

М. Белевич, А. Кокорин

ПРИМЕНЕНИЕ СИСТЕМ КОМПЬЮТЕРНОЙ АЛГЕБРЫ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

М. Belevich, A. Kokorin

APPLICATION OF THE COMPUTER ALGEBRA SYSTEMS IN MATHEMATICAL MODELING

Современное математическое моделирование немыслимо без использования компьютеров. Это, в свою очередь, вызывает развитие специального программного обеспечения помимо давно используемых средств: различного уровня алгоритмических языков и соответствующих компиляторов или интерпретаторов. В работе вкратце рассматриваются основные тенденции развития современных программных комплексов, ориентированных на математическое моделирование. В качестве иллюстрации приводится схема модели волнового пограничного слоя, реализованной в рамках системы компьютерной алгебры Mathematica.

Ключевые слова: системы компьютерной алгебры, системы численного анализа.

Present-day mathematical modeling is inconceivable without computers. This, in its turn, causes the development of the special software besides different algorithmic languages and corresponding compilers. In this paper, the main tendencies of the development of the contemporary mathematical modeling packages are considered. All this is illustrated with the scheme of the wave boundary layer model which has been written in the framework of the computer algebra system Mathematica.

Keywords: computer algebra system, numerical analysis software.

Введение

Процесс математического моделирования явлений природы для наглядности можно разделить на несколько этапов:

1. Постановка задачи.
2. Разработка и реализация алгоритма.
3. Решение математической задачи.
4. Обработка и интерпретация результатов.

Первый этап включает словесную формулировку проблемы, выбор адекватной математической модели, постановку граничных условий, оценку масштабов явления и численного значения параметров. На втором этапе производится выбор подходящего метода решения сформулированной математической задачи и способа его реализации. На третьем этапе проводятся основные вычисления, дающие численное решение поставленной задачи. И, наконец, на последнем этапе данные вычислений обрабатываются, визуализируются и интерпретируются в исходных терминах.

С появлением в середине прошлого века компьютеров на них поначалу, легла основная вычислительная работа (третий этап). Все остальное делалось вручную на листе бумаги, миллиметровки или кальки. В настоящее время, осо-

бенно с появлением персональных компьютеров, ситуация изменилась. Разработаны и продолжают развиваться многочисленные программы, позволяющие использовать компьютер практически на любом шаге исследования. Вместе с тем, наиболее востребованными в научной среде многие годы были и продолжают оставаться средства разработки компьютерных программ (компиляторы и текстовые редакторы), программы визуализации результатов расчетов или измерений и текстовые процессоры.

Почти на каждом из этапов требуется давать оценки и осуществлять выбор. В результате кристаллизуется формулировка задачи, оттачивается ее постановка, отбрасываются неприемлемые по тем или иным причинам методы решения либо разрабатывается новый и отрабатываются способы представления результатов вычислений и их интерпретация.

На каждом таком шаге решения крупной задачи исследователь сталкивается с необходимостью выполнения ряда сравнительно мелких подзадач, для чего могут привлекаться специальные компьютерные программы. Эти программы должны

- обеспечивать возможность быстрого проведения оценочных вычислений, т.е. играть роль калькулятора;
- уметь выполнять аналитические преобразования выражений (упрощать, дифференцировать, интегрировать и пр.);
- обладать средствами для построения и разработки алгоритмов (иметь встроенный язык программирования);
- давать возможность проведения численных расчетов;
- располагать развитыми графическими средствами.

Подобные программы разрабатываются, начиная с 60-х годов прошлого века и в настоящее время представлены не одним десятком. Среди программ общего назначения исторически сложились два основных направления разработок. В одном основной упор делается на аналитические преобразования, арифметику произвольной точности и т.п. Программы такого типа обычно называются системами компьютерной алгебры (Computer algebra systems или, сокращенно, CAS). Другое направление связано преимущественно с вычислениями с плавающей точкой, и потому относятся к программам численного анализа. Сведения о CAS и системах численного анализа, а также перечни имеющихся программных комплексов и результаты их сопоставления можно найти в английской версии Wikipedia в статьях “Computer algebra system”, “Comparison of computer algebra systems”, “List of numerical analysis software”, “Comparison of numerical analysis software” и др.

Особенности современных CAS

К настоящему времени разделение программ на указанные два направления стало достаточно условным, поскольку налицо явная тенденция к взаимопроникновению: системы компьютерной алгебры предоставляют средства для про-

ведения расчетов с плавающей точкой, а системы программ численного анализа обзаводятся подсистемами, позволяющими производить аналитические преобразования. Причем, последние обычно являются не самостоятельными разработками, а так или иначе связаны с независимо создававшимися CAS. Среди универсальных программ такого рода наиболее известными и коммерчески успешными стали следующие продукты.

