APPENDIX D

Solutions to Selected Exercises

CHAPTER 1

1.1
```
a = 3;
b = 5;
sum        =  a + b;
difference =  a - b;
product    =  a * b;
quotient   =  a / b;
```

CHAPTER 2

2.1 (a) comma should be replaced by decimal point
(e) asterisk should be omitted
(f) exponent must be integer
(h) comma should be replaced by decimal point

2.2 (b) decimal point not allowed
(c) first character must be a letter
(d) quotes not allowed
(h) blanks not allowed
(i) allowed but not recommended!
(k) asterisk not allowed
(l) allowed but not recommended!

2.3
(a) `p + w/u`
(b) `p + w/(u + v)`
(c) `(p + w/(u+v))/(p + w/(u-v))`
(d) `sqrt(x)`
(e) `y^(y+z)`
(f) `x^(y^z)`
(g) `(x^y)^z`
(h) `x - x^3/(3*2) + x^5/(5*4*3*2)`

2.4
```
(a)  i = i + 1
(b)  i = i^3 + j
(c)  if e > f
       g = e
     else
```

```
      g = f
    end
(d) if d > 0
      x = -b
    end
(e) x = (a + b)/(c * d)
```

2.5 (a) expression not allowed on left-hand side
(b) left-hand side must be valid variable name
(c) ditto

2.6
```
a = 2;
b = -10;
c = 12;
x = (-b + sqrt(b ^ 2 - 4 * a * c)) / (2 * a)
```

2.7
```
gallons = input('Enter gallons: ');
pints = input('Enter pints: ');
pints = pints + 8 * gallons;
litres = pints / 1.76
```

2.8
```
distance = 528;
litres = 46.23;
kml = distance / litres;
l100km = 100 / kml;
disp( 'Distance     Litres used     km/L     L/100km' );
disp( [distance litres kml l100km] );
```

2.9
```
t = a;
a = b;
b = t;
```

2.10
```
a = [a b];      % make 'a' into a vector
b = a(1);
a(1) = [];
```

2.11
```
(a) c = input('Enter Celsius temperature: ');
    f = 9 * c / 5 + 32;
    disp( ['The Fahrenheit temperature is: ' num2str(f)] );

(b) c = 20 : 30;
    f = 9 * c / 5 + 32;
    format bank;
    disp('        Celsius     Fahrenheit');
    disp([c'     f']);
```

2.12
```
degrees = 0 : 10 : 360;
radians = degrees / 180 * pi;
format bank;
disp('        Degrees     Radians');
disp([degrees'     radians']);
```

2.13
```
degrees = 0 : 30 : 360;
radians = degrees / 180 * pi;
sines = sin(radians);
cosines = cos(radians);
tans = tan(radians);
table = [degrees' sines' cosines' tans']
```

2.14
```
for int = 10 : 20
  disp( [int sqrt(int)] );
end
```

2.15
```
sum(2: 2: 200)
```

2.16
```
m = [5 8 0 10 3 8 5 7 9 4];
disp( mean(m) )
```

2.17 x = 2.0833, a = 4.

2.18
```
% With for loop
i = 1;
x = 0;
for a = i : i : 4
  x = x + i / a;
end

% With vectors
i = 1;
a = i : i : 4;
x = i ./ a;
sum(x)
```

2.19
```
(b) n = input('Number of terms? ');
    k = 1 : n;
    s = 1 ./ (k .^ 2);
    disp(sqrt(6 * sum(s)))
```

2.21
```
r = 5;
c = 10;
l = 4;
e = 2;
w = 2;
i = e / sqrt(r ^ 2 + (2 * pi * W * l-1 / (2 * pi * w *
c))^ 2)
```

2.22
```
con = [200 500 700 1000 1500];
for units = con
  if units <= 500
    cost = 0.02 * units;
  elseif units <= 1000
    cost = 10 + 0.05 * (units - 500);
  else
    cost = 35 + 0.1 * (units - 1000);
  end
  charge = 5 + cost;
  disp( charge )
end
```

