

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ НЕЛИНЕЙНЫХ ОБОБЩЕННО-МЕХАНИЧЕСКИХ СИСТЕМ В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MAPLE

Аннотация. Описаны алгоритмы и комплекс программ для математического моделирования в системах компьютерной математики нелинейных обобщенно-механических систем. Встроенные в пакет программные процедуры позволяют получать численные решения в форме сплайнов, В-сплайнов и кусочно-заданных функций. Описаны разработанные программные процедуры операций над сплайнами, позволяющие проводить аналитические вычисления с конвертированными численными решениями как с обычными функциями.

Ключевые слова: обыкновенное дифференциальное уравнение, численные методы, нелинейные обобщенно-механические системы.

Abstract. The algorithms and program system for mathematic modeling of nonlinear generic mechanics systems in the computers mathematic are present. The built-insystem programming procedures it makes possible to automatize the process of input equations and arrive at the numeric solutions and output it in functional form. It is provides switching of the numerical solving methods of the nonlinear system differential equations in in the specified point.

Keywords: ordinary differential equation, numerical methods, nonlinear generic mechanics system.

Введение

Объектом исследования настоящей статьи является математическое моделирование нелинейных обобщенно-механических систем (НОМС) в среде компьютерной математики (СКМ) Maple. Такие системы в наиболее общем случае описываются системой нелинейных обыкновенных дифференциальных уравнений (ОДУ), разрешенных относительно старших производных функций $y_i(t)$, вида

$$y_i^{(n_i)} = F_i(y_1, \dots, y_N, y_1', \dots, y_N', y_1'', \dots, y_N'', \dots, y_1^{(n_i-1)}, t) \quad (i = \overline{1, N}), \quad (1)$$

где $y^{(n)} = d^n f / dt^n$ – обозначение n -й производной функции f по независимой переменной t (времени); F_i – непрерывно-дифференцируемые функции своих переменных.

Будем в дальнейшем полагать выполненными начальные условия для системы (1):

$$y_i^{(k)}(t) \Big|_{t=t_0} = C_i^k \quad (k = \overline{1, n_i - 1}; \quad i = \overline{1, N}), \quad (2)$$

соответствующие стандартной задаче Коши, где C_i^k – начальные значения производных k -го порядка функций $y_i(t)$.

Целью исследования является разработка алгоритмов и пакетов (комплекса) программ в СКМ Maple для компьютерного исследования нелиней-

ных обобщенно-механических систем и построения многопараметрических программных процедур визуализации математических моделей этих систем.

1. Блок-схема комплекса программ

Для обеспечения гибкой работы с численными решениями были созданы специальные внутренние программные процедуры, позволяющие проводить стандартные операции анализа функции одной переменной со сплайнами, а также конвертировать их в кусочно-заданные функции. В результате был получен программный аппарат аналитического (приближенного) исследования НОМС в СКМ Maple. Важным достоинством разработанного комплекса программ является его независимость от версии Maple, начиная с версии 1997 г. Maple 5.5 и заканчивая версией 2009 г. Maple 14. Представленный комплекс программ состоит из двух независимых пакетов программ. Пакет программ DifEq содержит программы распознавания введенных: системы дифференциальных уравнений и начальных условий (a), программы автоматического преобразования введенных данных в задачу Коши для нормальной системы относительно унифицированных переменных (b), программы численного решения задачи Коши на заданном интервале с возможностью контроля и переключения метода численного интегрирования при заданном значении независимой переменной (c).

Пакет программ Splines содержит программы генерации равномерных кубических сплайнов и В-сплайнов по заданной функции на заданном интервале (d, f), программы конвертирования сплайнов и В-сплайнов в кусочно-непрерывные функции и обратные операции (e, g), программы операций над сплайнами (h). Операции указанных пакетов интегрируются в программе конвертирования численных решений в кусочно-заданные функции (i). Ниже описаны основные алгоритмы разработанного комплекса программ и продемонстрировано на примерах исполнение программных процедур.

