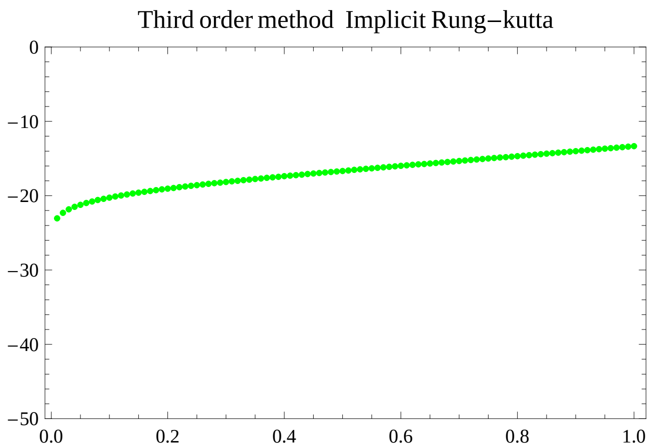
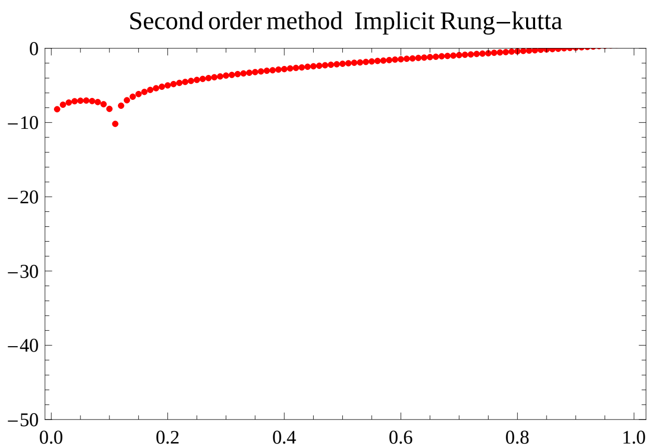
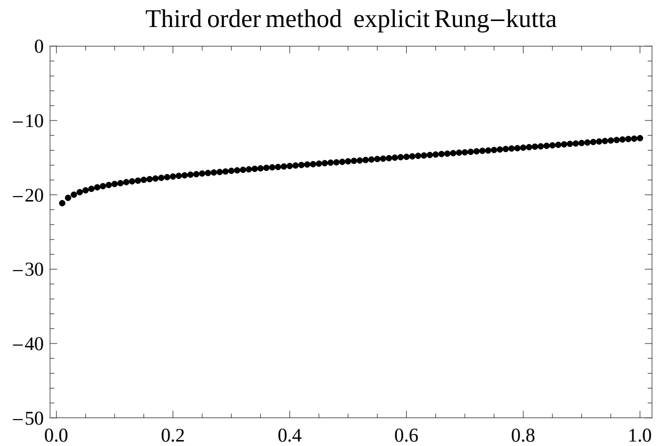
```
(*Third order method  Implicit Rung-kutta *)
(*{y′=e^t y(t)+-e^2t   t∈⌊0,1⌋
      y(0)=1
*)
(*  k1 =h f(tn,yn  )
    k2 =h f(tn+2/3 h+yn +1/3 k1 +1/3 k2 )
    yn+1 =yn +1/4 k1+ 3/4 k2 *)
f[t_, y_] := Exp[t] * y - Exp[2 * t];
eq = yy'[t] == Exp[t] * yy[t] - Exp[2 t];
sol = DSolve[{eq, yy[0] == 1}, yy[t], t];
exact = sol[[1]][[1]][[2]];
h = Input[" Enter lentgh of interval="];
y[0] = 1;
n = 1 / h;
Do[
  k1 = h * f[i * h, y[i]];
  knew2 = N[NSolve [kk2 ==  h * f[i * h + (2 / 3) * h, y[i] + (1 / 3) * k1 + (1 / 3) * kk2] ]] ;
  y[i + 1] = y[i] + (1 / 4) * k1 + (3 / 4) * knew2[[1]][[1]][[2]];
  , {i, 0, n - 1}];
Table[y[i], {i, 0, n}];
ar = Table[(exact /. t → (i * h)) - y[i], {i, 0, n}];
ar1 = Table[{h * i, Log[Abs[ar[[i + 1]]]]}, {i, 1, n}];
im3 = ListPlot[ar1, PlotRange → {-50, 0}, PlotStyle → Green, Frame → True,
  PlotLabel → "Third order method  Implicit Rung-kutta"(*,PlotMarkers→"°"*)]
(* -------------------------------------------------------*)
(*Second order method  Implicit Rung-kutta *)
(*k1 =h f(tn+1/2 h+yn +1/2 k1  );
  k2 =h f(tn,yn  ) ;
 yn+1 =yn +k1 ; *)                        f[t_, y_] := Exp[t] * y - Exp[2 * t];
eq = yy'[t] == Exp[t] * yy[t] - Exp[2 t];
sol = DSolve[{eq, yy[0] == 1}, yy[t], t];
exact = sol[[1]][[1]][[2]];
h = Input[" Enter lentgh of interval="];
y[0] = 1;
n = 1 / h;
Do[
  knew1 = N[NSolve [kk1 ==  h * f[i * h + (1 / 2) * h, y[i] + (1 / 2) * k1] ]] ;
  k2 = h * f[i * h, y[i]];
  y[i + 1] = y[i] + (1 / 2) * knew1[[1]][[1]][[2]];
  , {i, 0, n - 1}];
Table[y[i], {i, 0, n}];
ar = Table[(exact /. t → (i * h)) - y[i], {i, 0, n}];
ar1 = Table[{h * i, Log[Abs[ar[[i + 1]]]]}, {i, 1, n}];
im2 = ListPlot[ar1, PlotRange → {-50, 0}, PlotStyle → Red, Frame → True,
  PlotLabel → "Second order method  Implicit Rung-kutta"(*,PlotMarkers→"Λ"*)]
(* -------------------------------------------------------*)
(*Third order method  explicit Rung-kutta *)
(* NEARLY OPTIMAL METHOD*)
 (*  k1 =h f(tn,yn  )
    k2 =h f(tn+1/2 h+yn +1/2 k1  )
     k3 =h f(tn+3/4 h+yn +3/4 k2  )
     yn+1 =yn +2/9 k1+ 3/9 k2 + 4/9 k3
 *)
```