2.24
```
money = 1000;
for month = 1 : 12
  money = money * 1.01;
end
```

2.26
```
t = 1790 : 10 : 2000;
p = 197273000 ./ (1 + exp(-0.03134 * (t - 1913.25)));
disp([t'   p']);
pause;
plot(t,p);
```

2.27
```
(a) r = 0.15;
    l = 50000;
    n = 20;
    p = r * l * (1 + r / 12) ^ (12 * n) / ...
        (12 * ((1 + r / 12) ^ (12 * n) - 1))
```

2.28
```
r = 0.15;
l = 50000;
p = 800;
n = log(p / (p - r * l / 12)) / (12 * log(1 + r / 12))
```

CHAPTER 3

3.1 You should get a picture of tangents to a curve.

3.2 **(a)** 4

(b) 2

(c) The algorithm (attributed to Euclid) finds the HCF (Highest Common Factor) of two numbers by using the fact that the HCF divides exactly into the difference between the two numbers, and that if the numbers are equal, they are equal to their HCF.

3.3
```
f = input('Enter Fahrenheit temperature: ');
c = 5 / 9 * (f - 32);
disp( ['The Celsius temperature is: ' num2str(c)] );
```

3.4
```
a = input('Enter first number: ');
b = input('Enter second number: ');
if a < b
  disp( [ num2str(b) ' is larger.'] );
elseif a > b
  disp( [ num2str(a) ' is larger.'] );
else
  disp( 'Numbers are equal.' );
end
```

3.6
1. Input a, b, c, d, e, f
2. $u = ae - db$, $v = ec - bf$
3. If $u = 0$ and $v = 0$ then
   Lines coincide
   Otherwise if $u = 0$ and $v \neq 0$ then
   Lines are parallel
   Otherwise
   $x = v/u$, $y = (af - dc)/u$
   Print x, y

4. Stop.

```
a = input('Enter a: ');
b = input('Enter b: ');
c = input('Enter c: ');
d = input('Enter d: ');
e = input('Enter e: ');
f = input('Enter f: ');
u = a * e - b * d;
v = c * e - b * f;
if u == 0
  if v == 0
    disp('Lines coincide.');
  else
    disp('Lines are parallel.');
  end
else
  x = v / u;
  y = (a * f - d * c) / u;
  disp( [x y] );
end
```

CHAPTER 4

4.2

```
(a) log(x + x ^ 2 + a ^ 2)
(b) (exp(3 * t) + t ^ 2 * sin(4 * t)) * (cos(3 * t)) ^ 2
(c) 4 * atan(1)
(d) sec(x)^2 + cot(x)
(e) atan(a / x)
```

4.3

```
m = input('Enter length in metres: ');
inches = m * 39.37;
feet = fix(inches / 12);
inches = rem(inches, 12);
yards = fix(feet / 3);
feet = rem(feet, 3);
disp( [yards feet inches] );
```

4.5

```
a = 10;
x = 1;
k = input('How many terms do you want? ');
for n = 1 : k
  x = a * x / n;
  if rem(n, 10) == 0
    disp( [n x] );
  end
end
```

4.6

```
secs = input('Enter seconds: ');
mins = fix(secs / 60);
secs = rem(secs, 60);
hours = fix(mins / 60);
mins = rem(mins, 60);
disp( [hours mins secs] );
```

CHAPTER 5

5.2

```
(a) 1 1 0
(b) 0 1 0
(c) 1 0 1
(d) 0 1 1
(e) 1 1 1
(f) 0 0 0
(g) 0 2
(h) 0 0 1
```