1. Системы численного анализа
 - а. Mathcad, Parametric Technology Corp. [Кириянов, 2006], www.ptc.com/products/mathcad/
 - б. MatLab, The MathWorks Inc. [Алексеев, Чеснокова, 2005], www.mathworks.com/
2. Системы компьютерной алгебры
 - а. Maple, Waterloo Maple Inc. [Васильев, 2003], www.maplesoft.com/
 - б. *Mathematica*, Wolfram Research Inc. [Дьяконов, 2008; Шмидский, 2004], www.wolfram.com/

Все они прекрасно документированы и снабжены многочисленными примерами использования. Существует развитая техническая поддержка клиентов, внушительная библиотека обучающей литературы, а также огромное количество доступных написанных пользователями программ, работающих в рамках соответствующих систем.

Программа Mathcad привлекает, прежде всего, пользовательским интерфейсом. Он построен по принципу листа бумаги, на котором исследователь записывает свои формулы, их преобразования, несложные расчеты, графики и комментарии. Все это может располагаться почти в произвольном порядке, а формулы представляются в привычном виде, близком к типографскому. Такие особенности программы делают изучение ее и работу с ней простым и интуитивно понятным делом. С ее помощью можно быстро провести сравнительно несложные вычисления и взглянуть на таблицы получаемых значений. Есть возможность тут же рядом построить 2D или 3D графики получаемых зависимостей и оставить необходимые комментарии. Вычисления проводятся с плавающей точкой и, кроме того, можно провести несложные аналитические преобразования. Например, можно разложить некоторую функцию в ряд Тейлора, проинтегрировать ее, вычислить интеграл, упростить выражение и т.п.

Система MatLab строилась, прежде всего, как средство для интенсивных численных расчетов, как интерфейс к широко известным библиотекам программ численного анализа типа LAPACK. В настоящее время она располагает большим набором встроенных функций, развитым языком сверхвысокого уровня, мощной подсистемой визуализации, и т.д., что позволяет решать с ее помощью весьма сложные задачи.

На официальном сайте компании Waterloo Maple Inc. написано, что Maple является ведущим вычислительным средством в мире. В свою очередь, на сайте компании Wolfram Research Inc. указано, что *Mathematica* есть наиболее все-

объемлющее вычислительное приложение в мире. В некотором смысле все это так и есть. Обе системы являются прямыми конкурентами. В первую очередь они разрабатывались как программы для выполнения аналитических преобразований и проведения расчетов с произвольной точностью. Они обладают впечатляющими наборами встроенных функций, развитыми графическими возможностями и средствами выполнения расчетов с плавающей точкой. Каждая располагает интерпретатором языка сверхвысокого уровня, с помощью которого можно эффективно реализовывать сложные алгоритмы. Назвать лучшую не представляется возможным. Каждая система имеет многочисленных приверженцев. При этом нужно иметь в виду, что эти системы не тождественны друг другу: то, что отсутствует, либо неудачно реализовано в одной, может присутствовать или быть более эффективно реализовано в другой. Обе программы прекрасно документированы. Существует большое число книг, описывающих работу с ними. Те, что упомянуты в списке литературы, – лишь малая толика.

Наряду с коммерческими программами существует немало свободно распространяемых. Общая тенденция к доступности программного обеспечения, а в ряде случаев желание вдохнуть новую жизнь в не выдерживающие конкуренции коммерческие продукты приводит к тому, что появляется все большее число программ с лицензией GPL (General Public License) или ей подобными. Ниже перечислены некоторые из таких программ.

1. Системы численного анализа
 - а. SMath Studio, ru.smath.info/forum/
 - б. SciLab [Алексеев et al., 2008], www.scilab.org/
2. Системы компьютерной алгебры
 - а. Maxima [Ильина, Силаев, 2007], maxima.sourceforge.net/
 - б. Reduce [Климов, Руденко, 1989], www.reduce-algebra.com/

По своим возможностям эти программы близки к упомянутым коммерческим продуктам и позволяют решать аналогичный круг задач.

Численные и аналитические способности этих программ неоднократно тестировались, и результаты можно найти, например, по адресу www.scientificweb.de/ncrunch/. Хороший обзор указанных программ находится на сайте tatalmas.by.ru/kursov1/.

Варианты использования CAS в моделировании

Существуют простые и очевидные варианты использования систем численного анализа. Во-первых, как уже говорилось, они могут быть использованы как удобные средства для выполнения всевозможных несложных оценочных вычислений и их визуализации. К таким средствам, в первую очередь, относится программа Mathcad или ее бесплатный аналог SMath Studio. Они требуют минимума времени на освоение в объеме, достаточном для проведения осмысленных расчетов и простейшего моделирования.

Другой вариант применения обсуждаемых систем – использование их в качестве языков сверхвысокого уровня для реализации алгоритмов решения сис-

тем уравнений моделей и среды для работы написанных на этих языках программ и программных комплексов. Наиболее популярными здесь являются система MatLab и ей подобные (SciLab, Octave и др.). Системы численного анализа можно рассматривать в одном ряду с такими средствами программирования, как ассемблеры и алгоритмические языки типа fortran, С и т.п. Их задача – осуществить связь между компьютером и исследователем, освобождая последнего от действий, мало связанных с предметом исследования.

