Mathematical programs

Matrix based languages – octave

Ferenc Wettl
Department of Algebra

BUDAPESTI MŰSZAKI
MATEMATIKA
ÉS GAZDASÁGTUDOMÁNYI
INTÉZET
EGYETEM

Matrix based languages

-  MATLAB® (matrix laboratory – The Language of Technical Computing)
  - commercial program
  - millions of users
  - numerical calculations, signal and picture analysis, communication, control, finance,...
  - http://www.mathworks.com
- free clones:
  -  octave: GNU GPL, www.gnu.org/software/octave/
  -  Scilab: CeCILL license (GPL compatible), www.scilab.org (main developer: INRIA, France)
  -  FreeMat: GPL, freemat.sourceforge.net/
- Similar software:
  - julia julia: MIT licensed, (high-performance dynamic programming language for technical computing), julialang.org/
  -  R: GPL, statistics

Calculation

- real and imaginary numbers
-
```
octave> 1 + 1
ans =  2

octave> 2^23
ans =  8388608

octave> 2^-3
ans =  0.12500

octave> 2^123
ans =    1.0634e+37

octave> (1 + 2i) / (3 - 1i)
ans =  0.10000 + 0.70000i

octave> (1 - 1i)^8
ans =  16
```

Calculation with matrices

- real and imaginary matrices
-
```
octave> [1 1; 3 3] + [1 2; 2 3]*[3 1; 2 1]^-1
ans =
   -2.00000    6.00000
   -1.00000   10.00000

octave> [1 1; 3 3] + [1 2; 2 3] * [2 1; 1 1]^-1
ans =
    3.3307e-16    4.0000e+00
    2.0000e+00    7.0000e+00

octave> [1; 2]
ans =
    1
    2
octave> ans'
ans =
    1    2
```

```
octave> zeros (2, 3)
ans =
   0   0   0
   0   0   0
octave> ones (1, 4)
ans =
   1   1   1   1

octave> eye (3)
ans =
Diagonal Matrix
   1   0   0
   0   1   0
   0   0   1
octave> diag ([2 3 1])
ans =
Diagonal Matrix
   2   0   0
   0   3   0
   0   0   1
```

Matrix division and inverse

- $\mathbf{A/B = C}$ meaning: $\mathbf{A = CB}$
- $\mathbf{A\backslash B = C}$ meaning: $\mathbf{B = AC}$
- Az $\mathbf{Ax = b}$ two ways to solve a system of equations:
  (1) $\mathbf{x = A\backslash b}$, (2) if the matrix is invertible $\mathbf{A^{-1}b}$
- Solve the followin system of equations! $\left\{\begin{array}{r} x - 3y = 15 \\ 4x + 2y = 18 \end{array}\right\}$
- ```
  octave> [1 -3; 4 2] \ [15; 18]
  ans =
     6
    -3

  octave> [1 -3; 4 2]^-1 * [15; 18]
  ans =
     6
    -3
  ```
  ```

- Matrix division is possible even if the matrix is not invertible (the theory of this will be covered later).
- Using this method we can always get the solution in the rowspace. Let us solve the following:

$$x + y + 2z = 2$$
$$2x + 2y + 3z = 4$$

```
octave> [1 1 2; 2 2 3] \ [4; 8]
ans =
   2.0000e+00
   2.0000e+00
   7.1054e-15

octave> rref ([1 1 2; 2 2 3])
ans =
   1   1   0
   0   0   1
```

Egészek

- Most calculations use double precision floating point numbers. Integers are only used for data.
- integer b biten, vagy unsigned integer: int8, uint8, int16, uint16, int32, uint32, int64, uint64.
- ```
  octave> 10 * rand (2, 3)
  ans =

     8.390913   0.500552   6.421794
     9.608188   1.873848   0.028212

  octave> int8 (ans)
  ans =

     8   1   6
    10   2   0
  ```
  ```

Ranges

- beginning:step:end
-
```
octave> 1:4
ans =
   1   2   3   4

octave> 4:1
ans = [](1x0)

octave> 9:-3:1
ans =
   9   6   3

octave> 1.1:.237:2.1
ans =
    1.1000    1.3370    1.5740    1.8110    2.0480
```

Variables

```
octave> a = 3
a =  3

octave> m = [
  1 2 a
  2 a 4]
m =
   1   2   3
   2   3   4

octave> m' * m
ans =
    5    8   11
    8   13   18
   11   18   25
```