2. Пакет программ преобразования системы уравнений и решения задачи Коши (DifEq)

2.1. Распознавание информации об ОДУ (блок a)

Рассмотрим ОДУ n -го порядка, разрешенного относительно старшей производной функции $y(x)$:

$$y^{(n)} = F(x, y, y', \dots, y^{(n-1)}). \quad (3)$$

Для задания n -й производной функции $y(x)$ по переменной x в СКМ Maple существуют две альтернативные процедуры: `diff(y(x), x$n)` и `(D@@n)(y)(x)`. Вторая из этих процедур предусматривает возможность вычисления производной в заданной точке, что весьма удобно для задания начальных условий. Создадим программную процедуру `DifEq[DiffOp]`, позволяющую извлечь полную информацию о старшей производной в правой части уравнения (4)¹:

```
[> DifEq[DiffOp] := proc (X) local var, s, i, ss, ff, tt:
```

¹ Идею этого алгоритма подсказал А. В. Матросов, которому Ю. Г. Игнатьев выражает признательность.

```

if op(0,X)=diff then: var:=op(X)[2];s:=op(X)[1];
for i from 1 do
if(nops(s)=1) then
ff:=(op(0,s)); tt:=(op(1,s)); break:
else s:=op(s)[1]; end if; end do:
[diff,i,ff,tt]:
elif op(1,op(0,op(0,X)))=D then
[D,op(2,op(0,op(0,X))),
op(-1,op(0,X)),op(-1,X)]:
elif op(0,op(0,X))=D then
[D,1,op(-1,op(1,op(0,X))),op(-1,X)]:
else ff:= op(0,X):tt:=op(-1,X): [0,ff,tt]:
end if:end proc:

```

При действии на производную данная программная процедура создает упорядоченный список – тип оператора производной, порядок производной, имя неизвестной функции и имя независимой переменной. Это позволяет перейти в дальнейшем от обозначений пользователя к некоторым унифицированным внутренним именам зависимых и независимых переменных, что позволяет автоматизировать процессы работы с уравнениями.

2.2. Приведение задачи Коши для системы ОДУ произвольного порядка к задаче Коши для нормальной системы ОДУ с унифицированными именами переменных (блок b)

Блок b включает три программные процедуры, осуществляющие поэтапное конвертирование системы ОДУ (1) с начальными условиями (2) к задаче Коши для нормальной системы ОДУ с унифицированными именами переменных. Знание информации о старшей производной уравнения (4) позволяет конвертировать это уравнение к нормальной системе ОДУ с унифицированными обозначениями, что обеспечивает автоматизацию манипуляций с этими уравнениями. Договоримся о следующем порядке присвоения унифицированных имен зависимых переменных. Для этого рассмотрим упорядоченный список DifEq[MatAlf] из восьми имен: [X,Y,Z,U,V,W,Phi,Theta], который, конечно, можно продолжить, но для этого нет никаких причин практического характера. Обозначим переменную, введенную пользователем, посредством X , ее первую производную – Y , вторую производную – Z и т.д., а независимую переменную посредством t . Полученная система и будет являться искомой нормальной системой ОДУ с унифицированными именами переменных:

$$\frac{dX_i^1}{dt} = X_i^2; \quad \frac{dX_i^2}{dt} = X_i^3; \dots; \quad \frac{dX_i^{n_i-1}}{dt} = F_i \quad (i = \overline{1, N}), \quad (4)$$

где $F = F(t, X_1^1, \dots, X_N^{n_N})$ (для краткости положено $X^k = \text{DifEq}[\text{MatAlf}]_k \rightarrow X^1 = X, X^2 = Y, \dots$), и системой начальных условий для этих уравнений, соответствующих начальным условиям (2):

$$X_i^k(t_0) = C_i^k \quad (k = \overline{1, n_i-1}; \quad i = \overline{1, N}), \quad (5)$$

2.2.1. Приведение ОДУ произвольного порядка к нормальной системе ОДУ с унифицированными именами переменных