5.3

```
neg = sum(x < 0);
pos = sum(x > 0);
zero = sum(x == 0);
```

5.7

```
5.7    units = [200 500 700 1000 1500];
       cost = 10 * (units > 500) + 25 * (units > 1000) + 5;
       cost = cost + 0.02 * (units <= 500) .* units;
       cost = cost + 0.05 * (units > 500 & units <= 1000) .* (units - 500);
       cost = cost + 0.1 * (units > 1000) .* (units - 1000);
```

CHAPTER 6

6.5

```
function x = mygauss(a, b)
n = length(a);

a(:,n+1) = b;

for k = 1:n
  a(k,:) = a(k,:)/a(k,k);        % pivot element must be 1

  for i = 1:n
    if i ~= k
      a(i,:) = a(i,:) - a(i,k) * a(k,:);
    end
  end
end

% solution is in column n+1 of a:
x = a(:,n+1);
```

CHAPTER 7

7.1

```
function pretty(n, ch)
line = char(double(ch)*ones(1,n));
disp(line)
```

7.2

```
function newquot(fn)
x = 1;
h = 1;
for i = 1 : 10
  df = (feval(fn, x + h) - feval(fn, x)) / h;
  disp( [h, df] );
  h = h / 10;
end
```

7.3
```
function y = double(x)
y = x * 2;
```

7.4
```
function [xout, yout] = swop(x,y)
xout = y;
yout = x;
```

7.6
```
% Script file
for i = 0 : 0.1 : 4
  disp( [i, phi(i)] );
end

% Function file phi.m
function y = phi(x)
a = 0.4361836;
b = -0.1201676;
c = 0.937298;
r = exp(-0.5 * x * x) / sqrt(2 * pi);
t = 1 / (1 + 0.3326 * x);
y = 0.5 - r * (a * t + b * t * t + c * t ^ 3);
```

7.8
```
function y = f(n)
if n > 1
  y = f(n - 1) + f(n - 2);
else
  y = 1;
end
```

CHAPTER 8

8.1
```
balance = 1000;
for years = 1 : 10
  for months = 1 : 12
     balance = balance * 1.01;
   end
   disp( [years balance] );
 end
```

8.2
```
(a) terms = 100;
    pi = 0;
    sign = 1;
    for n = 1 : terms
      pi = pi + sign * 4 / (2 * n - 1);
      sign = sign * (-1);
    end

(b) terms = 100;
    pi = 0;
    for n = 1 : terms
      pi = pi + 8 / ((4 * n - 3) * (4 * n - 1));
    end
```

8.3

```
a = 1;
n = 6;
for i = 1 : 10
  n = 2 * n;
  a = sqrt(2 - sqrt(4 - a * a));
  l = n * a / 2;
  u = l / sqrt(1 - a * a / 2);
  p = (u + l) / 2;
  e = (u - l) / 2;
  disp( [n, p, e] );
end
```

8.5

```
x = 0.1;
for i = 1 : 7
  e = (1 + x) ^ (1 / x);
  disp( [x, e] );
  x = x / 10;
end
```

8.6

```
n =6;
T = 1;
i = 0;
for t = 0:0.1:1
  i = i + 1;
  F(i) = 0;
  for k = 0 : n
    F(i) = F(i) + 1 / (2 * k + 1) * sin((2 * k + 1) * pi * t / T);
  end
  F(i) = F(i) * 4 / pi;
end
t = 0:0.1:1;
disp( [t' F'] )
plot(t, F)
```

8.8

```
sum = 0;
terms = 0;
while (sum + terms) <= 100
  terms = terms + 1;
  sum = sum + terms;
end
disp( [terms, sum] );
```

8.10

```
m = 44;
n = 28;
while m ~= n
  while m > n
    m = m - n;
  end
  while n > m
    n = n - m;
  end
end
disp(m);
```

CHAPTER 9

9.1

```
t = 1790:2000;
P = 197273000 ./ (1+exp(-0.03134*(t-1913.25)));
plot(t, P), hold, xlabel('Year'), ylabel('Population size')
census = [3929 5308 7240 9638 12866 17069 23192 31443 38558 ...
          50156 62948 75995 91972 105711 122775 131669 150697];
census = 1000 * census;
plot(1790:10:1950, census, 'o'), hold off
```

