

Свободная матричная система FreeMat

Владимир Дьяконов,
д. т. н., профессор
vpdyak@yandex.ru

При расчетах и проектировании радиоэлектронных компонентов и устройств необходимы матричные системы компьютерной математики с полной поддержкой аппарата комплексных вычислений. Лидером мирового рынка стала коммерческая система MATLAB. Однако ее применение сдерживается большим объемом этой системы (более 5 Гбайт) и высокой стоимостью. В статье описана свободно распространяемая и бесплатная система FreeMat, способная выполнять такие вычисления и имеющая подобный MATLAB графический интерфейс пользователя при высокой степени графической визуализации вычислений.

Введение

FreeMat — свободная матричная система для инженерных и научных расчетов и обработки данных. Система имеет инструментарий Qt, обеспечивающий кросс-платформенность этого продукта. FreeMat использует средства объектно-ориентированного программирования и имеет интерфейс к программным модулям, написанным на языках программирования C, C++ и Fortran.

Размер установочных файлов системы на два порядка меньше, чем у системы MATLAB. Поэтому о полной замене куда более мощной системы MATLAB речь не идет. Система FreeMat может заменить MATLAB как средство для проведения основных матричных вычислений с комплексными данными, при разработке методов и моделей компонентов математического моделирования электронных цепей и устройств, моделировании и визуализации электрических сигналов, проведении аппроксимации нелинейной зависимости, например вольт-амперных характеристик активных приборов, выполнении быстрого преобразования Фурье и т. д.

FreeMat имеет интерпретирующий язык программирования и диалога, подобный применяемому в коммерческой матричной системе MATLAB фирмы The Math works и IDL Research System. FreeMat создана по лицензии GPL, предусматривающей свободное развитие и распространение этой системы. В ней есть полноценная IDE с возможностью создания и отладки кода, сохранения и выполнения истории команд, управления файлами. FreeMat — чисто численная система. Те, кому необходимы символьные (аналитические) вычисления, могут познакомиться с открытой системой компьютерной алгебры wxMaxima.

Последняя версия FreeMat 4.2 была выпущена осенью 2012 года [1] и доступна практически для всех известных платформ. Среди ее основных возможностей:

- Предельно простой графический интерфейс пользователя.
- Поддержка объектно-ориентированного программирования и перегрузки операторов.
- Поддержка автодополнения в консоли.
- Поддержка параллельных вычислений на базе технологии MPI.
- Поддержка OpenGL для построения графиков и поверхностей.
- Предоставление большинства известных методов численных вычислений, включая прямое и обратное быстрое преобразование Фурье (FFT), преобразование Гильберта; SVD-, LU- и QR-разложения матриц, работу с многомерными матрицами, интерполяцию и аппроксимацию функций и многое другое.

Система имеет обширное справочное руководство в 2526 страниц (файл формата PDF), подготовленное главным разработчиком системы — Самитом Басу [2]. Целью FreeMat было создание современного инструментария для реализации матричных численных методов и их визуализации при простом интерфейсе для внешних программ на C/C++/Fortran.

Интерфейс FreeMat для компьютеров Apple

Интерфейс пользователя системы FreeMat для компьютеров Apple (рис. 1) на первый взгляд напоминает упрощенный интерфейс матричной системы MATLAB (исключая последнюю реализацию MATLAB R2012b, которая имеет совершенно новый интерфейс [3]). На рис. 1 представлено окно командного режима (в левой части окна) и окно справочной системы — справки (в правой части окна). Все окна плавающие. В окне командного

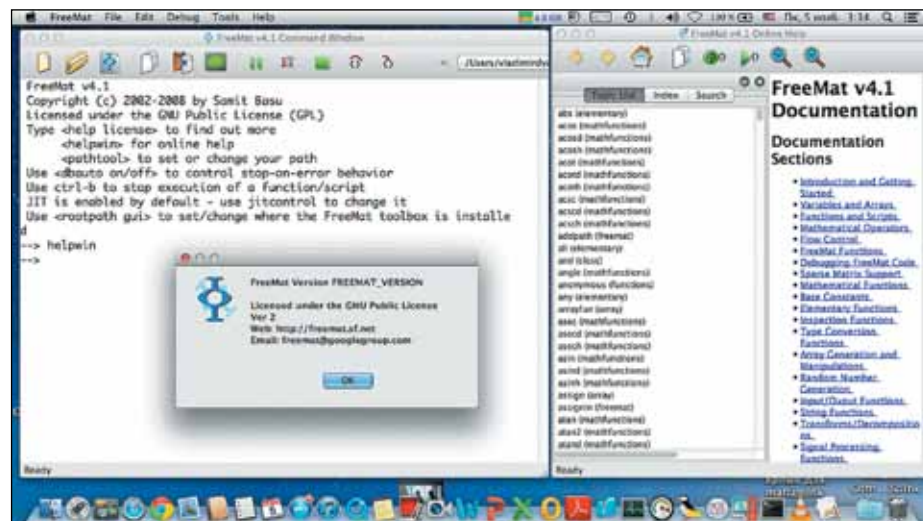


Рис. 1. Интерфейс системы FreeMat: окно командного режима (слева) и окно справки (справа)