Первый из указанных алгоритмов – алгоритм приведения ОДУ произвольного порядка к нормальной системе ОДУ, реализуется программной процедурой `DifEq[Oden_ConvNorm]`. Процедура `DifEq[Oden_ConvNorm]` использует две промежуточные процедуры: `DifEq[DiffOp]` и `DifEq[MatAlf]`. Рассмотрим исполнение этой процедуры на примере уравнения в обозначениях пользователя:

$$S^{IV}(\xi) = \left(S^{III}(\xi)\right)^2 S^3(\xi) + S'(\xi) + \sin \xi. \quad (6)$$

Результат применения нашей процедуры к уравнению (7) получается следующим:

$$\begin{aligned} &[>\text{DifEq}[\text{Oden_ConvNorm}] (\text{Eq1}, 2); \\ &[[4, S, \xi, D], [X_2, Y_2, Z_2, U_2], \\ &[S(\xi) = X_2(t), D(S)(\xi) = Y_2(t), \\ &(D^{(2)})(S)(\xi) = Z_2(t), (D^{(3)})(S)(\xi) = U_2(t), \xi = t], \\ &\left[\frac{d}{dt} X_2(t) = Y_2(t), \frac{d}{dt} Y_2(t) = Z_2(t), \frac{d}{dt} Z_2(t) = U_2(t), \right. \\ &\left. \frac{d}{dt} U_2(t) = U_2(t)^2 X_2(t)^3 + Y_2(t) + \sin(t) \right] \end{aligned}$$

Он состоит из упорядоченного списка четырех упорядоченных списков: первый содержит: [порядок уравнения, искомую функцию, независимую переменную, тип дифференциального оператора], введенные пользователем. Второй список содержит упорядоченную систему новых функций. Третий список содержит правила замены переменных в ОДУ, четвертый список содержит упорядоченную нормальную систему ОДУ. Как видно, процедура `DifEq[Oden_ConvNorm]` содержит два обязательных параметра, первый – дифференциальное уравнение, а второй k – любое имя или число, необходимое для пометки уравнения и входящих в него переменных. Соответствующая метка помещается как нижний индекс у соответствующих величин. В рассмотренном примере $k = 2$. В частности, если мы хотим отказаться от индексирования функций, достаточно вместо индекса ввести `` – два апострофа.

2.2.2. Приведение системы ОДУ произвольного порядка к нормальной системе ОДУ с унифицированными именами переменных

Аналогично вводится и однопараметрическая процедура приведения системы ОДУ (1) к нормальной системе ОДУ `DifEq[SysOden_ConvNorm]`. Результат применения этой процедуры, который для краткости мы не приводим, представляется в виде трех упорядоченных списков: в первом содержится один элемент S – число уравнений нормальной системы (число степеней свободы НОМС); во втором – преобразования пользовательских функций к унифицированным функциям нормальной системы; третий список есть упорядоченная нормальная система обыкновенных дифференциальных уравне-

ний, причем первые $S - N$ уравнений представляют собой результат стандартной замены переменных вида $y' = z$.

2.2.3. Приведение задачи Коши (1), (2) к задаче Коши для нормальной системы ОДУ с унифицированными именами переменных

На основе программ распознавания системы ОДУ (блок а) и программ приведения ОДУ к нормальному виду строится двухпараметрическая программная процедура `DifEq[SysCauchy_ConvNorm]` (System ODE, Inits Conditions) приведения системы ОДУ произвольного порядка (1) с начальными условиями (2) к задаче Коши для нормальной системы ОДУ относительно унифицированных переменных $X_i^k(t)$ (5) с начальными условиями (6), т.е. к задаче Коши для нормальной системы ОДУ. Результат применения процедуры выводит упорядоченный список из шести упорядоченных списков. В первом содержатся два числа: S – число уравнений нормальной системы, $M = \max(n_1, \dots, n_N)$ – максимальный порядок уравнений в исходной системе уравнений; во втором – упорядоченные списки новых переменных (число этих списков равно M), выбранных по следующему принципу: в первом списке содержатся новые переменные $X[i]$, полученные из независимых функций пользователя, во втором – первые производные от этих переменных $Y[i]$, если вторые производные от этих переменных содержатся в системе ОДУ, и т.д. до $(M - 1)$ -го списка. Таким образом, количество внутренних списков независимых функций совпадает с максимальным порядком уравнений исходной системы M . Третий список содержит преобразования пользовательских функций к унифицированным функциям нормальной системы, четвертый список есть упорядоченная нормальная система обыкновенных дифференциальных уравнений, причем первые $S - N$ уравнений представляют собой результат стандартной замены переменных вида $y' = z$; в пятом содержатся начальные условия для нормальной системы ОДУ в форме (6), в шестом – один элемент – начальное значение независимой переменной. Созданная процедура удобна для извлечения различной информации об исходной системе дифференциальных уравнений и начальных условий.

