

Математические пакеты

Введение в GNU Octave

Сучков Андрей Игоревич

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

5 сентября 2020 г.

Что есть математические пакеты?

- Пакет прикладных программ – набор взаимосвязанных модулей, предназначенных для решения задач определённого класса некоторой предметной области.
- Виды математических пакетов:
 - Для численных вычислений – MATLAB, *GNU Octave*, Scilab, Julia, *R* ...
 - Для символьных вычислений – Mathcad, Mathematica, Maple, Maxima, *SageMath* ...

- GNU Octave – свободная система для математических вычислений, использующая совместимый с MATLAB язык высокого уровня
- Разработан Джоном Итаном (John W. Eaton) в 1988 г.
- Написан на C/C++/Fortran, интерфейс на Qt
- Название дано в честь американского химика-технолога Октейва Левенспиля (Octave Levenspiel)

GNU Octave

Преимущества и недостатки

Преимущества:

- Высокая степень совместимости с MATLAB посредством реализации одинаковых функций
- Низкое потребление памяти (по сравнению с MATLAB)
- Свободная лицензия

Недостатки:

- Сложная документация
- Низкая скорость работы (вычислений) (по сравнению с MATLAB)

Правило

Векторизация – ключевая концепция языка!

$$f : \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_n) \end{bmatrix}$$

Следствие

Использование циклов только там, где это оправдано!

Философия GNU Octave

Пример

Листинг 1: Итерационное вычисление

```
1 x = 1:10_000_000;  
2 tic # time begin  
3   for i = 1:length (x)  
4     y(i) = sqrt (x(i));  
5   endfor  
6 toc # time end  
7 ## Elapsed time is 552.141 seconds.
```

Листинг 2: Векторизованное вычисление

```
1 x = 1:10_000_000;  
2 tic # time begin  
3   y = sqrt (x);  
4 toc # time end  
5 ## Elapsed time is 0.18814 seconds.
```

Представление данных

- В Octave всякая переменная представляется в виде матрицы (или массива)
- Octave сам определяет тип переменной, в зависимости от того, что вы ей присвоили

Листинг 3: Примеры различных типов данных

```
1 x = -0.5;  # variable 'x' is assigned a value
2             # of -0.5 of type double
3
4 A = [1, 2; 3, 4];  # variable 'A' is assigned a matrix
5                     # of dimension 2x2 of type integer
6
7 str = "GNU Octave";  # variable 'str' is assigned
8                      # a 1x10 array of type char
```

Листинг 4: Однострочные комментарии

```
1 # Single line comment
2 # Another line comment
```

Листинг 5: Блочные комментарии

```
1 #{
2     Block Comments
3     Comments, comments...
4     Blah, blah, blah...
5 #}
6
7 ## But this way much better!
8 ## Hey! That's pretty good!
```


Синтаксис

Перенос формул

Листинг 6: Перенос выражения

```
1 x = long_variable_name ...      # comment one
2   + longer_variable_name ...comment two
3   - 42;                          # last comment
```

Листинг 7: Перенос в строках

```
1 str = "This text starts in the first line \  
2 and is continued in the second line.";
```

Листинг 8: Перенос в круглых скобках

```
1 if (fine_dining_destination == on_a_boat
2     || fine_dining_destination == on_a_train)
3     seuss (i, will, not, eat, them, sam, i, am, i,
4            will, not, eat, green, eggs, and, ham);
5 endif
```

Синтаксис

Присваивание и простейшие арифметические операции

Листинг 9: Присваивание

```
1 a = b + 1; c = a; # in MATLAB & Octave
2 c = a = b + 1;    # only Octave
```

Листинг 10: Арифметические операции

```
1 x = 5 + 4 * a / (3
2   - 9);
```

Листинг 11: «Синтаксический сахар»

```
1 x++;
2 --y;
3 a += 4 * x;
4 z /= 3 + t--;
```

Синтаксис

Возведение в степень и деление справа налево

Листинг 12: Возведение в степень

```
1 x = a ^ (-2); # in MATLAB & Octave
2 y = b ** 5;   # only Octave
```

Листинг 13: Деление справа налево

```
1 x = a / b;
2 x = a * b ^ (-1); # equivalent
3
4 x = a \ b;
5 x = a ^ (-1) * b; # equivalent
```