С появлением ассемблеров и, далее, языков высокого уровня исследователь перестал тратить усилия на рутинные действия по распределению памяти, управлению периферийными устройствами, низкоуровневой организации вычислений и мог сосредоточиться на разработке алгоритма решения задачи. Системы численного анализа дают возможность сделать следующий шаг в этом направлении. Они предоставляют исследователю наборы высококачественных реализаций эффективных алгоритмов и удобные средства для описания разработанных им численных моделей, позволяя сосредоточиться на физике явления. В этом смысле они являются языками сверхвысокого уровня.

Наконец, наиболее интересным, на наш взгляд, представляется использование в моделировании развитых систем компьютерной алгебры, таких, как Maple, *Mathematica*, Reduce, Maxima и т.п. Уступая системам численного анализа в быстрой работе при выполнении вычислений с плавающей точкой, они, однако, позволяют также проводить вычисления с произвольной точностью и, кроме того, являются мощными средствами, выполняющими всевозможные аналитические действия (преобразование выражений, дифференцирование, интегрирование функций, разложение их в ряды, решение уравнений, неравенств и т.д.). Это позволяет использовать их не только при проведении вычислений, но фактически на всех этапах моделирования от формулировки и постановки задачи, выбора алгоритма решения до проведения численных расчетов, обработки и представления результатов, а в ряде случаев, и создания итоговых документов (статьи, отчета и пр.).

Имеющиеся здесь возможности мы далее иллюстрируем на примере построения модели волнового пограничного слоя с помощью CAS *Mathematica*. Поскольку у читателя не предполагается каких-либо знаний языка *Mathematica*, мы не публикуем полный листинг программы, реализующей модель, а как правило, ограничиваемся комментариями (шрифт Times New Roman), изредка вставляя в иллюстративных целях фрагменты программы и отклик системы (шрифт Courier New).

Пример использования CAS: Спектральная модель волнового пограничного слоя (ВПС), реализованная в системе Mathematica

Рассматривается задача обтекания двумерного волнового поля воздушным потоком, сонаправленным с фазовой скоростью волны.

1. Вывод уравнений ВПС в отслеживающих координатах

Исходная система уравнений

Система уравнений турбулентной жидкости в бескоординатной форме такова

$$\operatorname{div}(\rho \hat{v}) = 0,$$

$$\rho d_t \hat{v} = g^{-1}(\nabla \pi) + 2 \operatorname{div}(\mu_T D_{\text{turb}}).$$

Здесь ρ – плотность массы, $\hat{v} = (1, v^1, v^2, v^3)$ – 4-вектор скорости точки среды, g – метрический тензор, $g^{-1}(\nabla \pi)$ – градиент давления, $2 \mu_T D_{\text{turb}}$ – тензор напряжений Рейнольдса.

Гипотезы: волна двумерна, жидкость несжимаема и $\rho = \text{const}$.

Пространственная *область определения функций:* цилиндрическая область W , ограниченная снизу поверхностью волны $\eta(t, x)$.

В пространственно-временной области с координатами $(x^0 = t, x^1 = x, x^2 = z)$ выбирается метрика Минковского, а система пространственных координат – декартовой. Начало координат помещается на невозмущенной поверхности раздела «вода–воздух». Верхняя граница области W (высота ВПС) фиксируется на высоте h .

В этом случае координатная форма уравнений задачи такова (здесь p – нормированное на плотность отклонение давления от гидростатического,

$$v_T = \frac{\mu_T}{\rho}$$

а

$$(v^k)_{;k} = 0, \quad v^k = d_t x^k, \quad k = 1, 2,$$

$$v^\alpha (v^k)_{;\alpha} = -\delta^{\alpha k} (p)_{;\alpha} + (v_T (v^k)_{;\alpha})_{;\beta} \delta^{\beta \alpha}, \quad \alpha, \beta = 0, 1, 2.$$

Спрявление ВПС. Введение координат, базисов и метрик

Заменой координат $\zeta = h \frac{z - \eta}{h - \eta}, z = \eta + \left(1 - \frac{\eta}{h}\right) \zeta$ область W отображаем на прямоугольную область P с координатами $(\xi^1 = x, \xi^2 = \zeta)$, компонентами скорости $u^\alpha = d_t \xi^\alpha$, ($u^0 = 1, u^1 = u, u^3 = w$) и метрическим тензором G .

