УДК 629-78

Д.А. Храмов

ОСОБЕННОСТИ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ДИНАМИКИ ЭКСПЕРИМЕНТАЛЬНОЙ ТРОСОВОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНЫХ СРЕДСТВ НА ОСНОВЕ ЯЗЫКА PYTHON

Анотація. Проведено порівняльний аналіз використання пакетів Python(x,y) і MATLAB для дослідження динаміки експериментальної тросової системи. Показано, що в рамках розглянутої задачі обидва пакети є порівнянні за зручності роботи й можливостям для програмування. Зазначено перспективи використання вільного програмного забезпечення, заснованого на Python, для реалізації різних підходів до моделювання динамічних систем.

Создание космических тросовых систем (КТС) — систем, состоящих из спутников, соединенных протяженными гибкими нитями — представляется одним из перспективных направлений современной космонавтики [1—3]. Математические модели динамики КТС, как правило, достаточно громоздки. Поэтому верификация компьютерных программ, реализующих эти модели, представляет собой серьезную проблему [4]. Надежность результатов расчетов может быть обеспечена благодаря использованию различных программных средств и различных подходов к моделированию динамики подобных систем.

Рассмотрим систему, состоящую из твердого тела, подвешенного на невесомой упругой нити (рис. 1). Она представляет собой частный случай экспериментальной тросовой системы [5] и может служить для проверки программ расчета динамики последней.

В [6] рассмотрены особенности исследования динамики данной системы с помощью средств численного моделирования (MATLAB), с использованием символьного вывода уравнений движения (Maple), а также физическое моделирование[1] на основе блок-схемы рассматриваемой системы (MapleSim). Поскольку указанные программы являются коммерческими, то их совместное использование может потре-

[1] терм, н, пр, надлежащ, й разработч, кам [7].

бовать значительных финансовых затрат. В связи с этим возникает вопрос: существует ли свободное программное обеспечение (ПО), способное заменить коммерческое в данной области расчетов. Для определенности, ограничимся сравнением с MATLAB.

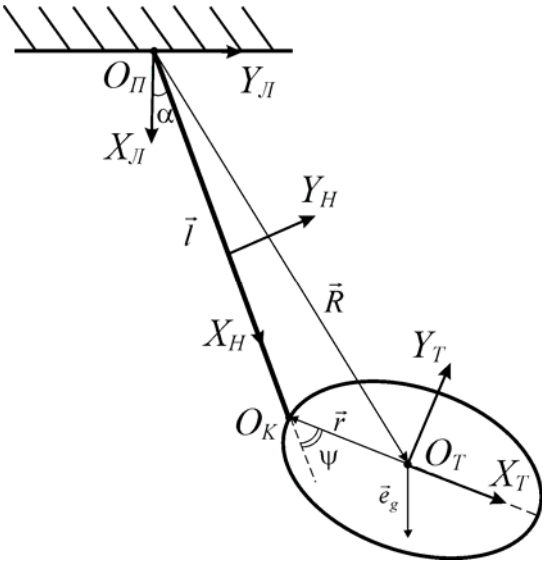

Рисунок 1 – Схема модели

Целью нашей работы является выбор свободного ПО для численного моделирования рассматриваемой системы, и сравнительный анализ особенностей моделирования с помощью данного ПО и MATLAB.

Математическая модель

Запишем уравнения движения системы, изображенной на рис. 1. Предполагая, что массой нити в сравнении с массой тела можно пренебречь, получим

$$\begin{aligned} m\ddot{\vec{R}} &= \vec{F}_{mp} + mg\vec{e}_g, \\ \dot{\vec{L}} &= \vec{r} \times \vec{F}_{mp}, \end{aligned} \qquad (1)$$

где m — масса тела; $\vec{R}$ — радиус-вектор центра масс тела (точка O_T) относительно точки подвеса нити $O_П$; $\vec{F}_{mp}$ — сила натяжения нити; $\vec{e}_g$ — единичный вектор по направлению действия ускорения земного тяготения $\vec{g}$; $\vec{r} = \overrightarrow{O_T O_K}$ (рис. 1); $\vec{L}$ — кинетический момент движения тела относительно центра масс.