2.3. Программные процедуры численного решения задачи Коши для системы (1) (блок с)

Блок с содержит две программные процедуры численного решения системы ОДУ (1) с начальными условиями (2) – трехпараметрическую программную процедуру `DifEq[NumDsolve]` (System ODE, Inits Conditions, Method) и пятипараметрическую процедуру `DifEq[ReNumDsolve]` (System ODE, Inits Conditions, Method1,x1,Method2). Первая команда создает процедуру решения системы ОДУ с помощью метода Method, встроенного в пакет Maple; значение этого параметра 45 соответствует методу Рунге-Кутты 4–5 порядков, 78 – методу Рунге-Кутты 7–8 порядков, *rosenbrock* – методу Розенброка, *stiff* – методу stiff интегрирования жестких уравнений, *classic* – классическому методу (по умолчанию методом Эйлера), *taylor* – методом разложения в ряды Тейлора (см., например, [1, 2]). При этом вывод решений осуществляется в виде упорядоченного списка с вложенными в него M упорядоченными списками. При этом первый упорядоченный список содержит численные значения N искомых функций: $[X[1](t), \dots, X[N](t)]$, порядок записи

значений функций в списке совпадает с порядком записи дифференциальных уравнений системы. Второй упорядоченный список содержит значения первых производных тех функций, производные которых не ниже второго порядка содержатся в системе ОДУ. Порядок записи значений первых производных функций в списке совпадает с порядком записи дифференциальных уравнений системы, при этом пропускаются значения производных тех функций, производные выше первого порядка которых не содержатся в системе ОДУ. Третий упорядоченный список содержит значения первых производных тех функций, производные которых не ниже третьего порядка содержатся в системе ОДУ. Порядок записи значений вторых производных функций в списке совпадает с порядком записи дифференциальных уравнений системы, при этом пропускаются значения вторых производных тех функций, производные выше второго порядка которых не содержатся в системе ОДУ, и т.д.

Для демонстрации формата ввода системы и вывода решений рассмотрим *пример интегрирования существенно нелинейной системы ОДУ с максимальным третьим порядком производных*

$$\frac{d^2 F}{dx^2} = \Phi \frac{d^2 Z}{dx^2}; \quad \frac{d^3 Z}{dx^3} = -F^2; \quad \frac{d\Phi}{dx} = F^3 = \sin x \quad (7)$$

с начальными условиями

$$F(\pi)=1; \quad F'(\pi)=0; \quad Z(\pi)=1; \quad Z'(\pi)=-1; \quad Z''(\pi)=-1/3; \quad \Phi(\pi)=0. \quad (8)$$

Процедура решения системы (8) с начальными условиями (9) методом Розенброка осуществляется одной командой:

```
> SS:=DifEq[NumDsolve](Eqs, Inits1, rosenbrock);
```

при этом решению мы произвольно присвоили имя SS. Тогда численное решение выводится функцией SS(x). В данном случае список решений состоит из трех внутренних списков ($M=3$), в первом из них содержатся три числа ($N=3$), это значения функций $[F(1), Z(1), \Phi(1)]$; во втором списке содержится два числа, поскольку в системе (8) содержатся лишь две производные не ниже второго порядка, это значения первых производных $[F'(1), Z'(1)]$, наконец, третий список содержит лишь одно число – это значение второй производной функции $Z''(1)$. Для вывода одного из этих списков (i -го) достаточно применить процедуру $SS(1)[i]$.