Система уравнений записывается так:

$$(u^k)_{;k} = 0, \quad u^k = d_t \xi^k, \quad k = 1, 2,$$

$$u^\alpha (u^k)_{;\alpha} = -G^{\alpha k} (p)_{;\alpha} + (v_T (u^k)_{;\alpha})_{;\beta} G^{\beta \alpha}, \quad \alpha, \beta = 0, 1, 2.$$

Подключение пакета ttc, реализующего работу с тензорами

<<ttc.m

Tools of Tensor Calculus 4.3 .1 version : June, 6, 2005

Session started on Jan, 25, 2011 at 14 h 41 min 29.5 s

Описываем координаты:

ctn – декартовы коорд. в W области, **flw** – отслеживающие координаты в P -области

InputCoordinates [ctn, {t, x, z}];

InputCoordinates [flw, {t, x, ζ}];

Задаем преобразование координат **ctn** ↔ **flw**:

InputCoordinateChange [ctn, flw, {t→t, x→x, z→η[t, x] + (1-η[t, x])/h}];

InputCoordinateChange [flw, ctn, {t→t, x→x, ζ→h (z-η[t, x]) / (h-η[t, x])}];

Определяем координатные базисные векторы:

e – декартов базис в W – области, **ε** – базис в P -области

e = ZZ[ctn]; ε = ZZ[flw];

Записываем разложение скорости по базисным векторам:

$v=(l, v1, v3) - \mathbf{ctn}, V=(l, u, w) - \mathbf{flw}$

v = e[-1] + v1[t, x, z] e[-2] + v3[t, x, z] e[-3];

V = ε[-1] + u[t, x, ζ] ε[-2] + w[t, x, ζ] ε[-3];

Задаем метрику в W области: **g** – индефинитная метрика Минковского

g = -e[t]*.e[t]+e[x]*.e[x]+e[z]*.e[z];

InputMetric[mink,ctn,g];

Метрика в P -области (в пределе $h \rightarrow \infty$)

G = Limit[Change [ctn, flw] [g], h->∞]

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 + \eta_x^2 & \eta_x \\ 0 & \eta_x & 1 \end{pmatrix}$$

Тензор **G** считаем метрическим в P -области

InputMetric [mink, flw, G];

Модельные уравнения

1. Уравнение неразрывности

eqn0 = Limit[V[i, .; -i] // Index [mink], h->∞]

$w_\xi + u_x$

2. Уравнение движения

преобразование давления

(**Pf** – в отслеживающих координатах, **Pc** – в декартовых координатах)

Pf = p[t, x, ζ]

Pc = Change [flw, ctn] [Pf]

$p[t, x, \xi]$

$$p \left[t, x, \frac{h(z-\eta)}{h-\eta} \right]$$

градиент давления в P -области

DtPf = AbsoluteD [mink] [Pf];

$DtPfi = DtPf[i] // Index[mink]$

$$\begin{pmatrix} p_x - \eta_x p_\xi \\ p_\xi (1 + \eta_x^2) - p_x \eta_x \end{pmatrix}$$

коэффициент турбулентной вязкости

(vTf – в отслеживающих координатах, vTc – в декартовых координатах)

$vTc = Change[flw, ctn][vTf[\zeta]]$

$$vTf \left[-\frac{h(-z + \eta)}{h - \eta} \right]$$

градиент коэффициента турбулентной вязкости

$DtvTf = Limit[AbsoluteD[mink][vTf[\zeta]], h \rightarrow \infty]$

$$\begin{pmatrix} -\eta_x vTf_\xi \\ (1 + \eta_x^2) vTf_\xi \end{pmatrix}$$

уравнение движения

$EQN[b_] := V1[a] V[b, .; -a] + DtPf[b] -$

$DtvTf[a] V[b, .; -a] - vTf[\zeta] V[b, .; a, .; -a] // Index[mink]$

первая компонента уравнения движения

$eqn1 = EQN[1]$

$$u_t + u u_x + w u_\xi + p_x - p_\xi \eta_x + (-u_\xi + u_x \eta_x - u_\xi \eta_x^2) vTf_\xi + \\ (-u_{x2} + 2 u_x \xi \eta_x + u_{\xi 2} (-1 - \eta_x^2) + u_\xi \eta_{x2}) vTf[\xi]$$

Вторая компонента уравнения движения

$eqn2 = EQN[2]$

$$w_t + u w_x + w w_\xi + p_\xi (1 + \eta_x^2) - p_x \eta_x + u^2 \eta_{x2} + (-w_\xi (1 + \eta_x^2) + \eta_x (w_x + u \eta_{x2})) vTf_\xi + \\ (-w_{x2} - w_{\xi 2} (1 + \eta_x^2) + (-2 u_x + w_\xi) \eta_{x2} + \eta_x (2 w_x \xi + 2 u_\xi \eta_{x2}) - u \eta_{x3}) vTf[\xi]$$

3. Уравнения Пуассона для давления: дифференцируем 1-е по x , а 2-е – по ξ

$DEqn2 = D[eqn2, x];$

$DEqn3 = D[eqn3, \xi];$

Суммируем и получаем уравнение для давления

$eqn23 = Simplify[DEqn2 + DEqn3]$

$$p_{x2} + p_{\xi 2} (1 + \eta_x^2) + 2 u_\xi w_x - u_x w_\xi + w_\xi^2 - 2 p_x \xi \eta_x + (-p_\xi + 2 u u_\xi) \eta_{x2} + \\ (-w_{x2} - w_{\xi 2} (1 + \eta_x^2) + 2 w_\xi \eta_{x2} + \eta_x (2 w_x \xi + u_\xi \eta_{x2}) - u \eta_{x3}) vTf_\xi + \\ (-w_\xi + w_x \eta_x - w_\xi \eta_x^2 + u \eta_x \eta_{x2}) vTf_{\xi 2}$$

