

# Numerical Analysis and Computing

## Random Notes #5 $\frac{1}{2}$

Peter Blomgren,  
(blomgren.peter@gmail.com)

Department of Mathematics and Statistics  
Dynamical Systems Group  
Computational Sciences Research Center  
San Diego State University  
San Diego, CA 92182-7720

<http://terminus.sdsu.edu/>

Fall 2014

# Outline

- 1 Comments, Clues, and Rants...
  - Matlab Functions... as arguments to Matlab Functions
  - Fancy “publishing” in Matlab...

## Function Calls

Cute idea: sending the mathematical function as a string argument to the function —

file: bisection.m

inline is depracted

```
function [ReturnValue] = bisection( fstring, a, b, TOL );  
f = inline( fstring, 'x' );  
...
```

Matlab prompt

```
>> my_root1 = bisection( 'sin(x)', -0.2, 0.5, 1e-8 );
```

## Function Calls — Other Variants, 1

```
file: bisection.m
```

```
function [ReturnValue] = bisection( f, a, b, TOL );  
  
fa = f(a);  
fb = f(b);  
...
```

```
Matlab prompt
```

```
"Anonymous Functions" replace inline functions
```

```
>> F = @(x) sin(x);  
>> G = @(x) (sin(x) .* cos(x) .* exp(x));  
>> my_root2 = bisection( F, -0.2, 0.5, 1e-8 );  
>> my_root3 = bisection( G, -0.3, 0.4, 1e-10 );
```

## Function Calls — Other Variants, 2

file: bisection.m

inline is deprecated

```
function [ReturnValue] = bisection( f_symb, a, b, TOL );  
f = matlabFunction( f_symb );  
...
```

## Matlab prompt

```
>> syms x  
>> F = sin(x);  
>> G = sin(x) .* cos(x) .* exp(x);  
>> my_root2 = bisection( F, -0.2, 0.5, 1e-8 );  
>> my_root3 = bisection( G, -0.3, 0.4, 1e-10 );
```

## Function Calls — Other Variants, 3

```
file: myGnarlyFunction.m
```

```
function [ReturnValue] = myGnarlyFunction( x );  
  
ReturnValue =  
(exp(sin(x.*cos(tan(x+pi/52))))+9).*sin(cos(x));
```

```
Matlab prompt
```

```
>> r4 = bisection( @myGnarlyFunction, -0.2, 0.5, 1e-8 );
```

## Function Calls — help...

For more information

- `>> help function`
- `>> help matlabFunction`
  - it has some cool features!
- `>> help punct`
  - See the entry for @ “At.”
- `>> help publish`
  - If you want to do fancy “publishing” of your code

## Publishing Example, 1/4

```
file: test_pub.m
```

```
%  
% This is the main driver...  
% (cannot be a function for publishing)  
%  
xv = 1.4 : 0.001 : 1.6;  
fxv = myGnarlyFunction(xv);  
r = bisection( @myGnarlyFunction, 1.4, 1.6, 1e-5 );  
%  
plot(xv,fxv);  
hold on  
axis([min(xv) max(xv) min(fxv)*1.05 max(fxv)*1.05])  
plot(r,myGnarlyFunction(r),'ro');  
grid on  
grid minor  
hold off
```



## Publishing Example, 2/4

## Matlab prompt

```
>> publish('test_pub', format_selection );  
where format_selection ∈ { 'html', 'doc', 'ppt', 'xml', 'latex' }
```

## Published result, part 1

```
%  
% This is the main driver...  
% (cannot be a function for publishing)  
%  
xv = 1.4 : 0.001 : 1.6;  
fxv = myGnarlyFunction(xv);  
r = bisection( @myGnarlyFunction, 1.4, 1.6, 1e-5 );  
%  
plot(xv,fxv);  
hold on  
axis([min(xv) max(xv) min(fxv)*1.05 max(fxv)*1.05])  
plot(r,myGnarlyFunction(r),'ro');  
grid on  
grid minor  
hold off
```

## Publishing Example, 3/4

## Published result, part 2

N	a	b	m	f(m)
1	1.400000	1.600000	1.500000	0.67242833
2	1.500000	1.600000	1.550000	0.24364039
3	1.550000	1.600000	1.575000	-0.03938094
4	1.550000	1.575000	1.562500	0.09713017
5	1.562500	1.575000	1.568750	0.02000099
6	1.568750	1.575000	1.571875	-0.01011431
7	1.568750	1.571875	1.570312	0.00457502
8	1.570312	1.571875	1.571094	-0.00279585
9	1.570312	1.571094	1.570703	0.00087822
10	1.570703	1.571094	1.570898	-0.00096090
11	1.570703	1.570898	1.570801	-0.00004194
12	1.570703	1.570801	1.570752	0.00041797
13	1.570752	1.570801	1.570776	0.00018798
14	1.570776	1.570801	1.570789	0.00007301
15	1.570789	1.570801	1.570795	0.00001553
16	1.570795	1.570801	1.570798	-0.00001321

## Publishing Example, 4/4

## Published result, part 3

