```
> restart:
> Digits:=15;
```

$$Digits := 15 \quad (1)$$

```
> Euler := proc( f,a,b,alpha, N )
  local h:
  local xx:
  local j:
  h:=evalf((b-a)/N):
  xx:=alpha:
  for j from 0 to N-1 do
  xx:= xx+f(a+j*h,xx)*h:
  od:
  xx;
  end;
```

Euler := **proc**(*f*, *a*, *b*, alpha, *N*) **(2)**

    local *h*, *xx*, *j*;

    $h := evalf((b-a)/N)$; $xx :=$ alpha; **for** j **from** 0 **to** $N-1$ **do** $xx := xx + f(a+j*h, xx)*h$ **end do**; *xx*

end proc

```
> ModEuler:=  proc( f,a,b,alpha, N )
  local h:
  local xx:
  local j:
  h:=evalf((b-a)/N):
  xx:=alpha:
  for j from 0 to N-1 do
  xx:= xx+(f(a+j*h,xx)+f(a+(j+1)*h,xx+f(a+j*h,xx)*h))*h/2:
  od:
  xx;
  end;
```

ModEuler := **proc**(*f*, *a*, *b*, alpha, *N*) **(3)**

    local *h*, *xx*, *j*;

    $h := evalf((b-a)/N)$;

    $xx :=$ alpha;

    for j **from** 0 **to** $N-1$ **do**

        $xx := xx + 1/2*(f(a+j*h, xx) + f(a+(j+1)*h, xx + f(a+j*h, xx)*h))*h$

    end do;

    xx

end proc

```
> RungeKutta:=  proc( f,a,b,alpha, N )
  local h:
  local xx:
  local j:
  local k1,k2,k3,k4:
  h:=evalf((b-a)/N):
  xx:=alpha:
  for j from 0 to N-1 do
  k1:= f(a+j*h,xx)*h:
  k2:= f(a+(j+1/2)*h,xx+k1/2)*h:
  k3:= f(a+(j+1/2)*h,xx+k2/2)*h:
  k4:= f(a+(j+1)*h,xx+k3)*h:
  xx:= xx+(k1+2*k2+2*k3+k4)/6:
  od:
  xx;
  end;
```

RungeKutta := **proc**(*f*, *a*, *b*, alpha, *N*) **(4)**

```
    local h, xx, j, k1, k2, k3, k4;
    h := evalf((b - a)/N);
    xx := alpha;
    for j from 0 to N - 1 do
        k1 := f(a + j*h, xx)*h;
        k2 := f(a + (j + 1/2)*h, xx + 1/2*k1)*h;
        k3 := f(a + (j + 1/2)*h, xx + 1/2*k2)*h;
        k4 := f(a + (j + 1)*h, xx + k3)*h;
        xx := xx + 1/6*k1 + 1/3*k2 + 1/3*k3 + 1/6*k4
    end do;
    xx
end proc
```

```
> p:= x-> x^3-x+7;
```

$$p := x \rightarrow x^3 - x + 7 \quad (5)$$

```
> d:=degree(p(x));
```

$$d := 3 \quad (6)$$

```
> N:=10;
```

$$N := 10 \quad (7)$$

```
> print("Euler Method");
  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  print(Euler(f,1.0,0.0,alpha,N)):
  od:

  print("Modified Euler Method");
  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  print(ModEuler(f,1.0,0.0,alpha,N)):
  od:

  print("Runge-Kutta Method");
  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  print(RungeKutta(f,1.0,0.0,alpha,N)):
  od:
```

"Euler Method"

$$1.04728328427204 + 1.57700567531532\, I$$

$$-2.17222271705722 - 0.0316265546588369\, I$$

$$1.12589104959449 - 1.54491705152830\, I$$

"Modified Euler Method"

$$1.04139992899138 + 1.50267577007144\, I$$

$$-2.08284693732553 + 0.0000149261144980\, I$$

$$1.04157427521343 - 1.50261841360371\, I$$

"Runge-Kutta Method"

$$1.04334253490268 + 1.50533876481422\,I$$
$$-2.08680850930348 - 0.0000643542682647\,I$$
$$1.04346473292639 - 1.50527188534978\,I \qquad (8)$$

```
> fsolve(p(x),x,complex);
```

$$-2.08674533988267,\ 1.04337266994133 - 1.50528388855442\,I,\ 1.04337266994133 + 1.50528388855442\,I \qquad (9)$$

Problem 6.1

```
> #restart;
> p := x -> x^3 -7*x^2 +3;
  d:=degree(p(x));
```

$$p := x \rightarrow x^3 - 7\,x^2 + 3$$
$$d := 3 \qquad (10)$$

```
> N:=10;
  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  print("Euler",Euler(f,1.0,0.0,alpha,N)):
  print("ModEuler",ModEuler(f,1.0,0.0,alpha,N)):
  print("RungeKutta",RungeKutta(f,1.0,0.0,alpha,N)):
  od:
  fsolve(p(x),x,complex);
```

$$N := 10$$
"Euler", $0.650882028361111 - 0.0087230632202095\,I$
"ModEuler", $0.689197715277092 - 0.00036234162324375\,I$
"RungeKutta", $0.689500917853165 - 0.00011938567946060\,I$
"Euler", $-0.650023240824978 - 0.0490164501406343\,I$
"ModEuler", $-0.630241203269439 + 0.0039025921395820\,I$
"RungeKutta", $-0.627164204188939 + 0.00006351310166193\,I$
"Euler", $6.55351357902532 + 0.044520829860033\,I$
"ModEuler", $6.91662128383679 + 0.005188428366550\,I$
"RungeKutta", $6.93775462645458 + 0.0001340602902109\,I$
$$-0.627161175445869,\ 0.689490715304173,\ 6.93767046014170 \qquad (11)$$