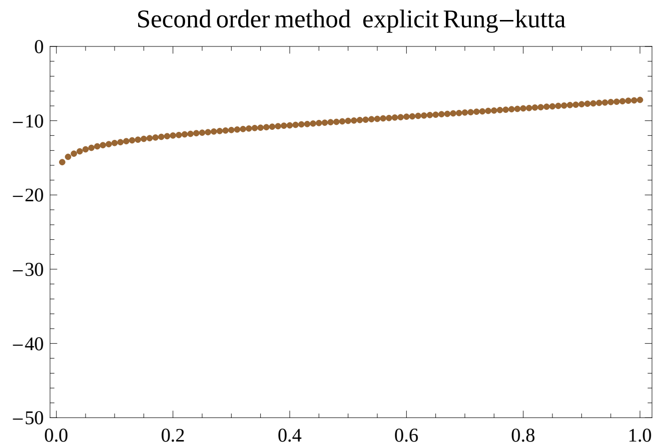
```
f[t_, y_] := Exp[t] * y - Exp[2 * t];
eq = yy'[t] == Exp[t] * yy[t] - Exp[2 t];
sol = DSolve[{eq, yy[0] == 1}, yy[t], t];
exact = sol[[1]][[1]][[2]];
h = Input[" Enter lentgh of interval="];
y[0] = 1;
n = 1 / h;
Do[
  k1 = h * f[i * h, y[i]] ;
   k2 =  h * f[i * h + 1 / 2 * h, y[i] + 1 / 2 * k1] ;
  k3 =  h * f[i * h + 3 / 4 * h, y[i] + 3 / 4 * k2] ;
  y[i + 1] = y[i] + 2 / 9 * k1 + 3 / 9 * k2 + 4 / 9 * k3
  , {i, 0, n - 1}];
Table[y[i], {i, 0, n}];
ar = Table[(exact /. t → (i * h)) - y[i], {i, 0, n}];
ar1 = Table[{h * i, Log[Abs[ar[[i + 1]]]]}, {i, 0, n}];
ex3 = ListPlot[ar1, PlotRange → {-50, 0}, PlotStyle → Black,
  Frame → True, PlotLabel -> "Third order method  explicit Rung-kutta"(*,
  PlotMarkers→"*"*)]
(*------------------------------------------------------*)
(*Second order method  explicit Rung-kutta *)
 (*  k1 =h f(tn,yn  )
    k2 =h f(tn+2/3 h+yn +2/3 k1  )
    yn+1 =yn +1/4 k1+ 3/4 k2
 *)
f[t_, y_] := Exp[t] * y - Exp[2 * t];
eq = yy'[t] == Exp[t] * yy[t] - Exp[2 t];
sol = DSolve[{eq, yy[0] == 1}, yy[t], t];
exact = sol[[1]][[1]][[2]];
h = Input[" Enter lentgh of interval="];
y[0] = 1;
n = 1 / h;
Do[
  k1 = h * f[i * h, y[i]] ;
   k2 =  h * f[i * h + 2 / 3 * h, y[i] + 2 / 3 * k1] ;
  y[i + 1] = y[i] + 1 / 4 * k1 + 3 / 4 * k2
  , {i, 0, n - 1}];
Table[y[i], {i, 0, n}];
ar = Table[(exact /. t → (i * h)) - y[i], {i, 0, n}];
ar1 = Table[{h * i, Log[Abs[ar[[i + 1]]]]}, {i, 0, n}];
ex2 = ListPlot[ar1, PlotRange → {-50, 0}, PlotStyle → Brown, Frame → True,
  PlotLabel -> "Second order method  explicit Rung-kutta"(*,PlotMarkers→"*"*)]
(* ------------------------------------------------------*)
(*Forth order method  explicit Rung-kutta *)
 (*  k1 =h f(tn,yn  )
    k2 =h f(tn+ h+yn + k1  )
     k3 =h f(tn+ h+yn + k2  )
      k4 =h f(tn+1/2 h+yn +1/2 k3  )
      yn+1 =yn +1/6 k1+ 2/6 k2 + 2/6 k3+1/6 k4
 *)
f[t_, y_] := Exp[t] * y - Exp[2 * t];
eq = yy'[t] == Exp[t] * yy[t] - Exp[2 t];
sol = DSolve[{eq, yy[0] == 1}, yy[t], t];
exact = sol[[1]][[1]][[2]];
```

