Runge's methods
KZ

(x, y) - initial data

h - the grid step

f - initial function

b - the right boundary

3 steps of order 3

```
solveDE[f_, x_, y_, h_, b_] := Module[{x0, k1, k2, k3, y0, array = {}, arrayx = {}},
  x0 = x;
  y0 = y;
  While[x0 < b,
   k1 = f[x0, y0];
   k2 = f[x0 + h / 2, y0 + h k1 / 2];
   k3 = f[x0 + h, y0 - h k1 + 2 h k2];
   y0 = y0 + h (k1 + 4 k2 + k3) / 6;
   AppendTo[array, y0];
   AppendTo[arrayx, x0];
   x0 += h];
  {arrayx, array} // Transpose]
```

Let's check our function with the integrated functions

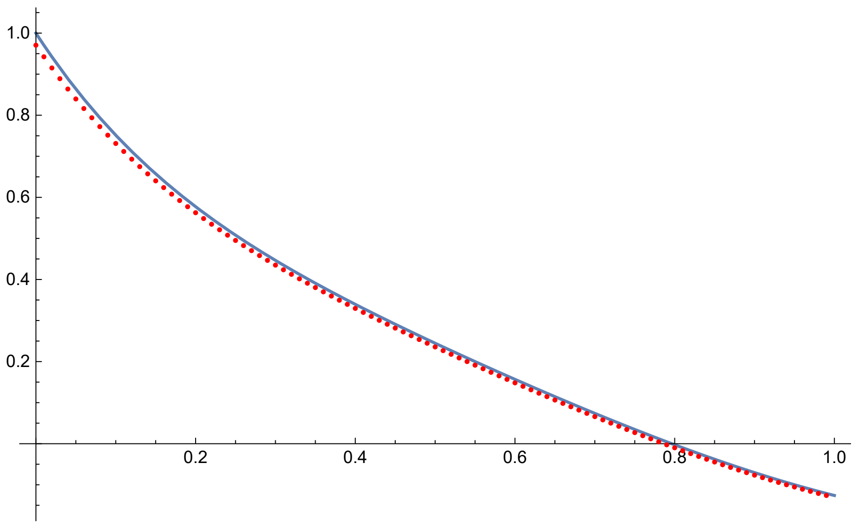
```
f[x_, y_] := Cos[3 x] - 4 y
```

```
p = solveDE[f, 0, 1, 0.01, 1];
```

```
s = DSolve[{y'[x] == -4 y[x] + Cos[3 x], y[0] == 1}, y[x], x][[1, 1]] // Values
```

$$\frac{1}{25} e^{-4 x} \left(21 + 4 e^{4 x} \mathrm{Cos}[3 x] + 3 e^{4 x} \mathrm{Sin}[3 x]\right)$$

```
Plot[s /. x → k, {k, 0, 1}, Epilog → {Red, PointSize[0.006], Point[p]}]
```

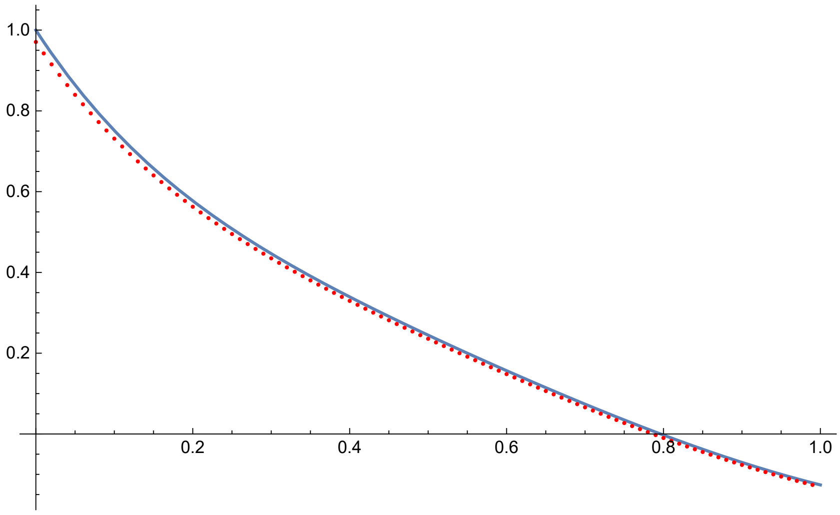


Classic Runge's method of order 4

```
solveDE4order[x_, y_, h_, b_] := Module[{x0, k1, k2, k3, k4, y0, array = {}, arrayx = {}},
  x0 = x;
  y0 = y;
  While[x0 < b,
   k1 = f[x0, y0];
   k2 = f[x0 + h / 2, y0 + h k1 / 2];
   k3 = f[x0 + h / 2, y0 + h k2 / 2];
   k4 = f[x0 + h, y0 + h k3];
   y0 = y0 + h (k1 + 2 k2 + 2 k3 + k4) / 6;
   AppendTo[array, y0];
   AppendTo[arrayx, x0];
   x0 += h];
  {arrayx, array} // Transpose]
```

```
p2 = solveDE4order[0, 1, 0.01, 1];
```

```
Show[{Plot[s /. x → k, {k, 0, 1}],
  Graphics[{Red, PointSize[0.005], Point[p2]}]}, PlotRange → Full]
```



2 steps of order 2

```
solveDE2order[x_, y_, h_, b_] := Module[{x0, k1, k2, y0, array = {}, arrayx = {}},
  x0 = x;
  y0 = y;
  While[x0 < b,
   k1 = f[x0, y0];
   k2 = f[x0 + h / 2, y0 + h k1 / 2];
   y0 = y0 + h k2;
   AppendTo[array, y0];
   AppendTo[arrayx, x0];
   x0 += h];
  {arrayx, array} // Transpose]
```

```
p3 = solveDE2order[0, 1, 0.01, 1];
```

```
Show[{Plot[s /. x → k, {k, 0, 1}],
  Graphics[{Red, PointSize[0.005], Point[p3]}]}, PlotRange → Full]
```