Запись промежуточных результатов

$equ0 \gg tmpEqns/10_equ0.txt$

$equ1 \gg tmpEqns/11_equ1.txt$

$equ2 \gg tmpEqns/12_equ2.txt$

$equ23 \gg tmpEqns/13_equ3.txt$

2. Спектральные уравнения ВПС

Постановка задачи. Граничные и начальные условия

Решается система уравнений турбулентной жидкости. Область определения функций – цилиндр $[0, L] \times [0, h]$. Система координат – «отслеживающая» (x, ζ) ; преобразования декартовых координат $(x, z) \leftrightarrow (x, \zeta)$ таковы: $\zeta = h \frac{z - \eta}{h - \eta}$, $z = \eta + \left(1 - \frac{\eta}{h}\right) \zeta$; размерности z и ζ равны. Преобразование координат выпрямляет нижнюю возмужанную границу области.

Граничные условия.

Периодические по горизонтали.

Верхняя граница:

$$u(t, x, h) = u_{\max} - \text{скорость потока вне ВПС},$$

$$w(t, x, h) = 0,$$

$$p(t, x, h) = 0.$$

Нижняя граница:

$$u(t, x, 0) = u_0(t, x) - \text{скорость движения поверхности},$$

$$w(t, x, 0) = 0,$$

$$p(t, x, 0) = 0.$$

Начальные условия:

$$u(0, x, \zeta) = u_0 + \frac{\zeta}{h} (u_{\max} - u_0) - \text{линейный профиль скорости от } u_0 \text{ до } u_{\max}.$$

$$w(0, x, \zeta) = 0,$$

$$p(0, x, \zeta) = 0.$$

Задача решается проекционным методом взвешенных невязок: решение представляется в виде усеченного ряда Фурье с последующей минимизацией проекции невязки на пространстве, натянутом на те же базисные функции. В качестве базисных функций берутся собственные функции оператора Лапласа:

$$e_{k,s}(t, x, \zeta) = e^x(k, t, x) \cdot e^\zeta(s, \zeta) = \left(\frac{1}{\sqrt{\pi}} \sin(t \omega_k + x k) \right) \cdot \left(\sqrt{\frac{2}{\pi}} \sin(s \zeta) \right).$$

Базисные функции занумерованы целыми положительными значениями волновых чисел k и s .

Для учета ненулевых граничных условий в базис включены еще две функции $e^{\zeta_{01}} = 1$, $e^{\zeta_{02}} = \zeta$.

Ряды Фурье для искоемых функций:

$$u(t, x, \zeta) = \left(u_0(t, x) + (u_{\max} - u_0(t, x)) \frac{\zeta}{h} \right) + \sum_{k,s} U^{k,s} e_k(t, x) e_s(\zeta);$$

$$w(t, x, \zeta) = \sum_k \left(\sum_s W^{k,s} e_s(\zeta) \right) e_k(t, x);$$

$$p(t, x, \zeta) = \sum_k \left(\sum_s P^{k,s} e_s(\zeta) \right) e_k(t, x);$$

при $u_0(t, x) = \sum_k U_0^k e_k(t, x)$ и $\eta(t, x) = \sum_k Z_k e_k(t, x)$;

Представление решений рядами Фурье, подстановка и минимизация невязки

Правила подстановки для разложения решений по базисным функциям и проектирование

$$\text{ruleA} = \left\{ \eta[t, x] \rightarrow \sum_{k=1}^{km} Z_k \text{ex}_k[t, x], \right.$$

$$u[t, x, \zeta] \rightarrow u_{\max} \frac{\zeta}{h} + \sum_{k=1}^{km} \left(\left(1 - \frac{\zeta}{h} \right) Z\omega_k + \sum_{s=1}^{sm} U_{k,s}[t] e_{\zeta_s}[\zeta] \right) \text{ex}_k[t, x],$$

$$w[t, x, \zeta] \rightarrow \sum_{k=1}^{km} \left(\sum_{s=1}^{sm} W_{k,s}[t] \text{ex}_k[t, x] e_{\zeta_s}[\zeta] \right),$$

$$p[t, x, \zeta] \rightarrow \sum_{k=1}^{km} \left(\sum_{s=1}^{sm} \Pi_{k,s}[t] e_{\zeta_s}[\zeta] \right) \text{ex}_k[t, x]$$

};