Предположим далее, что упругие свойства нити описываются законом Гука, а рассеивание энергии движения системы происходит за счет вязкого трения в материале нити

 ISSN 1562-9945

$$\vec{F}_{тр} = -\left[c\frac{\vec{l}}{l}\frac{(l-d)}{d} + \chi\, \dot{l}\frac{\vec{l}}{l} \right] \delta, \qquad \delta = \begin{cases} 1 & l > d, \\ 0 & l \le d, \end{cases}$$

где $\vec{l} = \overrightarrow{O_П O_К}$ (рис. 1), $l = \left|\vec{l}\right|$, $\dot{l} = \left(\vec{l}, \dot{\vec{l}}\right) \Big/ \left|\vec{l}\right|$; d — номинальная длина нити; c — коэффициент жесткости; χ — коэффициент демпфирования.

Для описания движения системы введем следующие правые системы координат.

$O_П X_л Y_л Z_л$ — лабораторная система координат (ЛСК) с началом в точке подвеса нити $O_П$. Ось $O_П X_л$ направлена вдоль местной вертикали, ось $O_П Z_л$ — перпендикулярно плоскости наблюдения в сторону наблюдателя, ось $O_П Y_л$ лежит в плоскости наблюдения;

$O_П X_н Y_н Z_н$ — система координат, связанная с нитью (СКН) с началом в точке $O_П$ (на рис. 1 начало координат СКН смещено из соображений наглядности). Ось $O_П X_н$ направлена вдоль линии нити, соединяющей точку $O_П$ и точку $O_К$, ось $O_П Z_н$ — вдоль мгновенной угловой скорости вращения вектора $\overrightarrow{O_П O_К}$;

$O_Т X_Т Y_Т Z_Т$ — система координат, связанная с телом (СКТ) с началом в центре масс тела $O_Т$. Оси СКТ направлены по главным центральным осям инерции тела.

Взаимная ориентация систем координат описывается следующим образом: $O_П X_н Y_н Z_н$ и $O_Т X_Т Y_Т Z_Т$ — углами Крылова ϕ, θ, ψ, соответственно углами крена, тангажа и рысканья; $O_П X_л Y_л Z_л$ и $O_П X_н Y_н Z_н$ — углами α, β (рис. 2) и радиусом-вектором $\vec{l} = \overrightarrow{O_П O_К}$ (рис. 1).

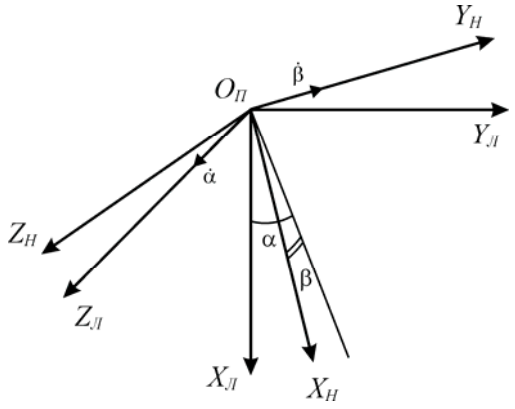

Рисунок 2 — Взаимная ориентация осей ЛСК и СКН

Матрицы перехода между системами координат:

$$A_{лн} = \begin{pmatrix} \cos\alpha\cos\beta & -\sin\alpha & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta & \cos\alpha & \sin\alpha\sin\beta \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$A_{нт} = \begin{pmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta \\ \sin\phi\sin\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & -\sin\phi\cos\theta \\ -\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi & \cos\phi\cos\theta \end{pmatrix}.$$

Кинематические уравнения, связывающие производные углов α, β по времени с проекциями вектора угловой скорости $\vec{\omega}_H$ движения СКН относительно ЛСК на оси СКН имеют вид

$$\dot{\alpha} = -\frac{\bar{\omega}_H(1)}{\cos\beta}, \qquad (2)$$
$$\dot{\beta} = \bar{\omega}_H(2).$$

Кинематические уравнения, связывающие производные углов Крылова ϕ, θ, ψ по времени с проекциями вектора угловой скорости $\vec{\omega}_T$ движения СКТ относительно ЛСК на оси СКТ:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{\cos\theta} \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{pmatrix} \bar{\omega}_T. \qquad (3)$$

Уравнения движения тела (1) и кинематические уравнения (2, 3) образуют замкнутую систему уравнений 1-го порядка с 12 неизвестными.

Выбор программных средств

Ориентируясь на MATLAB как по возможностям (векторизация кода, наличие необходимых библиотек), так и по удобству использования (простота овладения языком, наличие документации, среды разработки, отладчика, профилировщика), естественно рассмотреть его ближайшие аналоги — пакеты GNU Octave [8] и Scilab [9].