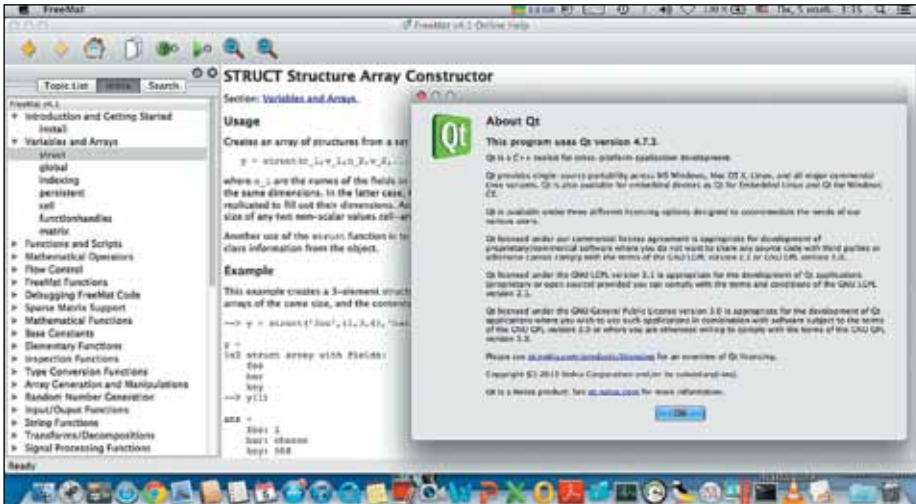


Рис. 2. Окно раздела Index справки

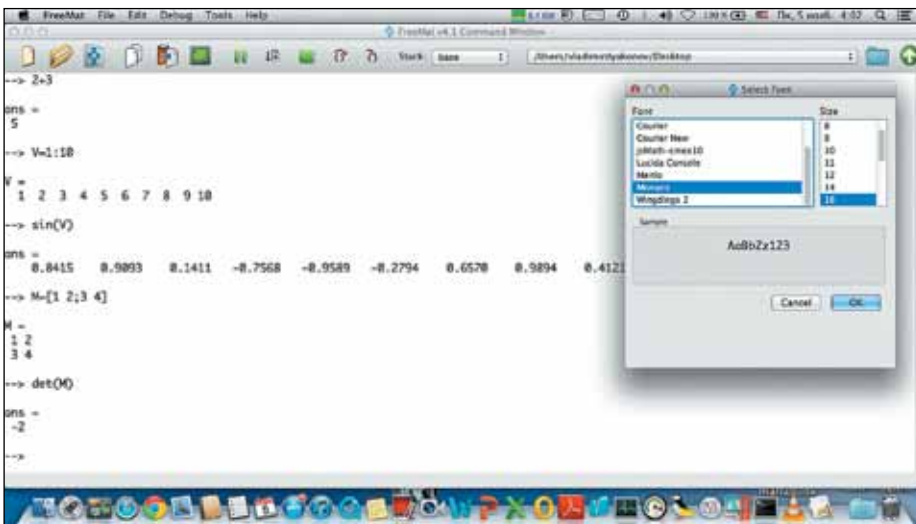


Рис. 3. Примеры простейших вычислений в командном окне и окно выбора шрифтов (внутри)

режима можно увидеть начальные данные о системе FreeMat. Окна имеют титульную строку с тремя цветными кружками в начале.

С помощью этих строк можно закрыть окно, перевести его в полноэкранный и масштабируемый режимы работы.

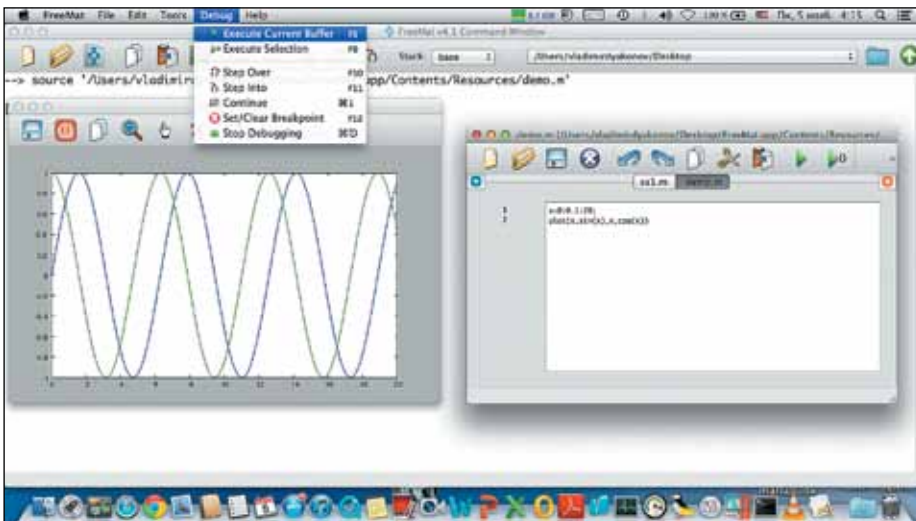


Рис. 4. Простая script-программа в редакторе FreeMat и графическое окно с результатом ее исполнения

Окно раздела **Index** показано на рис. 2. Внутри окна справки есть окно About Qt. Справочная информация содержит гиперссылки с названиями разделов и организована по аналогии с гипертекстовой справкой системы MATLAB.

Примеры простейших вычислений в командном окне даны на рис. 3. Внутри этого окна показано окно выбора шрифтов, позволяющее выбрать тип и размер шрифта, который используется в окне командного режима. Оно вызывается командой **Preferences-Font** в позиции **Edit** меню.

Язык программирования и диалога в системе FreeMat является объектно-ориентированным интерпретатором. Он детально описан в [2, 3], причем большая часть [2] посвящена описанию средств визуализации объектов языка. Он позволяет в командном окне набрать любое выражение или набор законченных инструкций и тут же их исполнить. Такой набор можно представить в виде script-файла, для создания которого используется встроенный редактор FreeMat.

На рис. 4 приведено окно редактора с простой программой, запущенной на выполнение активизацией кнопки с зеленым треугольником на панели инструментов редактора. Тут же показан результат исполнения программы — окно с графиками функций синуса и косинуса.

Вообще интерфейс FreeMat предельно прост. Он проще интерфейса ранних версий MATLAB [3–7]. Меню содержит всего шесть позиций, в каждой из которых число команд минимизировано:

- **FreeMat** — вывод данных о системе и средствах Qt, доступные сервисы, управление показом окон и выход из программы.
- **File** — открытие нового файла, ввод файлов и запись script-файлов.
- **Edit** — копирование и вставка выделенного в командном окне текста и команда Preferences с двумя позициями: выбора шрифтов и максимального числа линий в командном окне.
- **Debug** — пауза, возобновление работы, остановка и управление шагом исполнения программы.
- **Tools** — открытие окна инструментальных средств (редактора, установки рабочей директории, браузера файловой системы, истории, рабочего пространства и отладчика программ).
- **Help** — вызов окна справки.

Для выделения нужно нажать левую клавишу мыши, а затем использовать команду **Copy** в позиции меню **Edit** для переноса скопированного в буфер. Меню правой клавиши мыши в системе FreeMat нет.

