CMPSC/Math 451, Numerical Computation

Wen Shen

Department of Mathematics, Penn State University

Matlab: Effective programming

Three different ways of computing a vector product of two vectors.

$$z_i = x_i \cdot y_i, \qquad i = 1, 2, \ldots, n$$

Method 1:
Do not allocate memory space for z in advance.
Compute the elements in the new vector by a for-loop.

```
x = rand(n,1);     % random vector
y = rand(n,1);     % of length n
clear z;
t = cputime;
for i=1:n
   z(i) = x(i)*y(i);
end
cputime-t
```

Method 2:
Allocate space for z in advance, but still use a for loop.

```
z = zeros(n,1);
for i=1:n
   z(i) = x(i)*y(i);
end
```

Method 3:
Use vector operation in Matlab directly.

```
z = x.*y;
```

Result (The CPU-time measured in seconds):

n	Method 1	Method 2	Method 3
5000	2.86	0.24	0.00
10000	14.22	0.49	0.00
20000	59.65	0.97	0.01
100000	−−	4.87	0.03
1000000	−−	48.84	0.30

Moral: use pre-defined Matlab functions!

Romberg integration

$$I(f) = \int_0^{\pi/2} \cos(2x)e^{-x}dx, \qquad (= 0.2415759)$$

Result:

```
0.6221
0.3111    0.2074
0.2575    0.2397    0.2419
0.2455    0.2415    0.2416    0.2416
```

Error:

```
3.80e-01
6.94e-02   3.41e-02
1.59e-02   1.87e-03   2.81e-04
3.90e-03   1.11e-04   5.85e-06   1.47e-06
```

Expand the integration interval to 2π. Exact value$= 0.1996265$.
Result:

```
3.1475
1.7095    1.2302
0.5141    0.1156    0.0413
0.2570    0.1714    0.1751    0.1772
```

Error:

```
2.94e+00
1.50e+00   1.03e+00
3.14e-01   8.39e-02   1.58e-01
5.74e-02   2.82e-02   2.45e-02   2.24e-02
```

Matlab's adaptive Simpsons methode: quad

Syntax:

```
q = quad('f',a,b,toleranse,trace)
```

If the option *trace*$\neq 0$, Matlab will show the development of the integration.

```
quad('f3',0,2*pi,1.e-6,1)
```

Matlab's recursive numerical integration: quadl and quad8

Syntax and usage would be the same as for quad.
The functions use a high order recursive algorithm.