Несмотря на то, что отдельные тесты [10] показывают некоторое преимущество Octave по скорости вычислений в сравнении с MATLAB, Octave в целом производит впечатление пакета, находящегося в стадии разработки. Документация пакета скудна, а по сообщениям об ошибках зачастую трудно определить вызвавшие их причины. Собственная среда разработки отсутствует, а визуализация графиков происходит с помощью gnuplot.

 ISSN 1562-9945

В пакете Scilab имеется собственная среда разработки, поддерживается большое число библиотек для численных расчетов (Lapack, LINPACK, ODEPACK, Atlas и др.). В состав пакета также входит Scicos — инструмент для редактирования блочных диаграмм и симуляции (аналог Simulink). Существует литература на русском языке (например, [11]).

Однако оба пакета, что называется, «движутся в колее» MATLAB, не предлагая пользователю каких-либо оригинальных решений.

Возможен другой подход: использование языка программирования общего назначения, снабженного необходимыми библиотеками. Среди таких языков в научных вычислениях вплоть до недавнего времени широко использовались C/C++ и Fortran. Однако в последние годы в этом качестве все чаще используется Python [12, 13]. Достаточно сказать, что в рейтинге популярности языков программирования от 12.2011 Python занимает 8-е место (тогда как MATLAB — 22-е, а Fortran — 28-е) [14].

Для работы, кроме собственно Python (в комплект входят также средства отладки и профилирования), понадобится среда разработки (например, Spyder), а также библиотеки для научных расчетов Numpy (работа с массивами), Scipy (решение дифференциальных уравнений), графическая библиотека Matplotlib и консоль IPython. Все это ПО есть в открытом доступе, как по отдельности, так и в виде готовых сборок.

Среди последних обращает на себя внимание Python(x,y) [15, 16] — комплект из более чем 60 библиотек, который обновляется примерно раз в месяц. Среди библиотек, помимо перечисленных выше: SymPy — выполнение символьных расчетов, PP — параллельные вычисления, Qt — создание графических пользовательских интерфейсов, OpenCV — обработка изображений и компьютерное зрение. Именно эту сборку мы использовали в дальнейшей работе.

Особенности программной реализации

Основой расчетных программ на обоих языках является процедура численного интегрирования уравнений движения (рис. 3).

Одним из достоинств MATLAB является возможность векторизации кода, т. е. выполнения операций над массивом в целом, без ис-

пользования циклов. Такая возможность для Python реализована в библиотеке Numpy [17].

В Python, в отличие от MATLAB, необходимо в явном виде импортировать требуемые функции из библиотек. Так, использованию функций Numpy в коде предшествует строка

from numpy import *

позволяющая использовать все функции Numpy в основной программе.

Рисунок 3 — блок-схема программы расчета динамики системы

В табл. 1 приведены соответствия функций MATLAB и Numpy. Заметим, что индексация массивов в Python начинается с 0, а не с 1, как это принято в MATLAB.

Таблица 1

MATLAB	Numpy	Комментарии
[1 2 3; 4 5 6]	array([[1.,2.,3.], [4.,5.,6.]])	матрица размерности 2x3
a(2,5)	a[1,4]	элемент второй строки и пятого столбца
a(1:5,:)	a[0:5,:]	первые пять строк a
a'	a.T	транспонирование a
a * b	dot(a,b)	умножение матриц
1:10	arange(1.,11.)	создание диапазона значений от 1 до 10 с шагом 1
[a; b]	concatenate((a,b))	конкатенация строк a и b
A\b	linalg.solve(a,b)	решение системы A x = b

Более подробно сравнение реализаций операций над массивами в Numpy и MATLAB приведено на странице «NumPy for Matlab Users» [18]. Краткое введение в Numpy на русском языке приведено в [19].

 ISSN 1562-9945

Численное интегрирование системы ОДУ осуществляется с помощью функции integrate.odeint библиотеки Scipy [20]. Импорт этой функции осуществляется строкой

from scipy import integrate

Некоторые из функций-решателей ОДУ в MATLAB и их аналоги из Scipy приведены в табл. 2.

Таблица 2

MATLAB	Scipy	Метод интегрирования
ode113	integrate.odeint	Адамса
ode45	scipy.integrate.ode(f).set_integrator ('dopri5')	Рунге-Кутта порядка 4 и 5, основанный на модификации Дормана-Принса[2]

Рисование графика выполнялось с помощью библиотеки Matplotlib [21]. Строка кода, в которой задается возможность обращения к функциям Matplotlib через префикс p, имеет вид:

import matplotlib.pyplot as p

API Matplotlib специально разработан так, чтобы походить на API MATLAB. Фрагменты кода, отвечающие за построение графика с нанесением координатной сетки, приведены в табл. 3.