Читаем выведенные уравнения

`equ1 = ToExpression["<<" <> "tmpEqns/11_equ1.txt"]`

`equ2 = ToExpression["<<" <> "tmpEqns/12_equ2.txt"]`

`equ3 = ToExpression["<<" <> "tmpEqns/13_equ1.txt"]`

Подстановка разложений решения по базисным функциям и проектирование

`heeq1 = equ1 /. ruleA;`

$$\text{heqn1} = \int_0^h \int_0^L (\text{heeq1} e_{\zeta_s}[\zeta] \text{ex}_k[t, x]) dx d\zeta;$$

`heeq2 = equ2 /. ruleA;`

$$\text{heqn2} = \int_0^h \int_0^L (\text{heeq2} e_{\zeta_s}[\zeta] \text{ex}_k[t, x]) dx d\zeta;$$

`heeq3 = equ3 /. ruleA;`

$$\text{heqn3} = \int_0^h \int_0^L (\text{heeq3} e_{\zeta_s}[\zeta] \text{ex}_k[t, x]) dx d\zeta;$$

Формирование списка интегралов от комбинаций базисных функций

`Uint1 = Union[Flatten[Apply[List, (List@@heqn1), {1}]]];`

`Uint2 = Union[Flatten[Apply[List, (List@@heqn2), {1}]]];`

`Uint3 = Union[Flatten[Apply[List, (List@@heqn3), {1}]]];`

```
intlist = Select[UnsortedUnion[Join[Uint1, Uint2, Uint3]],
  Head[#] = Integrate &];
```

Формирование имен интегралов и функций для их вычисления

Число различных интегралов

```
nkint
```

```
42
```

Список интегралов и правил подстановки.

```
ColumnForm[Take[mint, nkint]]
```

$$\int_0^{2\pi} \text{ex}[k, t, x] dx \rightarrow \text{iex}[k]$$

$$\int_0^{2\pi} \text{ex}[k, t, x] \text{ex}[k1, t, x] dx \rightarrow \text{iexex}[k, k1]$$

(...)

$$\int_0^{\pi} e_{\xi}[s, \xi] e_{\xi}^{(0,1)}[s1, \xi] e_{\xi}^{(0,1)}[s2, \xi] d\xi \rightarrow \text{ie}_{\xi} e_{\xi} dz1 e_{\xi} dz1[s, s1, s2]$$

$$\int_0^{\pi} e_{\xi}[s, \xi] e_{\xi}^{(0,2)}[s1, \xi] d\xi \rightarrow \text{ie}_{\xi} e_{\xi} dz2[s, s1]$$

Замена интегралов вызовами функций

```
equationU = Simplify[heqn1 /. nint];
```

```
equationW = Simplify[heqn2 /. nint];
```

```
equationP = Simplify[heqn3 /. nint];
```

Вычисление матрицы системы уравнений и вектора правых частей

Equation #1: $B1_{m,n}^{k,s} d_t U_{k,s} + C1_{m,n} = 0$

```
C1 = equationU /. {dU[k_, s_] -> 0};
```

```
B1 = (equationU - C1) /. {dU[k_, s_] -> 1};
```

Equation #2: $B2_{m,n}^{k,s} d_t W_{k,s} + C2_{m,n} = 0$

```
C2 = equationW /. {dW[k_, s_] -> 0};
```

```
B2 = (equationW - C2) /. {dW[k_, s_] -> 1};
```

Equation #3: $B3_{m,n}^{k,s} \Pi_{k,s} + C3_{m,n} = 0;$

```
C3 = equationP /. {\Pi[k_, s_] -> 0}
```

$$\begin{aligned}
 & \sum_{k1=1}^{km} \left(\sum_{k3=1}^{km} \frac{1}{h^2} (2 U_{max} i_{exexexdx2}[k, k1, k3] (h i_{\zeta} e_{\zeta}[s] - i_{\zeta} e_{\zeta}[s]) Z[k3] Z\omega[k1]) + \right. \\
 & \quad \sum_{k2=1}^{km} \left(\sum_{s2=1}^{sm} \left(\sum_{k3=1}^{km} \frac{1}{h} (2 i_{exexexexdx2}[k, k1, k2, k3] \right. \right. \\
 & \quad \quad (h i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2] - i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2]) U[k2, s2] \\
 & \quad \quad Z[k3] Z\omega[k1]) + \\
 & \quad \quad \left. \left. \frac{1}{h} ((-2 i_{exexexdx1}[k, k1, k2] i_{\zeta} e_{\zeta}[s, s2] + \right. \right. \\
 & \quad \quad i_{exexexdx1}[k, k2, k1] (-h i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2] + \\
 & \quad \quad \left. \left. i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2])) W[k2, s2] Z\omega[k1]) \right) \right) + \\
 & \quad \sum_{s1=1}^{sm} \left(\sum_{k2=1}^{km} \left(\sum_{s2=1}^{sm} \left(\sum_{k3=1}^{km} 2 i_{exexexexdx2}[k, k1, k2, k3] i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s1, s2] \right. \right. \right. \\
 & \quad U[k1, s1] U[k2, s2] Z[k3] + \\
 & \quad (-i_{exexexdx1}[k, k2, k1] i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s1, s2] U[k1, s1] + \\
 & \quad 2 i_{exexexdx1}[k, k1, k2] i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2, s1] U[k1, s1] + \\
 & \quad \left. \left. i_{exexex}[k, k1, k2] i_{\zeta} e_{\zeta} dz1 e_{\zeta} dz1[s, s1, s2] W[k1, s1]) \right. \right. \\
 & \quad \left. \left. \left. W[k2, s2] \right) \right) \right) + \\
 & \quad \sum_{k2=1}^{km} \left(\sum_{s2=1}^{sm} \left(\sum_{k3=1}^{km} \frac{1}{h} (2 U_{max} i_{exexexdx2}[k, k2, k3] i_{\zeta} e_{\zeta} e_{\zeta} dz1[s, s2] \right. \right. \\
 & \quad U[k2, s2] Z[k3]) + \\
 & \quad \left. \left. \frac{2 U_{max} i_{exexdx1}[k, k2] i_{\zeta} e_{\zeta}[s, s2] W[k2, s2]}{h} \right) \right) + \\
 & \quad \sum_{k3=1}^{km} \frac{2 U_{max}^2 i_{exexdx2}[k, k3] i_{\zeta} e_{\zeta}[s] Z[k3]}{h^2}
 \end{aligned}$$