Интерфейс системы FreeMat для компьютеров класса PC

Для компьютеров класса PC с операционной системой Windows последней версией

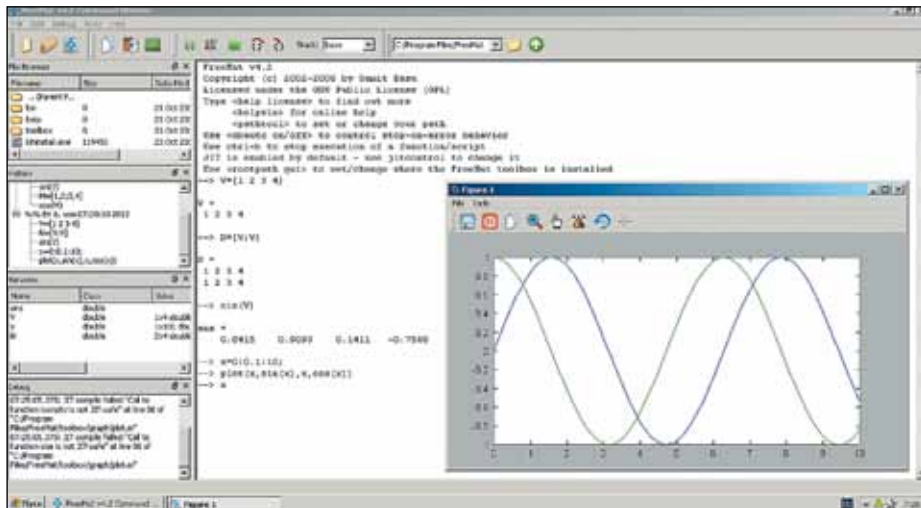


Рис. 5. Интерфейс системы FreeMat 4.2 для компьютеров класса PC с операционной системой Windows

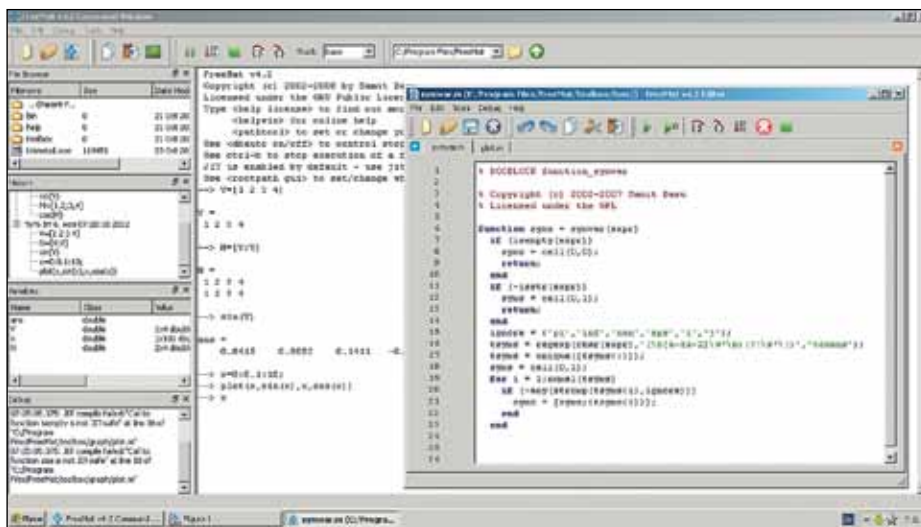


Рис. 6. Окно редактора с листингом функции

является FreeMat 4.2. Окно этой системы еще более напоминает окно системы MATLAB (до версии R2012b) (рис. 5). Окна менеджера файлов, истории команд, переменных и отладчика располагаются внутри окна командного режима. Пункта FreeMat в меню нет, а его позиции перешли в позицию Help. Иной вид имеет титульная строка: управляющие окном кнопки переместились в конец титульной строки и имеют привычный для пользователей Windows (до Windows 8) вид прямоугольников.

Файловая система FreeMat более открытая, чем у версии для MAC. Окно выбора загружаемого файла открывает сразу доступ к директории FreeMat с папками bin (системные файлы), help (файлы справки) и toolbox (папки с файлами функций объектов). На рис. 6 показано командное окно, в котором видно окно редактора с листингом одной из функций. Он демонстрирует второй тип программ — функции, имеющие список входных параметров и специальный формат задания.

Примеры на рис. 5 и 6 показывают, что работа в системе FreeMat для PC ничем не отличается от таковой для компьютеров Apple. Аналогична и справочная система. Поэтому последующий материал статьи относится к обоим описываемым версиям системы FreeMat.

Матричные вычисления

Работу в системе FreeMat следует начинать с уточнения рабочей директории. Команда `pathtool` выводит окно установки директории, которая должна содержать исполняемые операторы и функции. Рекомендуется сразу установить рабочую директорию.

FreeMat имеет ряд констант:

- Pi = 3,1416;
- i и j — мнимые единицы;
- eps = 2,2204×10⁻¹⁶ — погрешность численных методов;
- e = 2,7183 — основание натурального логарифма;
- inf — бесконечность;

- NaN — не цифровой результат (например 0/0);
- ans — последний результат выражения и др.

Набор констант не принципиально отличается от принятого в системе MATLAB.

Переменная в системе FreeMat — это область памяти, имеющая имя (идентификатор) и хранящая присвоенные ей численные значения. Как в MATLAB, даже переменная для хранения одного числа рассматривается как матрица, имеющая размер 1×1. Для присваивания значения переменным используется знак равенства. FreeMat различает регистр, поэтому `x` и `X` — это разные переменные. Набор типов переменных — тот же, что и в MATLAB.

Векторы задаются в квадратных скобках, и для разделения их элементов используется пробел или запятая. Точка с запятой используется как разделитель элементов столбцов. Векторы могут быть двух типов: вектор-строка и вектор-столбец. Примеры задания простых векторов и матриц мы уже привели на рисунках. Точка с запятой в конце строки блокирует выдачу результата вычислений.