Таблица 3

MATLAB	Matplotlib
plot(t,x) grid on	p.plot(t,x) p.grid() p.show()

Полный текст программ находится по адресу [22].

Результаты расчетов

Расчеты проводились при следующих значениях параметров системы: $m = 1$ кг; моменты инерции тела $J_x = J_y = J_z = 1$ кг·м^2; $d = 1$ м; $c = 100$ Н; $\chi = 0$; $\bar{r} = [-0{,}1;\ 0;\ 0]$ (проекция $\vec{r}$ на оси СКТ).

[2] Я в программе не использовался.

Численное интегрирование проводилось методом Адамса на промежутке времени $[0,50]$ с.

Исследовались зависимости углов α (рис. 1) и ψ, а также длины нити l от времени. Результаты расчетов, выполненных в Python, приведены на рис. 4—6 ($l(0) = 1$ м, $\alpha(0) = 10^\circ$, $\psi(0) = 0^\circ$).

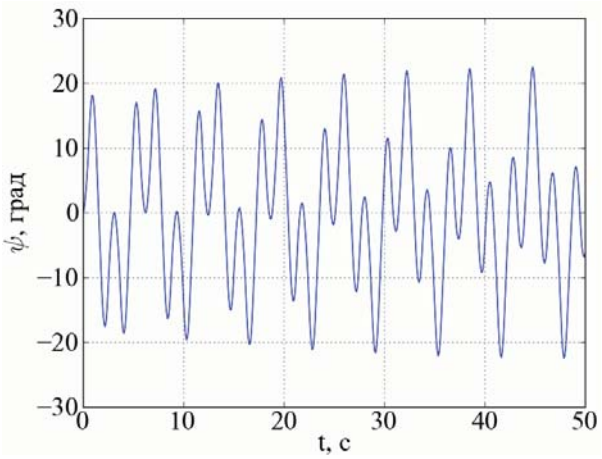

Рисунок 4 — График изменения угла ψ со временем

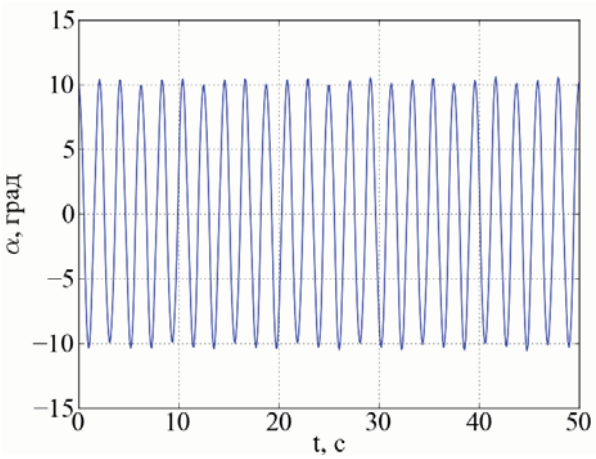

Рисунок 5 — График изменения угла α со временем

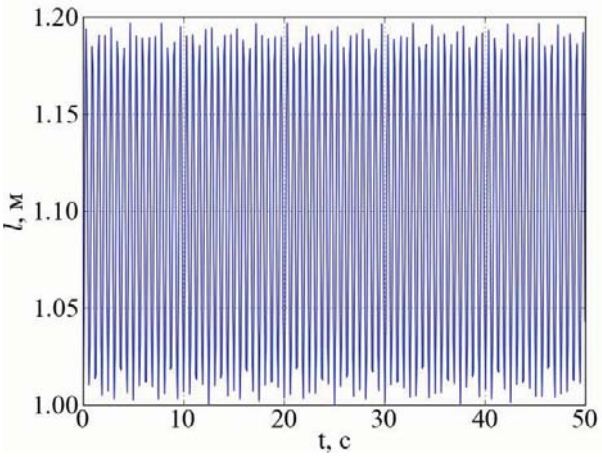

Рисунок 6 — График изменения длины нити l со временем

На рис. 7 приведен график изменения во времени модуля разности расчетных значений угла ψ, полученных в MATLAB и Python (соответствующие графики для α и l аналогичны приведенному на рис. 7).