B3 = (equationP - C3) /. {Pi[k_, s_] -> 1}

$$\begin{aligned}
 & i_{exexdx2}[k, k1] i_{\zeta} e_{\zeta}[s, s1] + i_{\zeta} e_{\zeta} dz2[s, s1] \\
 & \left(i_{exex}[k, k1] + \sum_{k3=1}^{km} \sum_{k4=1}^{km} i_{exexexdx1} e_{\zeta} dx1[k, k1, k3, k4] Z[k3] Z[k4] \right) - \\
 & i_{\zeta} e_{\zeta} dz1[s, s1] \\
 & \left(\sum_{k2=1}^{km} 2 i_{exexdx1} e_{\zeta} dx1[k, k1, k2] Z[k2] + \sum_{k3=1}^{km} i_{exexexdx2}[k, k1, k3] Z[k3] \right)
 \end{aligned}$$

Вспомогательные вычисления

Вычисление интегралов, вывод и тестирование соответствующих функций

Вывод функций, вычисляющих компоненты матрицы СЛАУ и вектора правой части

3. Пример расчета эволюции ВПС

Начальные установки. Модуль вычисления давления

```
pressure := Module[{} ,
  Do[ (* s, k *)
    c3[[sm (k - 1) + s]] = C3[k, s];
    Do[
      b3[[sm (k - 1) + s, sm (k1 - 1) + s1]] = B3[k, s, k1, s1]
      , {k1, km}, {s1, sm}]
      , {k, km}, {s, sm}];
    bai = Inverse[b3];
    Пn = -bai.c3;
    Do[П[[k, s]] = Пn[[sm (k - 1) + s]], {k, km}, {s, sm}];
  ];
```

Основной цикл

```
tbeg = TimeUsed[];
While[(t = t + Δt) < tmax,
  (***** t *)
  nt = nt + 1;
  tpbeg = TimeUsed[];
  pressure;
  tpend = TimeUsed[];
  (** VELOCITY **)
  tsbeg = TimeUsed[];
  Do[(* s, k *)
    c1[[sm (k - 1) + s]] = C1[k, s]; c2[[sm (k - 1) + s]] = C2[k, s];
    Do[(* s1, k1 *)
      b[[sm (k - 1) + s, sm (k1 - 1) + s1]] = B1[k, s, k1, s1],
      {k1, km}, {s1, sm}],
      {k, km}, {s, sm}];
    bai = Inverse[b];
    c1ai = bai.c1; c2ai = bai.c2;
    (** evolution eqns **)
    Un = Un - Δt c1ai;
    Wn = Wn - Δt c2ai;
    Do[U[[k, s]] = Un[[sm (k - 1) + s]]; W[[k, s]] = Wn[[sm (k - 1) + s]],
      {k, km}, {s, sm}],
    tsend = TimeUsed[];
    (** NUMOUT **)
    Numout[nt, nstep];
    (** GRAPHOUT **)
    Graphout[nt, gstep];
  ]
  (* t *)
```


Fourier coeffs:

$$U = \begin{pmatrix} -0.79551 & -0.112865 & -0.0300304 \\ -0.0688162 & -0.0423719 & -0.0200821 \\ 0.0214692 & 0.023523 & 0.014227 \\ -0.000542564 & 0.00176012 & 0.00237197 \end{pmatrix}$$

$$W = \begin{pmatrix} 1.62866 & 0.199155 & 0.115328 \\ 0.489337 & 0.0901955 & 0.0854025 \\ -0.0739176 & -0.0476227 & -0.029397 \\ 0.00587429 & 0.000708963 & -0.00363702 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} -0.088928 & -0.0538339 & -0.0250818 \\ -0.0529646 & 0.0174237 & -0.0139099 \\ 0.0146914 & 0.0188532 & 0.0156644 \\ 0.00246282 & 0.00235534 & 0.00288691 \end{pmatrix}$$