Матрицы могут быть двухмерными и многомерными. С матрицами возможны различные арифметические операции с применением операторов «+», «-», «*» и «/». Если нужно обеспечить почленную операцию, перед ее обозначением ставится точка. Одинарная кавычка справа от матрицы означает ее транспонирование:

```
--> M=[1; 2; 3; 4]
M = 1
    2
    3
    4
--> M'
ans = 1 3
     2 4
```

Обращение матрицы задается ее возведением в степень (-1), как `M^(-1)`, или применением функции `inv`:

```
--> inv(M)
ans = -2.0000    1.0000
       1.5000   -0.5000
```

С помощью функции можно решить систему линейных уравнений (в том числе с комплексными коэффициентами, например уравнений Кирхгофа для электрических цепей).

Для вычисления детерминанта квадратной матрицы используется функция `det`:

```
--> det(M)
ans = -2
```

Имеется и множество других матричных функций и функций линейной алгебры. При математическом моделировании важное место занимают разреженные массивы и матрицы. У них большая часть элементов имеет нулевое значение. Это позволяет уменьшить время выполнения матричных

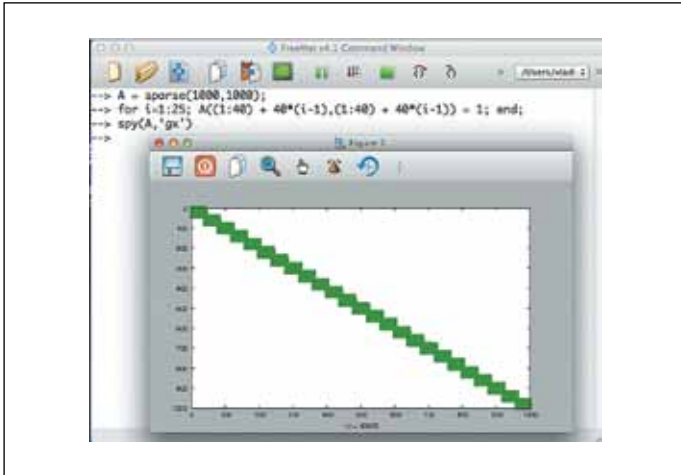


Рис. 7. Задание и визуализация разреженной матрицы

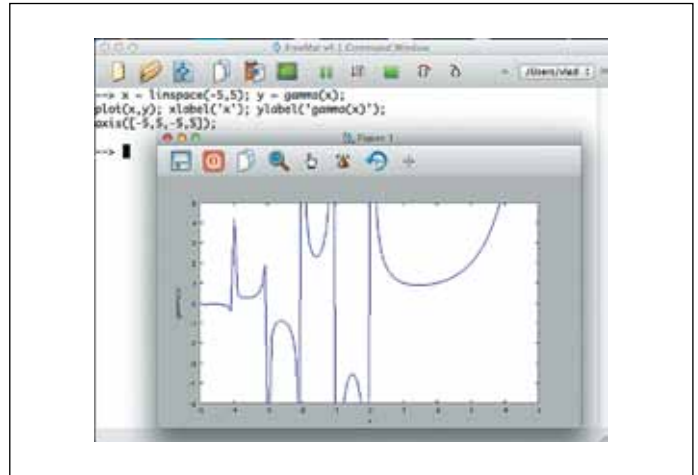


Рис. 8. График гамма-функции

операций. FreeMat имеет восемь операций с разреженными матрицами. Одна из них — визуализация созданной разреженной матрицы — показана на рис. 7.

Операторы и функции

FreeMat имеет полный набор арифметических операций и математических функций, присущий современным системам компьютерной математики. Ввиду его общеизвестности [6, 7] и громоздкости описания мы не будем описывать его в деталях. Впрочем, небольшие тонкости тут есть, например отсутствие функции натурального логарифма **ln**: она обозначена как **log**:

```
--> log(2)
ans = 0.6931
```

Зато есть функция логарифма по основанию 10 — **log10** и по основанию 2 — **log2**. Последний символ **N** указывает на отношение функции к классу гиперболических функций, **D** означает, что угол измеряется в градусах (degrees). А буква **A** (от arc) указывает на обратную функцию. Это говорит лишь о необходимости ознакомиться с соответствующим разделом справочной системы FreeMat.

Визуализация элементарных функций осуществляется простейшей графической функцией **plot**. На рис. 4 приведены примеры для тригонометрических функций синуса и косинуса. А вот похвастать обширным набором специальных математических функций FreeMat не может: их заметно меньше, чем в системах компьютерной алгебры Maple и Mathematica. И они размещены в общем разделе Mathematical Functions. Имеются гамма-функция (рис. 8), логарифмическая гамма-функция, полиномы Лежандра, функция ошибок. Все эти функции широко применяются в практике расчетов радиоэлектронных устройств.

В системе FreeMat элементарными называется большая группа совершенно иных функций. В их числе, например, функция ап-

проксимации производной конечными разностями:

```
y=diff(x,k,dim)
```

где **x** — массив исходных данных, **k** — порядок производной. Эта функция позволяет строить графики производной функции, например затухающей синусоиды (рис. 9):

```
x=0:0.05:10; S=e.^(-x/2).*sin(x);
D=diff(S); plot(x/0.05,S,D/0.05)
```

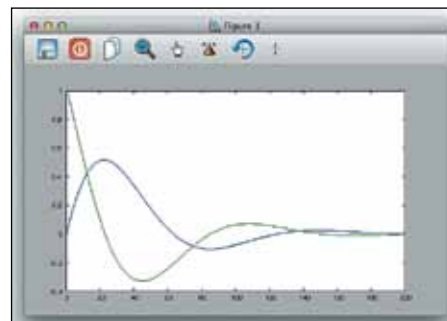


Рис. 9. Графики затухающей синусоиды и ее производной

Обратите внимание на то, что по горизонтальной оси отложено не реальное время, а номера отсчетов функции. Задать свои функции можно рядом способов, например как объект, с помощью функции **inline**:

```
--> a=inline('x*sin(x)')
a =
inline function object
f(x) = x*sin(x)
--> a(1)
ans = 0.8415
--> sin(1)
ans = 0.8415
```