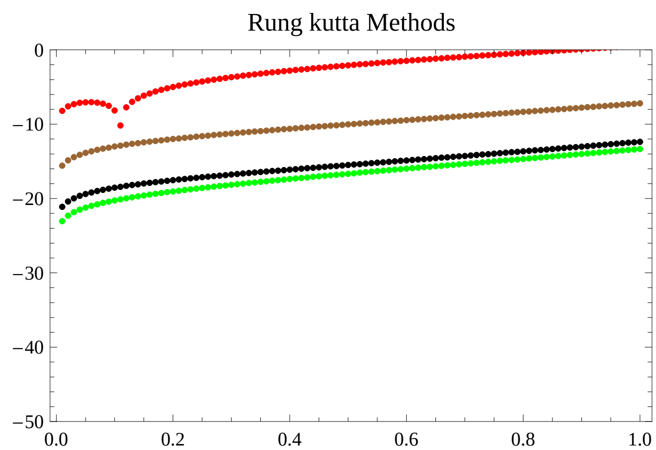
```
h = Input[" Enter lentgh of interval="];
y[0] = 1;
n = 1 / h;
Do[
  k1 = h * f[i * h, y[i]] ;
   k2 =  h * f[i * h + h, y[i] + k1] ;
  k3 =  h * f[i * h + h, y[i] + k2] ;
  k4 =  h * f[i * h + 1 / 2 * h, y[i] + 1 / 2 * k2] ;
  y[i + 1] = y[i] + 1 / 6 * k1 + 2 / 6 * k2 + 2 / 6 * k3 + 1 / 6 * k4
  , {i, 0, n - 1}];
Table[y[i], {i, 0, n}];
ar = Table[(exact /. t → (i * h)) - y[i], {i, 0, n}];
ar1 = Table[{h *i, Log[Abs[ar[[i + 1]]]]}, {i, 0, n}];
ex4 = ListPlot[ar1, PlotRange → {-50, 0}, PlotStyle → Yellow,
  Frame → True, PlotLabel -> "Forth order method  explicit Rung-kutta"(*,
  PlotMarkers→"*"*)]
Show[{im3, im2, ex3, ex2, ex4}, PlotLabel → "Rung kutta Methods"]
```






Third order method explicit Rung−kutta
0
−10
−20
−30
−40
−50
0.0
0.2
0.4
0.6
0.8
1.0


Second order method explicit Rung−kutta
0
−10
−20
−30
−40
−50
0.0
0.2
0.4
0.6
0.8
1.0


Rung kutta Methods
0
−10
−20
−30
−40
−50
0.0
0.2
0.4
0.6
0.8
1.0