Программная процедура

```
DifEq[ReNumDsolve](Eqs, Inits, Method1, x1, Method2)
```

построена на основе рассмотренной выше процедуры $DifEq[NumDsolve]$ и встроенной в Maple программной процедуры $piecewise(x \geq x_0 \text{ and } x \leq x_1, f_1, x > x_1, f_2)$ кусочно-заданной функции, так что на интервале $[x_0, x_1]$ при численном интегрировании системы ОДУ применяется метод $Method1$, а на интервале $(x_1, \dots)$ – $Method2$. При этом начальными условиями для численного интегрирования методом $Method2$ являются результаты численного интегрирования системы ОДУ в точке x_1 , полученные методом $Method2$. Указанный метод можно назвать *методом интегрирования с перезагрузкой начальных условий*. Его следует применять в тех случаях, когда на некотором отрезке стандартные методы интегрирования не дают хороших результатов. Следует отме-

тить, что программные процедуры `DifEq[NumDsolve]` и `DifEq[ReNumDsolve]` не требуют выполнения пользователем предварительных операций по приведению системы ОДУ к нормальному виду, так как эти операции являются встроенными процедурами в указанные; нормирование системы ОДУ и извлечение всей необходимой информации производится автоматически.

2.4. Точность и скорость вычислений

Для тестирования пакета программ `DifEq` на точность вычислений, а также выяснения немаловажного для задач компьютерного моделирования вопроса о скорости вычислений различными численными методами можно провести численное интегрирование различных точно решаемых нелинейных систем ОДУ различными методами с помощью программных `DifEq[NumDsolve]` и `DifEq[ReNumDsolve]`. При этом реальное время, необходимое для интегрирования системы и представления численных результатов в графическом виде, можно вычислить с помощью встроенной в Maple функции `time()`, которая определяет точное время в секундах по встроенному в компьютер таймеру. Точность и скорость вычислений тестировались по целому ряду точно решаемых нелинейных систем ОДУ, примеры которых можно найти во многих учебниках по дифференциальным уравнениям. Рассмотрим следующий пример задачи Коши для нелинейной системы ОДУ [3]:

$$\frac{dy}{dx} = \frac{y}{x}; \quad \frac{dz}{dx} = -\frac{y}{z}, \quad (9)$$

с начальными условиями

$$y\left(\frac{1}{2}\right) = \frac{1}{2}; \quad z\left(\frac{1}{2}\right) = \frac{\sqrt{3}}{2}. \quad (10)$$

Указанная задача Коши точно решается:

$$y(x) = x; \quad z(x) = \sqrt{1-x^2} \quad (z \geq 0), \quad (11)$$

так что $x^2 + y^2 = 1$, т.е. конфигурационной траекторией системы является окружность единичного радиуса.

Хотя найденное точное решение задачи Коши имеет весьма простой вид, с точки зрения применения численных методов интегрирования система (9) является достаточно неудобной: она не является автономной и имеет особые точки при $x = 0$, $z = 0$. Поэтому система ОДУ (10) и подобные ей удобны для тестирования численных методов решения ОДУ. Численное решение, например методом Розенброка, сразу достигается командой

```
> SS1:=DifEq[NumDsolve]([diff(y(x),x)=y(x)/x,diff(z(x),x)=-y(x)/z(x)],
[y(1/2)=1/2,z(1/2)=sqrt(3)/2],rosenbrock);
```

и выводится в функциональном списочном виде, соответствующем векторной функции скалярного аргумента.

Продemonстрируем построение *конфигурационной траектории* системы (11), (12) $r = [y(x), z(x)] = [X_1(t), X_2(t)]$ при интегрировании методом Розенброка. При этом построение траектории достигается простой стандартной командой Maple, в которой мы указали две необязательные опции:

scaling = CONSTRAINED – для сохранения масштаба, color = black – для определения цвета кривой:

```
>plot([SS4(t)[1,1],SS4(t)[1,2],t=-1..1],scaling=CONSTRAINED,color=black);
```

Для выяснения вопроса о точности методов интегрирования будем вычислять логарифм модуля разности численного $Z(t)$ и точного $Z_0(t)$ решений (11): $L(t) = \lg |Z(t) - Z_0(t)|$ (рис. 1).