Численные методы

FreeMat ориентирована на разработку, а не на применение уже готовых численных

методов. Поэтому число последних резко сокращено, чтобы минимизировать размер кодов системы. В разделе численных методов справки FreeMat есть только две функции численного интегрирования методом трапеций и одна функция **ode45** — для решения дифференциального уравнения первой степени методом Рунге-Кутты с заданными граничными условиями. Например, для уравнения разряда конденсатора **RC**, равном 1, дифференциальное уравнение будет таким: $y' = -y$ (y — производная). Тогда, при $y(0) = 1$ (начальное напряжение на конденсаторе) решение задается как:

```
--> [t,y]=ode45(@(-y),[0 3],1);
```

Если убрать точку с запятой в конце этого программного выражения, то будут выведены вектора **t** и **y** решения в заданном интервале изменения **t** от 0 до 3. Можно построить и график решения — это будет экспонента разряда:

```
--> plot(t,y,'o-')
```

Пример интегрирования методом трапеций:

```
--> x=[1 2 3 4];
--> trapz(x)
ans = 7.5000
```

Эти примеры элементарны. Однако функций для реализации основных численных методов линейной алгебры, работы с полиномами, интерполяции, аппроксимации, оптимизации и др. представлено много, и это именно те функции, которые широко применяются в практике электротехнических расчетов [5, 6]. Все расчеты ведутся с помощью комплексных чисел и данных, широко применяемых в практике таких расчетов. Приведем лишь несколько примеров.

Например, по вектору коэффициентов полинома $A(x) = 1x^3 - 2x^2 - 3x - 4$ можно найти его корни с помощью функции `roots(A)`:

```
--> roots([1 -2 -3 -4])
ans = 3.2843 + 0.0000i
      -0.6421 + 0.8975i
      -0.6421 - 0.8975i
```

Следующая программа, созданная в редакторе, дает аппроксимацию функции косинуса с функцией Гаусса (рис. 10):

```
x = linspace(0,5,50);
y = x.*e.^(-x)
a = polyfit(x,y,6)
f = polyval(a,x);
plot(x,y,'r-',x,f,'g*');
```

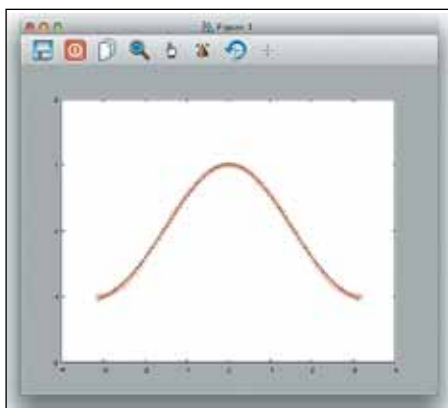


Рис. 10. Приближение функции косинуса функцией Гаусса

А другая программа обеспечивает полиномиальную аппроксимацию заданной во второй строке зависимости, используя полином 6-й степени (рис. 11):

```
x = linspace(0,5,50);
y = x.*e.^(-x)
a = polyfit(x,y,6)
f = polyval(a,x);
plot(x,y,'r-',x,f,'g*');
```

Используемая здесь функция имеет падающий участок, как вольт-амперная характеристика туннельного диода или λ -диода и λ -транзистора. С помощью приведенной программы можно легко выяснить, какого порядка нужно использовать полином для

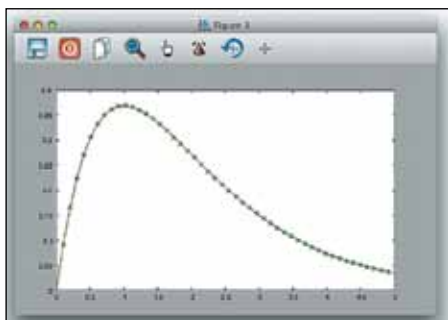


Рис. 11. Пример приближения полиномом 6-й степени функции с падающим участком

полиномиальной аппроксимации этих приборов и какую точность обеспечивает такое приближение.

Примеры на быстрое преобразование Фурье

В числе функций преобразования имеются функции прямого (`fft`) и обратного (`ifft`) преобразования Фурье. Они находят очень широкое применение в обработке сигналов в радиотехнике и технике связи. После вначале прямого, а затем обратного преобразования вектора он восстанавливается с указанием мнимых частей элементов выходного вектора (даже если они нулевые):

```
--> x=[1 2 3 4];
--> X=fft(x)
X = Columns 1 to 2
      10.0000 + 0.0000i -2.0000 + 2.0000i
      Columns 3 to 4 -2.0000 + 0.0000i -2.0000 - 2.0000i
--> x=ifft(X)
x = Columns 1 to 2
      1.0000 + 0.0000i 2.0000 + 0.0000i
      Columns 3 to 4 3.0000 + 0.0000i 4.0000 + 0.0000i
```

Прямое преобразование Фурье обычно используется для представления сигналов, описанных функцией времени, в частотной области. Ниже дан фрагмент программы, имитирующей создание 256 точек сигнала $y(t)$:

```
t=0:0.01:2.56;
y = sin(2*2*pi*t)+0.5*sin(5*2*pi*t);
```

Установим генератор случайных чисел с нормальным распределением и смоделируем зашумленный сигнал, добавив в вектор y случайную компоненту:

```
randn('state',0);
yn=y+2*randn(size(t));
```

Теперь зададим функцией `figure(1)` первое графическое окно и зададим в нем построение функций `plot` и зашумленного сигнала:

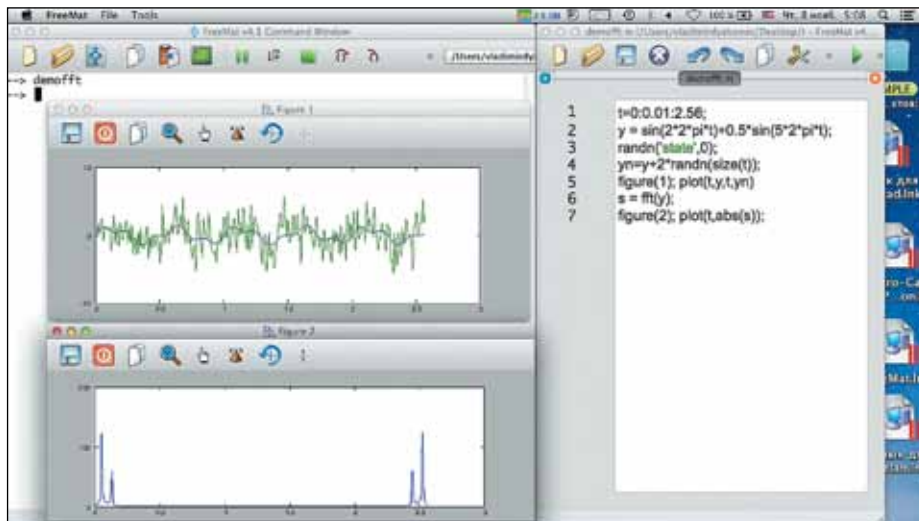


Рис. 12. Пример создания файла demofft.m и исполнения команды demofft

```
figure(1); plot(t,y,t,yn)
```

