Topic and contents

UNSW, School of Mathematics and Statistics

MATH2089 – Numerical Methods

Week 02 – Polynomials/Matlab functions

- MATLAB M-files
  - `horner.m`
  - `npdfex.m`
  - `npdf.m`

Polynomial evaluation

- Polynomial $p(x)$ of degree n

$$p(x) = \sum_{k=1}^{n+1} a_k x^{k-1} = a_1 + a_2 x + a_3 x^2 + \cdots + a_{n+1} x^n$$

  - e.g. $p(x) = 2 + 5x + 10x^2$ with vector coefficients $\mathbf{a} = (2, 5, 10)$
  - Only requires addition, multiplication, integer powers
- MATLAB function `polyeval.m`
- Function `polyeval` must be in file `polyeval.m`
- Comments at beginning document function: `help polyeval`
- Element by element multiplication `.*`
- Input vector/array $\Longrightarrow$ output vector/array of same size

Horner's Method

- Equivalent expression

$$p(x) = a_1 + x(a_2 + x(a_3 + x(a_4 + \cdots + x(a_n + x a_{n+1}))\cdots))$$

  - e.g $p(x) = 2 + x(5 + 10x)$
  - Evalute $p_1 = x \cdot 10 + 5$ first, then evaluate $p_2 = x \cdot p_1 + 2$
-

```
function p = horner(a, x)
n = length(a)-1;
p = a(n+1)*ones(size(x));
for k = n : -1 : 1
   p = x .* p + a(k);
end;
```


Flops

- Question: Which function is faster?
- `polyeval`
  - Simple implementation requires the evaluation of x^k ($k-1$ multiplications) for eack k, plus n multiplications and n additions:
  - Total of $2n + \sum_{k=1}^{n}(k-1) = \frac{(n-1)n}{2} + 2n = \frac{n^2}{2} + O(n)$ flops
- `horner` requires only n multiplications, n additions: $2n$ flops.

MATLAB – Built-in functions

- Built-in functions provided as executables
  - Functions that are frequently used
  - Functions that take less time
- Still have a `.m` file for documentation
  - Example: `exp`, `sqrt`,
  - MATLAB: `help exp`, `type exp.m`
- MATLAB elementary functions: `help elfun`
- Can check using `exist` command

Example (Built-in functions)

Find the value of $y = e^{\sqrt{709\pi}}$

MATLAB – Anonymous functions

- Anonymous functions are only available in current workspace
- MATLAB:
  `fnhandle = @(arglist) expression;`

Example (Standard Normal probability density function (PDF))

The standard (mean $\mu = 0$, standard deviation $\sigma = 1$) normal probability density function is

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}, \quad z \in \mathbb{R}.$$

- Anonymous function with one argument, may be a vector/array
  - Definition
    ```
    myfun = @(z) (1/sqrt(2*pi)) * exp(-z.^2/2);
    ```
  - Usage
    ```
    x = linspace(-5, 5, 201);
    f = myfun(x);
    plot(x, f);
    ```


MATLAB – Anonymous functions cont.

Example (Normal probability density function (PDF))

For a general mean $\mu \in \mathbb{R}$, standard deviation $\sigma > 0$ the normal probability density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad x \in \mathbb{R}.$$

- Anonymous function, more than one argument: M-file: `npdfex.m`

```
% MATH2089: File = npdfex.m
% Anonymous function for normal PDF
npdf = @(x, mu, sig) (1/(sig*sqrt(2*pi)))*exp(-(x-mu).^2/(2*sig^2))

% Usage to plot normal PDF
figure(1)
x = linspace(-10,10,1001);
f = npdf(x, 1, 3);
plot(x, f);
grid on
title('Plot of bell shaped curve')
```


MATLAB – function M-files

- The function `fname` must be in the file `fname.m`
- First non-comment line must be
  `function [OutputArgList] = fname(InputArgList)`
  - `InputArgList` is comma separated list of variables with values from calling script/function
  - `OutputArgList` is comma separated list of variables given values in function
- Variables, except `OutputArgList`, used inside function are local to function
- Use comments to document function M-files
  - Calling sequence: input and output arguments
  - Purpose of function
- Useful MATLAB built-in functions
  - `nargin` gives number of input arguments
  - `nargout` gives number of output arguments requested

MATLAB – function M-file npdf.m

```
function [f, df] = npdf(x, mu, sigma)
% [f, df] = npdf(x, mu, sigma)
% MATH2089: File = npdf.m
% Calculate values of the normal probability density function
% with mean mu and standard deviation sigma at the values in x
% -- Input arguments --
% x     = value(s) where function is to be evaluated (can be vector/array)
% mu    = mean of distribution (scalar, default value mu = 0)
% sigma = standard deviation of distribution (scalar, default sigma = 1)
%         If you want mu = 0 but sigma ~= 1 then both must be specified
% -- Output arguments --
% f     = values of the normal PDF at the values in x
%         If the input x is a vector or an array the output f will be
%         a vector/array of the same size
% df    = values of derivative of normal PDF at the values in x
%         Derivative(s) only calculated if function is called with two
%         output arguments
%         If the input x is a vector or an array the output df will be
%         a vector/array of the same size
```


MATLAB – function M-file npdf.m cont

```
% Set default values for input arguments
if nargin < 3
    sigma = 1;
end;
if nargin < 2
    mu = 0;
end;

% Use local variables to evaluate function
const = 1/(sigma*sqrt(2*pi));
z = x - mu;
var = sigma^2;
f = const*exp(-z.^2/(2*var));

% If function is called with two output arguments also calculate derivative
if nargout > 1
    df = -(z/var).*f;
else
    df = [];
end;
```