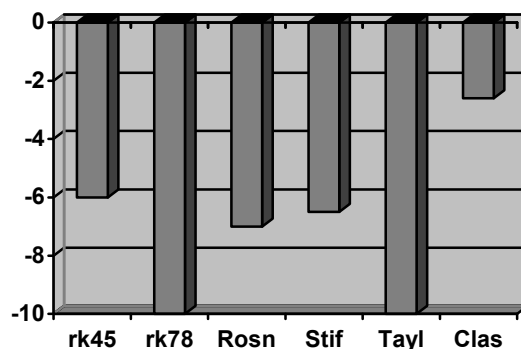


Рис. 1. Зависимость среднего логарифма абсолютной погрешности, $\langle L \rangle$, методов численного решения задачи Коши: rk45 – метод Рунге-Кутты 4–5 порядков; rk78 – метод Рунге-Кутты 7–8 порядков; Rosn – метод Розенброка; Stif – метод stiff; Tayl – метод разложений Тейлора; Clas – классический метод

Время T расчета и построения конфигурационных диаграмм можно вычислить с помощью встроенной в Maple команды time(). На гистограмме (рис. 2) показана зависимость времени выполнения данной операции от метода численного интегрирования [14–16].

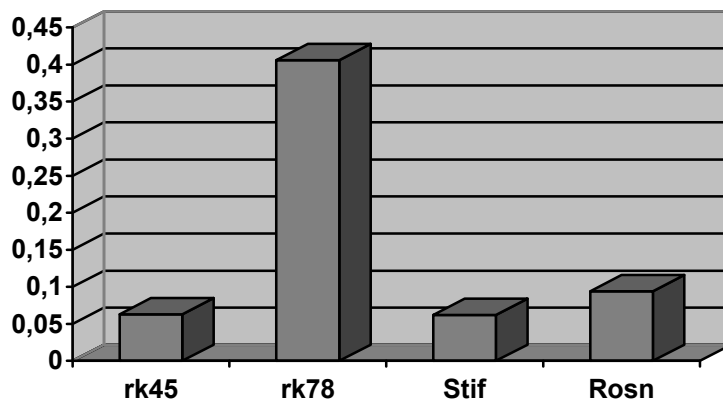


Рис. 2. Зависимость среднего времени T (в секундах) расчета и построения конфигурационной диаграммы для системы (11), (12) от методов численного решения задачи Коши: rk45 – метод Рунге-Кутты 4–5 порядков; rk78 – метод Рунге-Кутты 7–8 порядков; Rosn – метод Розенброка; Stif – метод stiff; Clas – классический метод

3. Программные процедуры сплайновой интерполяции функций

Для создания аппарата аналитического исследования решений нелинейной системы ОДУ и проведения компьютерного моделирования нелинейных обобщенно-механических систем были созданы специализированные программные процедуры, позволяющие выполнять над сплайнами алгебраические и интегродифференциальные операции, а также операции взаимного конвертирования сплайнов и В-сплайнов в кусочно- и кусочно-параметрически заданные функции. Таким образом, удалось добиться важного результата – создать комплекс программ, позволяющий проводить аналитическое компьютерное исследование нелинейных обобщенно-механических систем. Используя процедуру Spline, создадим необходимую в дальнейшем промежуточную 6-параметрическую обратную к Spline простую процедуру генерации равномерных n -кусочных кубических сплайнов относительно функции $f(x)$, заданной на отрезке $[a, b]$, Splines[Spline F] (f, x, a, b, n, z) с передачей имени z ее независимому аргументу:

```
> Splines[SplineF] := proc (f, x, a, b, n, z) local F, X, Basa:
F := (X) -> subs (x=X, f): Basa := [seq ([a+i*(b-a)/n,
eval (F(a+i*(b-a)/n))], i=0..n)]:
CurveFitting[Spline] (Basa, z): end proc:
```