Эти виды зависимости соответствуют временному представлению сигналов. Переведем сигнал в частотную область, используя функцию быстрого преобразования Фурье — `fft`:

```
s = fft(y);
```

Теперь можно построить график спектра сигнала и представить его в частотной области:

```
figure(2); plot(t,abs(s));
```

Соберем эти программные предложения воедино в редактор программного кода FreeMat и запишем как файл `demofft.m` (рис. 12). На этом рисунке показано и командное окно FreeMat с результатом выполнения команды `demofft`.

Результат представлен двумя графическими окнами. На первом показаны графики исходного и зашумленного сигнала. Нетрудно заметить, что форма зашумленного сигнала не имеет почти ничего общего с исходным сигналом, поскольку шум практически его забивает. Определить полезные составляющие спектра сигнала в частотной области четко выделяет обе синусоидальные составляющие сигнала и позволяет оценить их частоту и амплитуду.

Исполнение FreeMat-файла в среде MATLAB

А теперь проверим, будет ли исполняться созданный во FreeMat файл в среде системы MATLAB, к применению которой привыкли специалисты по электротехнике и радиотехнике. Запустим эту систему и загрузим файл

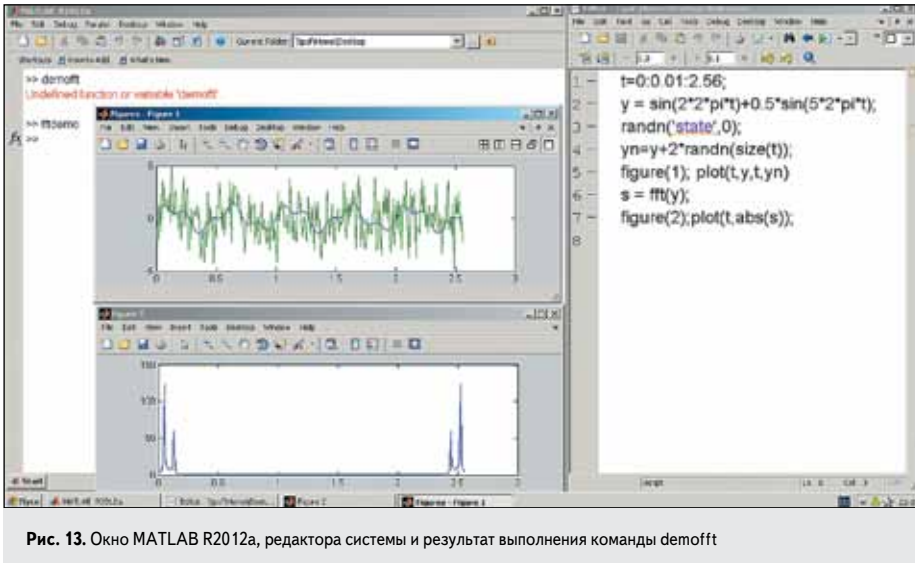


Рис. 13. Окно MATLAB R2012a, редактора системы и результат выполнения команды demofft

demofft.m в ее редактор. На рис. 13 показано окно MATLAB с окном редактора и результатом выполнения команды demofft.

Как видно, в первый раз система MATLAB оказалась выполнять команду, поскольку не нашла файла с ее именем в своей рабочей дирек-

