / С. К. Годунов. М. : Наука, 1979.
5. Рихтмайер, Р. Принципы современной математической физики / Р. Рихтмайер. М. : Мир, 1982.
6. Аграновский, М. Л. Об одном разложении в гильбертовом пространстве / М. Л. Аграновский, Р. Д. Баглай // *Журн. вычисл. математики и мат. физики*. 1977. Т. 17, № 4. С. 871–878.
7. Поспелов, В. В. О приближении функций нескольких переменных произведениями функций одного переменного: препр. / В. В. Поспелов ; Ин-т прикл. математики АН СССР. М., 1978.
8. Поспелов, В. В. О погрешности приближения функции двух переменных суммами произведений функций одного переменного / В. В. Поспелов // *Журн. вычисл. математики и мат. физики*. 1978. Т. 18, № 5. С. 1307–1308.

M. P. Varygina, I. V. Kireev, O. V. Sadovskaya, V. M. Sadovsky

### SOFTWARE FOR THE ANALYSIS OF WAVE MOTION IN MOMENT CONTINUA ON PARALLEL SUPERCOMPUTERS

*For numerical analysis of dynamic problems of Cosserat elasticity theory on multiprocessor computer systems the parallel algorithms are worked out, which software implementation is fulfilled in Fortran-95 using the SPMD technology and the MPI library. The software package is provided by means of the data bulk compression with controlled information loss, permitting multiply decrease the network traffic while copying files with the results of computations from the remote cluster, and serving for the compact storage of numerical solutions in read-only memory of the computer.*

*Keywords: Cosserat continuum, elastic waves, high-performance computing.*

УДК 004.021

М. А. Селиванова, Т. А. Фомина, Е. Е. Шукшина

### АДАПТИВНАЯ СИСТЕМА ОБУЧЕНИЯ ТЕРМИНОЛОГИЧЕСКОЙ ЛЕКСИКЕ АНГЛИЙСКОГО ЯЗЫКА

*В качестве реализации подхода Л. А. Растрюгина к адаптивному обучению рассмотрена программная система адаптивного обучения терминологической лексике английского языка.*

*Ключевые слова: адаптивная система, терминологическая лексика, обучение.*

Процесс разработки любого достаточно сложного программного продукта, как правило, разделяют на этапы концептуализации, анализа, проектирования, реализации и сопровождения [1].

На этапе концептуализации выявляется сущность требований к программной системе: идентифицируются решаемые ею задачи, определяется среда, в которой она будет функционировать, необходимость соответствия стандартам, выясняются различного рода требования к пользовательскому интерфейсу, производительности и надежности создаваемой системы.

На этапе анализа определяется поведение системы с точки зрения ее пользователей: выявляются группы пользователей, формируются сценарии их взаимодействия с системой, приводящие к решению стоящих перед нею задач.

На этапе создания архитектуры для реализации системы определяется ее организация, выбираются структурные элементы, составляющие систему, и их интерфейсы, задается порядок взаимодействия элементов друг с другом, а также структурные и поведенческие элементы подсистем. На данном этапе принимаются решения, касающиеся архитектурного стиля, направляющего и определяющего всю организацию системы: ее статических и динамических элементов, их интерфейсов, кооперации и способа их объединения. На этом этапе также выбирают инструментальные средства разработки.

На этапе реализации системы происходит создание программного продукта по разработанным на предыдущих шагах спецификациям, проведение различного рода испытаний и внесение необходимых изменений в про-