Введем предварительно трехпараметрическую процедуру Splines[Conv_List](sp, t, z) конвертирования сплайна sp функции f на заданном отрезке в упорядоченный список, состоящий из четного числа элементов, в котором каждая пара представляет собой упорядоченный набор элементов, первый из которых – неравенство, устанавливающее верхнюю границу интервала, а второй – сплайн на данном интервале:

```
> Splines[Conv_List] := proc (sp, t, z) local LSS0, LSS,
T, ddd, NN0, NN1, t1, DDD:
LSS0 := convert (sp, list): LSS := (T) ->
subs (t=T, LSS0): NN0 := nops (LSS0): NN1 := (NN0+1) / 2:
ddd := rhs (op (3, LSS (t))) - rhs (op (1, LSS (t))):
t1 := rhs (op (1, LSS (t))) - ddd: DDD := t1 + NN1 * ddd:
[seq (op (i, LSS (z)), i=1..NN0-1),
(z < DDD, op (-1, LSS (z)))] : end proc:
```

Заметим, что процедура автоматически распознает параметры сплайна. Создадим также и обратную процедуру конвертирования списка в кусочно-заданную функцию, piecewise:

```
> Splines[Conv_Piece] := proc (Ls, t, z)
local T, LS0, LSS0:
LS0 := (T) -> subs (t=T, Ls):
LSS0 := (t) -> subs (op (-2, LS0 (t)) = NULL, LS0 (t)):
piecewise (op (LSS0 (z))):
end proc:
```

Здесь применена встроенная процедура Maple *NULL* – команда пустого множества $\emptyset$. В качестве примеров генерации сплайнов мы рассмотрим следующие три:

```
> PPS6 := (z) -> Splines[SplineF] (sin(x), x, 0, 2*Pi, 6, z):
```

```
>PPS12:=(z)->Splines[SplineF](sin(x),x,0,2*Pi,12,z):
>PPS24:=(z)->Splines[SplineF](sin(x),x,0,2*Pi,24,z):
```

Все три сплайна созданы по отношению к функции $f(x) = \sin x$ на отрезке $[0, 2\pi]$ с именем независимой переменной z : сплайн PPS6 содержит шесть интервалов, PPS12 – 12 интервалов и PPS24 – 24 интервала. Эффективную проверку корректности созданных программных процедур можно провести, дважды конвертируя сгенерированный сплайн с помощью процедур `Splines[Conv_List]` и `Splines[Conv_Piece]`. Проверка показывает результаты, которые графически неотличимы от оригинала.

В заключение авторы выражают признательность профессору В. П. Дьяконову и Д. П. Голоскокову за полезное обсуждение результатов статьи, а профессору М. Н. Кирсанову и А. В. Матросову за ряд ценных замечаний по алгоритмизации процедуры определения параметров дифференциальных операторов в пакете Maple.

Список литературы

1. **Фокс, Л.** Numerical Solution of Ordinary Differential Equations for Scientists and Engineers / L. Fox, D. F. Mayers. – N.Y. : Springer, 1987. – 624 p.
2. **Филиппов, А. Ф.** Сборник задач по дифференциальным уравнениям / А. Ф. Филиппов. – Ижевск : НИЦ «Регулярная и хаотическая динамика», 2000. – 176 с.
3. **Роджерс, Д.** Математические основы машинной графики / Д. Роджерс, Дж. Адамс. – М. : Мир, 2001. – 452 с.

Игнатъев Юрий Геннадьевич

доктор физико-математических наук,
профессор, заведующий кафедрой
геометрии и математического
моделирования, Татарский
государственный гуманитарно-
педагогический университет

E-mail: ignatjev_yu@rambler.ru

Ignatyev Yuriy Gennadyevich

Doctor of physical and mathematical
sciences, professor, head of sub-department
of geometry and mathematical modeling,
Tatar State University of Humanities
and Pedagogy

Абдулла Халид Хусейн

аспирант, Татарский государственный
гуманитарно-педагогический
университет

E-mail: khaled_alyfee@yahoo.com

Abdulla Khalid Khuseyn

Postgraduate student, Tatar State
University of Humanities and Pedagogy

УДК 531; 531.3; 531.5; 517.9; 519.6; 519.8

Игнатъев, Ю. Г.

Математическое моделирование нелинейных обобщенно-механических систем в системе компьютерной математики Maple / Ю. Г. Игнатъев, Х. Х. Абдулла // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2010. – № 2 (14). – С. 66–75.