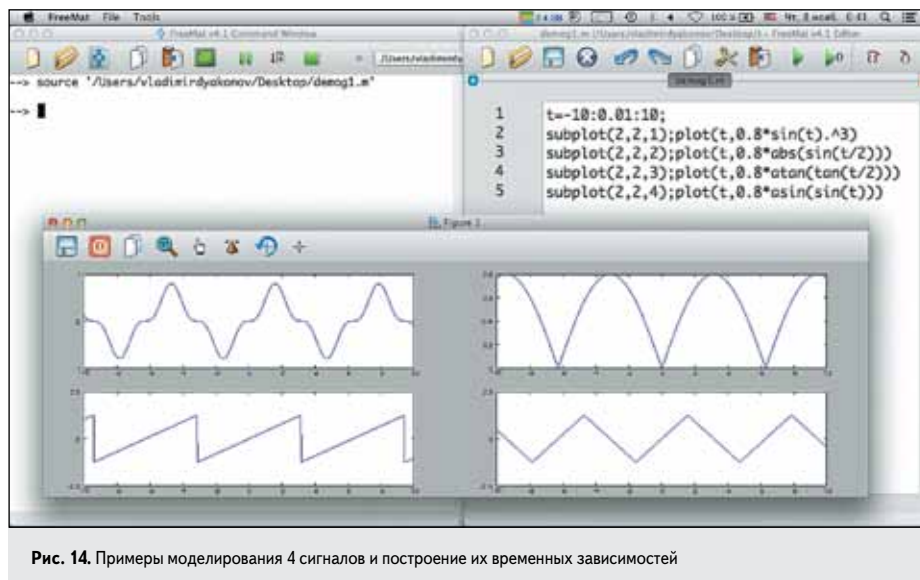


Рис. 14. Примеры моделирования 4 сигналов и построение их временных зависимостей

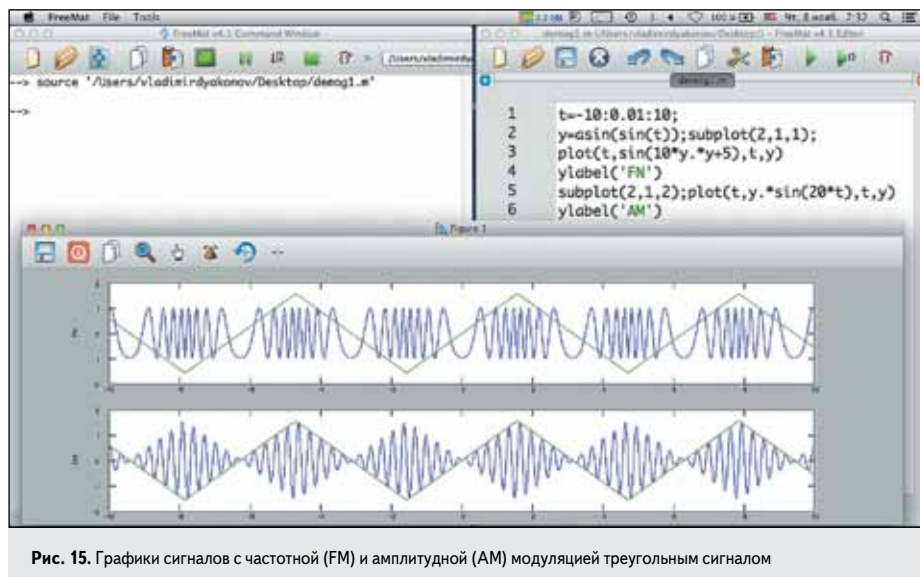


Рис. 15. Графики сигналов с частотной (FM) и амплитудной (AM) модуляцией треугольным сигналом

тории. После установки нужной директории команда была исполнена и получен результат, полностью повторяющий полученный ранее (сравните рис. 12 и 13). Сравните также интерфейсы пользователя FreeMat и MATLAB: они практически одинаковы. Поэтому пользователь, освоивший работу в системе FreeMat, при необходимости легко перейдет к работе в более мощной и дорогой системе MATLAB.

Графическая визуализация в системе FreeMat

Визуализация с помощью 2D-графики широко используется в технике представления различных сигналов. Некоторые идеализированные периодические сигналы можно представить как комбинацию из тригонометрических и обратных тригонометрических функций (рис. 14). Здесь показано построение четырех графиков в одном окне.

Важным достоинством такого представления сигналов является однозначность и хорошее представление периодичности в пределах изменения времени от $-\infty$ до $+\infty$. Это очень важно при имитационном моделировании сигналов. Подобная техника применяется и при графической визуализации сложных сигналов. Для примера на рис. 15 представлена визуализация частотной (FM) и амплитудной (AM) модуляции.

Графики функций комплексного переменного Z и случайных чисел (функция rand) на плоскости (рис. 16) строятся следующим скрипт-файлом, представляющим четыре окна от figure1 до figure4:

```
figure(1)
x = linspace(-1,1,512)*ones(1,512); y = x';
Z = exp(-(x.^2+y.^2)/0.3); image(Z);
colormap(copper);
figure(2)
x = rand(512);
x((-64:63)+256,(-128:127)+256) = 1.0;
image(x); colormap(gray)
figure(3)
x = linspace(-1,1,512)*ones(1,512);
y = x'; Z = exp(-(x.^2+y.^2)/0.3);
image(Z);
figure(4)
x = linspace(-1,1,512)*ones(1,512);
y = x'; Z = exp(-(x.^2+y.^2)/0.3);
image(Z); colormap(gray);
```

Такой способ позволяет строить разные графики с индивидуальной установкой цветов в каждом из них. В прямых апострофах можно в строковом виде задавать различные опции, меняя стиль и цвет линий, вводя титульные и иные надписи и т. д. Следующий пакет скрипт-файлов иллюстрирует эти возможности (рис. 17):

```
figure(1); x = linspace(0.4,512);
y = e.^x; semilogy(x,y,x,2.*y,'r-');
title('Graphic with y log.')
figure(2); x = linspace(0.1,10,4);
y = x; loglog(x,y,x,2.*y,'y-');
title('Graphic with loglog')
figure(3); x = linspace(-5,5,500);
t = exp(-x.^2); y = t.*cos(2*pi*x.^3);
plot(x,y); hold on; plot(x,t,'g','x',t,'b-')
figure(4); [x,y] = meshgrid([-2.25:2]);
z = x.*exp(-x.^2-y.^2); contour3(x,y,z,30);
axis square; view(-15,25)
```

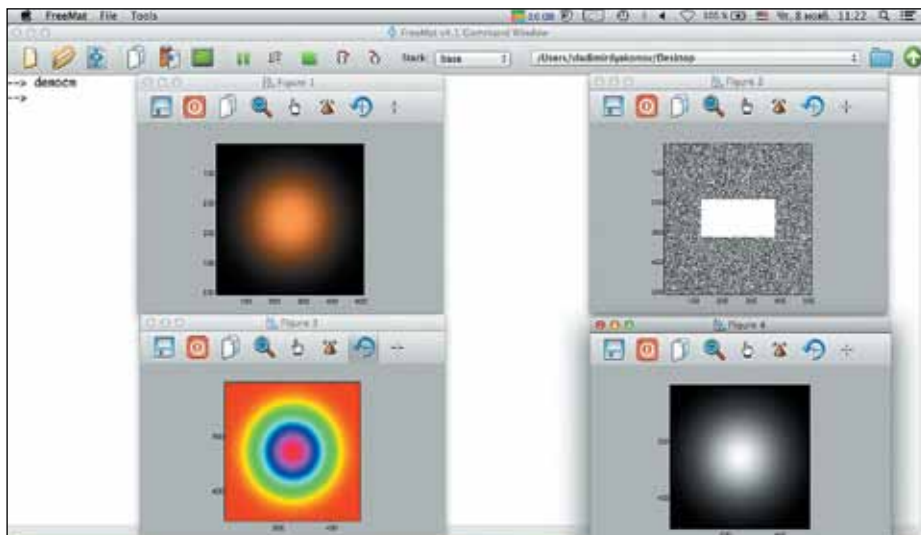



Рис. 16. Различные графики на плоскости

Построение еще четырех 3D-графиков (рис. 18) обеспечивает следующая script-программа:

```
figure(1); t=0:(2*pi/100):(2*pi);
x=cos(t^2).*(2+sin(t^3).^3); y=sin(t^2).*(2+sin(t^3).^3);
z=cos(t^3).^3; tubeplot(x,y,z,0.14*sin(t^5)+.29,t,10);
figure(2); x = repmat(linspace(-1,1),[100,1]);
y = x'; r = x.^2+y.^2; z = exp(-r^3).*cos(5*r);
c = r; surf(x,y,z,c); axis equal; view(3)
figure(3); [x,y] = meshgrid(linspace(-1,1,50));
z = x.*exp(-(x.^2+y.^2)); h = contour(z);
clabel(h,'backgroundcolor',[1,1,6],'edgecolor',[7,7,7]);
figure(4); x = repmat(linspace(-1,1),[100,1]);
y = x'; r = x.^2+y.^2; z = exp(-r^3).*cos(5*pi*r);
surf(x,y,z); axis equal; view(3)
```