9.2

```
a = 2;
q = 1.25;
th = 0:pi/40:5*pi;
subplot(2,2,1)
plot(a*th.*cos(th), a*th.*sin(th)), ...
     title('(a) Archimedes')    % or use polar
subplot(2,2,2)
plot(a/2*q.^th.*cos(th), a/2*q.^th.*sin(th)), ...
     title('(b) Logarithmic')   % or use polar
```

9.4

```
n=1:1000;
d = 137.51;
th = pi*d*n/180;
r = sqrt(n);
plot(r.*cos(th), r.*sin(th), 'o')
```

9.6

```
y(1) = 0.2;
r = 3.738;
for k = 1:600
  y(k+1) = r*y(k)*(1 - y(k));
end
plot(y, '.w')
```

CHAPTER 11

11.1

```
x = 2;
h = 10;
for i = 1 : 20
  h = h / 10;
  dx = ((x + h) ^ 2 - x * x) / h;
  disp( [h, dx] );
end
```

CHAPTER 13

13.1
```
heads = rand(1, 50) < 0.5;
tails = ~heads;
heads = heads * double('H');
tails = tails * double('T');
coins = char(heads + tails)
```

13.2
```
bingo = 1 : 99;
for i = 1 : 99
  temp = bingo(i);
  swop = floor(rand * 99 + 1);
  bingo(i) = bingo(swop);
  bingo(swop) = temp;
end
for i = 1 : 10 : 81
  disp(bingo(i : i + 9))
end
disp(bingo(91 : 99))
```

13.4
```
circle = 0;
square = 1000;
for i = 1 : square
  x = 2 * rand - 1;
  y = 2 * rand - 1;
  if (x * x + y * y) < 1
    circle = circle + 1;
  end
end
disp( circle / square * 4 );
```

CHAPTER 14

14.1 **(a)** Real roots at 1.856 and -1.697, complex roots at $-0.0791 \pm 1.780i$.
(b) 0.589, 3.096, 6.285, …(roots get closer to multiples of π).
(c) 1, 2, 5.
(d) 1.303.
(e) -3.997, 4.988, 2.241, 1.768.

14.2 Successive bisections are: 1.5, 1.25, 1.375, 1.4375, and 1.40625. The exact answer is 1.414214…, so the last bisection is within the required error.

14.3 22 (exact answer is 21.3333).

14.4 After 30 years the exact answer is 2117 ($1000e^{rt}$).

14.6 The differential equations to be solved are:

$$dS/dt = -r_1 S,$$
$$dY/dt = r_1 S - r_2 Y.$$

The exact solution after 8 h is $S = 6.450 \times 10^{25}$ and $Y = 2.312 \times 10^{26}$.

14.8
```
function s = simp(f, a, b, h)
x1 = a + 2 * h : 2 * h : b - 2 * h;
sum1 = sum(feval(f, x1));
x2 = a + h : 2 * h : b - h;
sum2 = sum(feval(f, x2));
s = h / 3 * (feval(f, a) + feval(f, b) + 2 * sum1 + 4 * sum2);
```

With 10 intervals ($n = 5$), the luminous efficiency is 14.512725%. With 20 intervals it is 14.512667%. These results justify the use of 10 intervals in any further computations involving this problem. This is a standard way of testing the accuracy of a numerical method: halve the step-length and see how much the solution changes.

14.9
```
% Command Window
beta = 1;
ep = 0.5;
[t, x] = ode45(@vdpol, [0 20], [0; 1], [], beta, ep);
plot(x(:,1), x(:,2))

% Function file vdpol.m
function f = vdpol(t, x, b, ep)
f = zeros(2,1);
f(1) = x(2);
f(2) = ep * (1 - x(1)^2) * x(2) - b^2 * x(1);
```