Problem 6.2

```
> restart:
> Digits:=20;
```

$$Digits := 20 \qquad (12)$$

```
> p := x -> x^3 -7*x^2 +3;
```

$$p := x \rightarrow x^3 - 7\,x^2 + 3 \qquad (13)$$

Euler as predictor + Newton as corrector
Runge-Kutta as predictor and Newton as corrector

```
> N:=10;
  a:=1.0:
  b:=0.0:
  N:=10;
  d:=degree(p(x));

  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  dHx:= unapply(diff(H(t,x),x),t,x):
```

```
h:=evalf((b-a)/N):
xx:=alpha:
for k from 0 to N-1 do
xx:= xx+f(a+k*h,xx)*h:
xx:= xx- H(a+(k+1)*h,xx)/dHx(a+(k+1)*h,xx):
od:
print("Euler",xx);
xx:=alpha:
for k from 0 to N-1 do
k1:= f(a+k*h,xx)*h:
k2:= f(a+(k+1/2)*h,xx+k1/2)*h:
k3:= f(a+(k+1/2)*h,xx+k2/2)*h:
k4:= f(a+(k+1)*h,xx+k3)*h:
xx:= xx+(k1+2*k2+2*k3+k4)/6:
xx:= xx- H(a+(k+1)*h,xx)/dHx(a+(k+1)*h,xx):
od:
print("Runge-Kutta",xx);
od:
```

$$N := 10$$
$$N := 10$$
$$d := 3$$
$$\text{"Euler"}, 0.68949024457654550798 + 3.229734429904315\ 10^{-8}\ I$$
$$\text{"Runge-Kutta"}, 0.68949071530417318996 - 9.022460214\ 10^{-20}\ I$$
$$\text{"Euler"}, -0.62715991900238091039 - 6.725098555663752\ 10^{-7}\ I$$
$$\text{"Runge-Kutta"}, -0.62716117544586942164 - 2.53408795388\ 10^{-17}\ I$$
$$\text{"Euler"}, 6.9390020198013923012 + 0.0011771887472812400 74\ I$$
$$\text{"Runge-Kutta"}, 6.9376704600119335705 - 7.9839384794184\ 10^{-11}\ I \quad \textbf{(14)}$$

```
> fsolve(p(x),x,complex);
```

$$-0.62716117544586941312, 0.68949071530417319030, 6.9376704601416962228 \quad \textbf{(15)}$$

Problem 6.3

```
> Digits:=20;
```

$$Digits := 20 \quad \textbf{(16)}$$

```
> p := x -> x^3 -6*x^2 +12*x-8;
```

$$p := x \rightarrow x^3 - 6\,x^2 + 12\,x - 8 \quad \textbf{(17)}$$

```
> d:=degree(p(x));
```

$$d := 3 \quad \textbf{(18)}$$

Euler without Newton

```
> N:=10;
  a:=1.0:
  b:=0.0:
  N:=10;
  d:=degree(p(x));

  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  dHx:= unapply(diff(H(t,x),x),t,x):
  h:=evalf((b-a)/N):
  xx:=alpha:
  for k from 0 to N-1 do
  xx:= xx+f(a+k*h,xx)*h:
  od:
  print("Euler",xx);
```

```
print("error", abs(xx-2));
od:
```

$N := 10$

$N := 10$

$d := 3$

"Euler", 1.4880178073577713756 − 0.2010430137817078I321 I

"error", 0.55004005215363731719

"Euler", 2.0589226447841402744 + 1.0528641069551032166 I

"error", 1.0545115958502898315

"Euler", 2.0785145565937421440 − 1.1922462426296878593 I

"error", 1.1948287068286066545 **(19)**

```
> fsolve(p(x),x,complex);
```

2., 2., 2. **(20)**

Runge-Kutta and Newton

```
> N:=10;
  a:=1.0:
  b:=0.0:
  N:=10;
  d:=degree(p(x));

  for j from 0 to d-1 do
  alpha:= evalf(exp(2*j*Pi*I/d)):
  RandGamma:= 0.196743+0.7981302*I:#very provisional random gamma
  H:= unapply((1-t)*p(x)+t*RandGamma*(x^d-1),t,x):
  f:= unapply(-diff(H(t,x),t)/diff(H(t,x),x),t,x):
  dHx:= unapply(diff(H(t,x),x),t,x):
  h:=evalf((b-a)/N):
  xx:=alpha:
  for k from 0 to N-1 do
  k1:= f(a+k*h,xx)*h:
  k2:= f(a+(k+1/2)*h,xx+k1/2)*h:
  k3:= f(a+(k+1/2)*h,xx+k2/2)*h:
  k4:= f(a+(k+1)*h,xx+k3)*h:
  xx:= xx+(k1+2*k2+2*k3+k4)/6:
  xx:= xx- H(a+(k+1)*h,xx)/dHx(a+(k+1)*h,xx):
  od:
  print("Runge-Kutta",xx);
  print("error", abs(xx-2));
  od:
```

$N := 10$

$N := 10$

$d := 3$

"Runge-Kutta", 1.8755797325817693387 − 0.0486853100655087388 76 I

"error", 0.13360637095811994546

"Runge-Kutta", 1.9859949247121240875 + 0.15481914667292782378 I

"error", 0.15545131170354469229

"Runge-Kutta", 2.1272306591820383169 − 0.12983380022179632134 I

"error", 0.18178134204568220642 **(21)**

Better than Euler, but still pretty bad. The problem is the root is a multiple root. Techinques to deal with singular endpoints are called endgames: we will talk about these later in the semester.