Листинг 14: «Синтаксический сахар»

```
1 x \= a;
2 y ^= 3;
3 z **= -0.5;
```

Синтаксис

Поэлементные арифметические операции

Листинг 15: Возведение матрицы в степень

```
1 A = [1, 2; 3, 4];  
2 B = A ^ 2;      # equivalent: B = A * A  
3 C = A .^ 2;     # equivalent: C_ij = (A_ij)^2
```

Листинг 16: «Синтаксический сахар»

```
1 A = [1, 2; 3, 4];  
2 A .^= 2;
```

Примечание

Операции «+» и «.+», а также «-» и «.-» эквивалентны.

Синтаксис

Логические операции

Листинг 17: Логические операции

```
1 if (5 < x && x < 8) # logical AND
2     printf ("Good!\n")
3 endif
4
5 if (x >= 3 || y < 0) # logical OR
6     printf ("Good!\n")
7 endif
```

Синтаксис

Поэлементные логические операции

Листинг 18: Поэлементные логические операции

```
1 ! x;      # logical NOT
2 x != y;   # logical NOT EQUAL
3 x == y;   # logical EQUAL
4 x < y;     # logical LESS
5 x <= y;    # logical LESS OR EQUAL
6 x > y;     # logical GREAT
7 x >= y;    # logical GREAT OR EQUAL
8 x & y;     # logical AND
9 x | y;     # logical OR
```

Системная переменная ans

Листинг 19: Принцип работы ans

```
1 >> 2 + 3
2 ans = 5
3 >> 2 * ans
4 ans = 10
5 >> x = ans ^ 2
6 x = 100
7 >> ans
8 ans = 10
9 >>
```

Замечание

Использование ans в собственных функциях и скриптах – дурной тон!

Комплексные числа

i (...), j (...), I (...), J (...) – функции, возвращающие матрицу из мнимых единиц ($\sqrt{-1}$)

Листинг 20: Представление комплексных чисел

```
1 z = 5 - 4i;
2
3 x = -2; y = 7;
4 w = x + y * 1i;
5
6 i = 10;
7 t = x + y * i; # NOT complex
8
9 iscomplex (w); # return TRUE
10 iscomplex (t); # return FALSE
11
12 x1 = real (z); # Re(z)
13 y1 = imag (z); # Im(z)
14 z1 = conj (z); # complex conjugate
```


Математические константы: π и e

Листинг 21: Математические константы

```
1 pi (2, 3); # create matrix 2x3 with pi
2 e (3); # create matrix 3x3 with e
3
4 sin (pi / 6); # return 0.5
5
6 e ^ (-x ^ 2 / 2);
7 exp (-x ^ 2 / 2); # better
```

Логические переменные

Листинг 22: Логические переменные

```
1 true (2, 3); # create matrix 2x3 with logical 'true'
2 false (3); # create matrix 3x3 with logical 'false'
3
4 true | false # is TRUE
5 true & false # is FALSE
6
7 logical (-3) # \
8 logical (10) # > is TRUE
9 logical (pi) # /
10
11 logical (0) # is FALSE
12
13 islogical (x);
14 isbool (x); # equivalent
```

Машинная бесконечность и неопределённость

Листинг 23: Машинные переменные

```
1 Inf (2, 3);    # create matrix 2x3 with positive infinity
2 NaN (3);      # create matrix 3x3 with NaN (Not-A-Number)
3
4 Inf == Inf    # is TRUE but this is ridiculous
5 NaN == NaN    # always FALSE
6
7 isinf (x);
8 isnan (x);
```

Справка в Octave

- Чтобы получить справку о функции, воспользуйтесь функцией `help`
- Чтобы найти функцию по некоторому слову, воспользуйтесь функцией `lookfor`

Листинг 24: Использование функций помощи

```
1 help sin      # get help text for sine
2 help ","      # get help text for comma
3 help comma    # equivalent
4
5 lookfor "fourier" # search for the string "fourier"
6                  # in the documentation
```

Создание векторов

- Для создания вектора с равноотстоящими значениями используется оператор двоеточие.
- Для создания вектора с произвольными значениями используются квадратные скобки. Значения в квадратных скобках разделяются запятыми или пробелами.
- **Важно:** нумерация элементов вектора начинается с **1**!
- Обращение к элементу вектора осуществляется в круглых скобках.