Data types: typeinfo, scalar, matrix, range, string

- real and complex scalar and matrix, range, strings...
-
```
octave> typeinfo (a)
ans = scalar
octave> typeinfo (1.23)
ans = scalar

octave> typeinfo (m)
ans = matrix

octave> typeinfo ([1 + 2i 1 - 2i])
ans = complex matrix

octave> typeinfo (1:4)
ans = range

octave> typeinfo("something")
ans = string
```

Indexes

```
octave> M
M =
     8     1     6
    10     2     0

octave> M (1, 3)
ans =  6

octave> M (1, [2 3 1])
ans =
    1    6    8

octave> M (:, [2 3 1])
ans =
     1     6     8
     2     0    10
```

Functions

- ```
  octave> function f1
    1 + 1
  endfunction

  octave> f1
  ans =  2
  ```

- ```
  octave> function f2 (a, b)
    a^2 + b^2
  endfunction

  octave> f2 (3, 4)
  ans =  25
  ```

- ```
  function [aplusb, atimesb] = operator (a, b)
    aplusb = a + b;
    atimesb = a * b;
  endfunction
  octave> [c d] = operator (2, 3)
  c =  5
  d =  6
  ```

- General form of a function definition (red means optional):

  `function` *return value = function name (list of arguments)*

  octave commands

  `endfunction`
  ```

Logic values (1 = true, 0 = false, bool type)

```
octave> 4 > 1
ans =  1

octave> 4 < 1
ans = 0

octave> 4 == 1
ans = 0

octave> 4 >= 1
ans =  1

octave> (4 >= 1) == 1
ans =  1
octave> typeinfo (ans)
ans = bool

octave> a = true
a =  1
```

Conditional commands

- `if` (*condition*)

  octave commands

  `else` # *this part*

  octave commands # *is optional*

  `endif`

Conditional command example

```
function realorcomplex (a, b, c)
  d = b^2 - 4*a*c;  # ; silent calculation
  if (d >= 0)
    "real"
  else
    "complex"
  endif
endfunction

octave> realorcomplex (1, 1, 1)
ans = komplex

octave> realorcomplex (1,2,1)
ans = valos
```

Vectorization

- Our aim is to avoid repeating code and cycles using vector and matrix operations.
- This results in compact and more efficient code.
- Using the vector $(v_1, v_2, \ldots, v_n)$ create the following vector: $(v_2 - v_1, v_3 - v_2, \ldots, v_n - v_{n-1})$!

```
octave> l = [3 4 6 2 5 1]
l =
   3   4   6   2   5   1
octave> l(2:6) - l(1:5)
ans =
   1   2  -4   3  -4
```

- Apply the function $f(x) = x^2 + 3x + 1$ to a matrix!

```
octave> function func(a)
  a^2 + 3*a + 1
endfunction
octave> func( [1 1;0 2])
ans =
    5    7
    1   11
```

- Apply the function $f(x) = x^2 + 3x + 1$ to every element of a matrix!

```
octave> function func(a)
  a .^ 2 + 3 .* a + 1
endfunction
octave> func ([1:3 5])
ans =
    5   11   19   41
```

Questions

- What and how are the following commands used: `eye`, `ones`, `zeros`, `diag`?
- Create a random 4×5 matrix named `m` that contains integers in the range from 0 to 9!
- List some of the data types! How can we extract this information from a variable (or data)?
- Write a function with input x that returns $x + \frac{4}{x^2} + \frac{1}{x^3}$!
- Write the same function, but applied to a matrix, so that it applies the function to its every element!
- What is the result of the operation `1 == (2 > 4)` and `0 == (2 > 4)`? What is this data type named?
- Write a function that returns the sign of its input as a string!

Questions 2

- Create a 10×10 matrix, with the following block matrix form:

$$\begin{bmatrix} \mathbf{O}_5 & \mathbf{I}_5 \\ \mathbf{I}_5 & \mathbf{O}_5 \end{bmatrix}$$

- Solve the following system of equations using matrix division and inverse:

$$\begin{aligned} x - 2y &= 1 \\ 2x + y &= 7 \end{aligned}$$

- Calculate the solution of the following system of equations in its rowspace:

$$\begin{aligned} x - 2y + 3z &= -1 \\ 2x + y + z &= 3 \end{aligned}$$