Эти примеры показывают обширные возможности 2D- и 3D-графики системы FreeMat. Они вполне эквивалентны реализациям основных графических функций в системе MATLAB.

Заключение

В версиях 4.1/4.2 система FreeMat приобрела типичный для системы MATLAB (до последней версии MATLAB 2012 б, появившейся осенью 2012 года и кардинально переработанной) современный и вполне профессиональный графический интерфейс пользователя. В отличие от явно перенасыщенных позициями меню, панелями и кнопками интерфейсов программ Mathcad и Maple [7] интерфейс FreeMat предельно минимизирован и не содержит ничего лишнего. Это можно сказать и в отношении набора средств для математических вычислений. Основные операторы и функции FreeMat подобны применяемым в MATLAB [4–6], но они полностью открыты и бесплатны. Сравнить обе системы по их полным возможностям вряд ли есть смысл: установочный файл системы FreeMat занимает на жестком диске компьютера меньше 44 Мбайт, тогда как установочные файлы MATLAB занимают на два порядка больше — более 5 Гбайт.

При этом FreeMat — свободно распространяемая и бесплатная система, тогда как MATLAB

дорогая и более мощная система. FreeMat целесообразно использовать перед применением коммерческой системы MATLAB, например в процессе изучения основ компьютерного проектирования технических устройств и систем различного назначения, при начальной отработке численных методов вычислений и их графической визуализации.

Литература

1. <http://freemat.sourceforge.net>
2. Busu S. FreeMat v.4.1 Documentation. 10 июля 2011 г.
3. www.mathworks.com
4. Дьяконов В. П. MATLAB. Полное руководство. М.: ДМК-Пресс, 2012.
5. Дьяконов В. П. MATLAB и SIMULINK для радиоинженеров. М.: ДМК-Пресс, 2011.
6. Дьяконов В. П., Пеньков А. А. MATLAB и SIMULINK в электроэнергетике. Справочник. М.: Горячая линия – Телеком, 2009.
7. Дьяконов В. П. Энциклопедия компьютерной алгебры. М.: ДМК-Пресс, 2009.

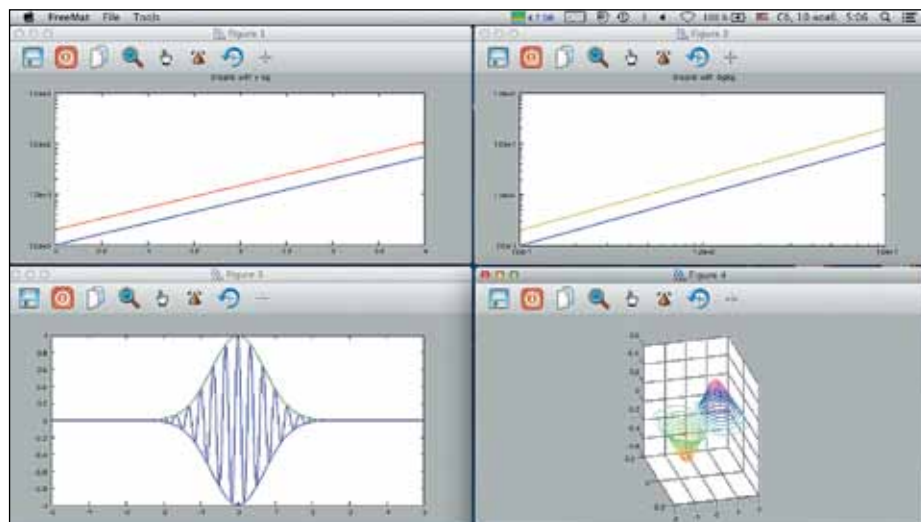


Рис. 17. Применение графических функций с опциями

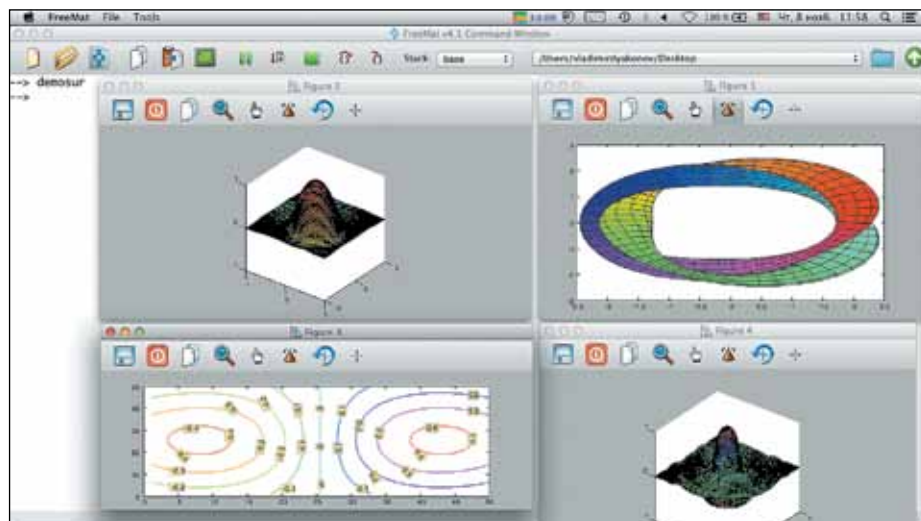


Рис. 18. Примеры построения 3D-графиков поверхностей и объемных фигур