Создание векторов

Примеры

Листинг 25: Векторы

```
1 a = 1:2:5;      # create vector [1, 3, 5]
2 a = [1, 3, 5];  # equivalent
3
4 b = 1:10;       # equivalent 1:1:10
5
6 c = 1:2:4;      # create vector [1, 3]
7
8 a(1)           # get first element
9 a(end)         # get last element
10 a(end-1)      # get penultimate element
11
12 b(5:9)         # get 5th through 9th elements
13 b([4, 6, 7])   # get 4th, 6th and 7th elements
14 b(b > 5)       # get all elements which great 5
```

Матрицы

- Ввод элементов матрицы также осуществляется в квадратных скобках, при этом элементы строки отделяются друг от друга пробелом или запятой, а строки разделяются между собой точкой с запятой или переводом строки.
- **Важно:** нумерация элементов матрицы начинается с 1!
- Обратиться к элементу матрицы можно указав после имени матрицы, в круглых скобках, через запятую, номер строки и номер столбца, на пересечении которых элемент расположен.

Матрицы

Примеры

Листинг 26: Матрицы

```
1 A = [1, 2; 3, 4];  
2  
3 B = [1, 2, 3  
4      4, 5, 6];  
5  
6 B(1,1); # get 1  
7 B(1,:); # get [1, 2, 3]  
8 B(:,2); # get [2; 5]
```


Функции для создания матриц и векторов

Листинг 27: Создание матрицы и вектора

```
1 x = linspace (0, 5, 10); # return a row vector with N
2                             # linearly spaced elements
3
4 E = eye (3, 2); # return an identity matrix
5 A = ones (3); # return a matrix whose elements are all 1
6 B = zeros (2, 3); # return a matrix whose elements are all 0
7
8 C = diag (X, k); # return a diagonal matrix
9                 # with vector X on diagonal k
10
11 U = rand (3, 2) # return matrix with U(0, 1) elements
12 N = randn (2, 3) # return matrix with N(0, 1) elements
```

Индексирование (продолжение)

Листинг 28: Работа с индексами

```
1 x = 1:3;
2
3 x(2) = 5;          # now x = [1, 5, 3]
4 x(5) = -3;         # now x = [1, 5, 3, 0, -3]
5 x(3) = [];         # now x = [1, 5, 0, -3]
6                   # "[]" is empty matrix
7 x(2:end) = 10;      # x = [1, 10, 10, 10]
8 x = [x, x];        # duplicate vector
9
10 A = [1, 2; 3, 4];
11 B = A(:);          # B = [1; 3; 2; 4]
12
13 a = [10, 20];
14 I = [1, 1, 2; 2, 1, 2];
15 A = a(I);          # A = [10, 10, 20; 20, 10, 20]
16
17 B = 42(ones (3, 2));
```

Функции для работы с векторами и матрицами

Листинг 29: Работа с векторами и матрицами

```
1 n = length (x); # return length for vector x
2
3 [m, n] = size (A); # return the size of A
4 m = rows (A); # return the number of rows of A
5 n = columns (A); # return the number of columns of A
6
7 s = sum (A, k); # return sum by dimension k
8 ## Also prod
9 m = min (A, [], k); # return min by dimension k
10 ## Also max
```

Функции для работы с векторами и матрицами

Ещё полезные функции

Листинг 30: Полезные функции

```
1 A = repmat (M, m, n);    # repeat matrix or n-D array
2 A = reshape (M, m, n);   # return a matrix with
3                             # the specified dimensions
4 a = det (M);
5
6 M = M.';    # matrix transpose operator
7 M = M';     # conjugate transpose operator
```

Поэлементные операции с операндами несовпадающих размеров

Листинг 31: Автосогласование размеров

```
1 A = [1, 2, 3
2       2, 3, 4
3       3, 4, 5];
4 b = [100, 200, 300].';
5
6 A_add_b = A + b
7
8 >> A_add_b =
9
10      101      102      103
11      202      203      204
12      303      304      305
```