Разница расчетных значений, по-видимому, связана с тем, что пакеты реализуют различные варианты метода Адамса [23]. Так, в MATLAB решатель ode113 основан на методе Адамса-Башфорта-Моултона (Adams-Bashforth-Moulton PECE), тогда как функция

 ISSN 1562-9945

integrate.odeint в Skypy использует решатель Lsoda из FORTRAN-библиотеки Odepack.

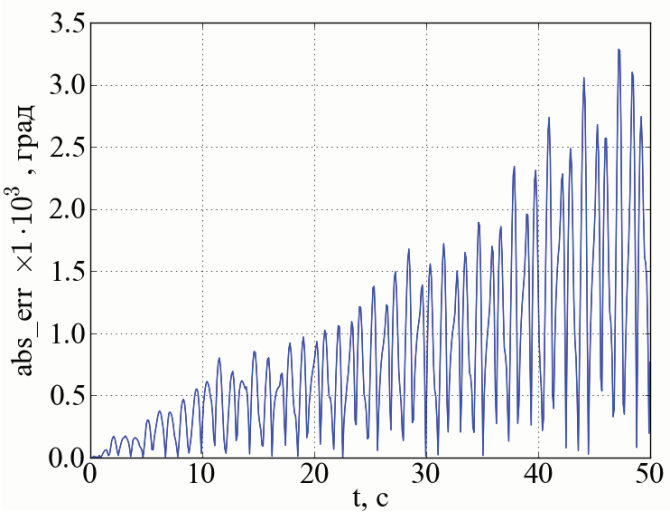

Рисунок 7 — График изменения во времени модуля разности расчетных значений угла ψ, полученных в MATLAB и Python

В таблице 4 приведены результаты тестирования скорости расчетов MATLAB и Python при различном числе точек внутри промежутка интегрирования.

Таблица 4

Время расчета, с	Число точек				
	50	500	5000	50 000	500 000
MATLAB	1,09	1,11	1,28	2,58	16,9
Python	4,90	4,68	4,73	6,3-	6,09

Для тестирования использовался компьютер с процессором Intel Core 2 Duo 2.33 ГГц, 2 Гб DDR 2 ОЗУ, Windows XP SP 3 32-bit. Использовались MATLAB v7.11.0.584 (R2010b) и Python(x,y) 2.7.2.1. Тест был выполнен в консольном режиме, результаты усреднены 10 раз.

Результаты тестирования показывают некоторое преимущество MATLAB в скорости расчетов. Следует, однако, отметить, что при этом не предпринималось каких-либо специальных мер по оптимизации Python-программы. В то же время результаты других тестов, в частности, [24], показывают, что такие меры могут привести к существенному росту производительности. В частности, использование Cython [25] (такая возможность есть в Python(x,y)), что влечет за собой незначительную переделку программы, позволяет добиться быстродействия, сравнимого с программами, написанными на C++.

Выводы и перспективы использования

Сравнение Python(x,y) с MATLAB показало, что в рамках рассмотренной задачи оба пакета обладают сопоставимыми возможностями для программирования и удобством работы. Наличие обширной документации и литературы на русском языке позволяют быстро освоить Python. Существенно и то, что данное ПО никак не связано с MATLAB, и в своем развитии опирается на активное и быстро растущее сообщество Python-программистов.

Использование свободного ПО, основанного на Python, представляется перспективным и для реализации других подходов к моделированию динамических систем, рассмотренных в [5]. Так, SymPy [26] — Python-библиотека для символьных вычислений — уже сейчас обладает возможностями, сравнимыми с Symbolic Math Toolbox MATLAB. Заявленной же целью разработчиков является создание полнофункциональной системы компьютерной алгебры. Система компьютерной алгебры Sage [27], использующая в качестве встроенного языка Python, помимо символьных вычислений позволяет подготавливать научно-техническую документацию с использованием редактора формул и в перспективе может заменить такие пакеты как Maple и Mathematica. JModelica.org [28] — программное средство для моделирования сложных динамических систем, основанное как и MapleSim на языке Modelica, использует Python в качестве языка сценариев.