Fields:

$$u = \begin{pmatrix} 0.631864 & 0.394679 & 0.255115 & 0.510071 \\ 0.66052 & 0.239479 & -0.0031851 & 0.450226 \\ 0.618958 & 0.121567 & -0.200542 & 0.35692 \\ 0.435602 & 0.0437173 & -0.25272 & 0.217673 \\ 0.093223 & -0.0195218 & -0.122016 & 0.0483147 \end{pmatrix}$$

$$w = \begin{pmatrix} -0.353054 & 0.0841906 & 0.399388 & -0.129823 \\ -0.557596 & 0.142025 & 0.612658 & -0.204631 \\ -0.64852 & 0.139704 & 0.720376 & -0.219965 \\ -0.500172 & 0.0804358 & 0.57368 & -0.154633 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$p = \begin{pmatrix} 0.00512862 & 0.00359137 & -0.016035 & 0.00529851 \\ 0.0124631 & -0.00355804 & -0.0205999 & 0.0124579 \\ 0.0401095 & -0.00453408 & -0.0463483 & 0.00867718 \\ 0.0498614 & 0.0020121 & -0.0576968 & -0.000818738 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

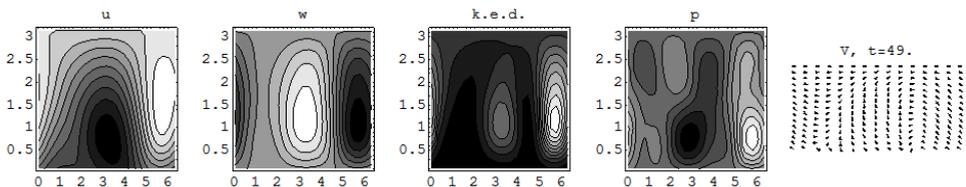
massnew=2.81404 × 10⁻¹⁶

enrgnew=0.152913

rate of change: mass = 3.46244 × 10⁻¹⁵ energy = 0.170448

mean: u = 0.28 w = -2.24978 × 10⁻¹⁷ p = 2.10917 × 10⁻¹⁷

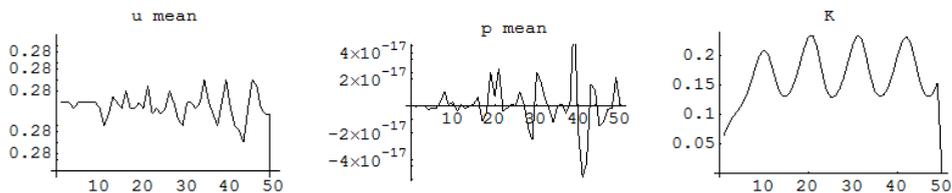
-----> t = 49. nt = 490 no = 49



Эпилог (эволюция средних величин:

горизонтальной скорости, отклонения давления, кинетической энергии)

```
plmu = ListPlot[mul, PlotLabel → "u mean", PlotJoined → True,  
  DisplayFunction → Identity];  
plmp = ListPlot[mpr, PlotLabel → "p mean", PlotJoined → True,  
  DisplayFunction → Identity];  
plme = ListPlot[men, PlotLabel → "K", PlotJoined → True,  
  DisplayFunction → Identity];  
Show[GraphicsArray[{plmu, plmp, plme}]]
```



```
tend=TimeUsed[];
```

```
Print["Total time per run:",tend-tbeg];
```

```
Total time per run:3199.83
```



Created with Wolfram *Mathematica* 8.0

Литература

1. Алексеев Е.Р., Чеснокова О.В., Рудченко Е.А. Scilab: Решение инженерных и математических задач. БИНОМ. Лаборатория знаний. – М., 2008.
2. Алексеев Е.Р., Чеснокова О.В. MATLAB 7. Самоучитель. – М.: Пресс, 2005.
3. Васильев А.Н. Maple 8. Самоучитель. – М.: Диалектика, 2003.
4. Дьяконов В.П. Mathematica 5.1/5.2/6. Программирование и математические вычисления. – М.: ДМК-Пресс, 2008.
5. Ильина В.А., Силаев П.К. Система аналитических вычислений МАХИМА для физиков-теоретиков. – М.: МГУ им. М.В. Ломоносова, 2007.
6. Кирьянов Д.В. Самоучитель Mathcad 13. – СПб.: БХВ-Петербург, 2006.
7. Климов Д.М., Руденко В.М. Методы компьютерной алгебры в задачах механики. – М.: Наука, 1989.
8. Шмидский Я.К. Mathematica 5. Самоучитель. Система символьных, графических и численных вычислений. – М.: Диалектика, 2004.