ЛИТЕРАТУРА

1. Белецкий В. В. Динамика космических тросовых систем / В. В. Белецкий, Е. М. Левин. — М. : Наука, 1990. — 329 с.
2. Lorenzini E. C. Tethers in Space Handbook / E. C. Lorenzini, M. L. Cosmo. — 3rd edition. — Smithsonian Astrophysical Observatory, 1997. — 241 p. ([Электронный ресурс] — Режим доступа : http://www.tethers.com/papers/TethersInSpace.pdf)
3. Levin E. M. Dynamic analysis of space tether missions / E. M. Levin. — San Diego: American Astronautical Society, 2007. — 453 p.
4. Храмов Д. А. Использование пакета символьных вычислений Maple для моделирования динамики космической тросовой системы со сферическим шарниром // Системные технологии. — 2004. — № 3 (32) — с. 110–116. ([Электронный ресурс] — Режим доступа : http://dkhramov.dp.ua/uploads/Sci/TSGS/ST_khramov_maple.tif)
5. Волошенюк О. Л., Пироженко А. В. Модель процессов стабилизации движения концевых тел вращающейся космической тросовой системы в наземных экспериментах//Техническая механика.—2010—№ 3, с. 106–116.
6. Храмов Д. А. Особенности моделирования динамики экспериментальной тросовой системы современными компьютерными программами // Техническая механика. — 2011. — № 3, с. 91–102. ([Электронный ресурс] — Режим доступа:

 ISSN 1562-9945

http://dkhramov.dp.ua/uploads/Sci/HomePage/dkhramov.doc)
7. High-Performance Physical Modeling and Simulation [Электронный ресурс] : веб-страница. — Режим доступа : http://www.maplesoft.com/products/maplesim/
8. Octave [Электронный ресурс] : веб-сайт. — Режим доступа : http://www.gnu.org/software/octave/
9. Scilab [Электронный ресурс] : веб-сайт. — Режим доступа: http://www.scilab.org/
10. Тестирование быстродействия MATLAB и GNU/Octave [Электронный ресурс] : веб-страница. — Режим доступа : http://mydebianblog.blogspot.com/2010/10/matlab-gnuoctave.html
11. Алексеев Е. Р. Scilab: Решение инженерных и математических задач / Е. Р. Алексеев, Е. А. Чеснокова, Е. А. Рудченко. — М. : ALT Linux ; Бином. Лаборатория знаний, 2008. — 260 с. ([Электронный ресурс] — Режим доступа : http://www.altlinux.org/Books: Altlibrary/scilab)
12. Langtangen H. P. Python Scripting for Computational Science / H. P. Langtangen. — 3rd Edition — Berlin, Heidelberg : Springer, 2009. — 750 p.
13. Бизли Д. Python. Подробный справочник / Д. Бизли — СПб. : Символ-Плюс, 2010. — 864 с.
14. TIOBE Programming Community Index for December 2011 [Электронный ресурс] : веб-страница. — Режим доступа : http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
15. Pythonxy [Электронный ресурс] : веб-сайт. — Режим доступа : http://code.google.com/p/pythonxy/
16. Python(x, y) [Электронный ресурс] : веб-страница. — Режим доступа : http://habrahabr.ru/blogs/python/113065/
17. NumPy [Электронный ресурс] : веб-сайт. — Режим доступа : http://numpy.scipy.org/
18. NumPy for Matlab Users [Электронный ресурс] : веб-страница. — Режим доступа : http://www.scipy.org/NumPy_for_Matlab_Users
19. NumPy, пособие для новичков. Часть 1 [Электронный ресурс] : веб-страница. — Режим доступа : http://habrahabr.ru/blogs/python/121031/
20. SciPy [Электронный ресурс] : веб-сайт. — Режим доступа : http://www.scipy.org/
21. Matplotlib [Электронный ресурс] : веб-сайт. — Режим доступа : http://matplotlib.sourceforge.net/
22. Dkhramov.dp.ua [Электронный ресурс]. — Режим доступа : http://dkhramov.dp.ua/uploads/Sci/HomePage/tb.rar
23. Шампайн Л. Ф. Решение обыкновенных дифференциальных уравнений с использованием MATLAB / Л. Ф. Шампайн, И. Гладвел, С. Томпсон. — СПб.: Издательство «Лань», 2009. — 304 с.
24. PerformancePython [Электронный ресурс] : веб-страница. — Режим доступа : http://www.scipy.org/PerformancePython
25. Cython: C-Extensions for Python [Электронный ресурс] : веб-сайт. — Режим доступа : http://cython.org/
26. SymPy [Электронный ресурс] : веб-сайт. — Режим доступа : http://sympy.org/
27. Sage [Электронный ресурс] : веб-сайт. — Режим доступа : http://www.sagemath.org/
28. JModelica.org [Электронный ресурс] : веб-сайт. — Режим доступа : http://www.jmodelica.org/