In[3148]:=

```
sol = NDSolve[{x'[t] == 10 (y[t] - x[t]),
   y'[t] == x[t] (28 - z[t]) - y[t], z'[t] == x[t] * y[t] - 8/3 * z[t],
   x[0] == 1, y[0] == 1, z[0] == 1}, {x[t], y[t], z[t]}, {t, 0, tend}]
```

Out[3148]=

```
{{x[t] → InterpolatingFunction[  Domain: {{0., 1.}}
Output: scalar ][t],
  y[t] → InterpolatingFunction[  Domain: {{0., 1.}}
Output: scalar ][t],
  z[t] → InterpolatingFunction[  Domain: {{0., 1.}}
Output: scalar ][t]}}
```

In[3149]:=

```
plotX = Plot[x[t] /. sol, {t, 0, tend}, PlotStyle → Black];
plotY = Plot[y[t] /. sol, {t, 0, tend}, PlotStyle → Black];
plotZ = Plot[z[t] /. sol, {t, 0, tend}, PlotStyle → Black];
h = 0.05; t0 = 0; tend = 1;
grid = Table[i, {i, t0, tend, h}];
gridS = Table[i, {i, t0, 0.35, 0.005}]
```

Out[3152]=

```
{0., 0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05, 0.055, 0.06,
 0.065, 0.07, 0.075, 0.08, 0.085, 0.09, 0.095, 0.1, 0.105, 0.11, 0.115, 0.12,
 0.125, 0.13, 0.135, 0.14, 0.145, 0.15, 0.155, 0.16, 0.165, 0.17, 0.175,
 0.18, 0.185, 0.19, 0.195, 0.2, 0.205, 0.21, 0.215, 0.22, 0.225, 0.23, 0.235,
 0.24, 0.245, 0.25, 0.255, 0.26, 0.265, 0.27, 0.275, 0.28, 0.285, 0.29,
 0.295, 0.3, 0.305, 0.31, 0.315, 0.32, 0.325, 0.33, 0.335, 0.34, 0.345, 0.35}
```

In[3153]:=

```
runge = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/runge.txt", "Table"];
rungestep = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/runge_step.txt", "Table"];

expeuler = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/explicit_euler.txt", "Table"];
impeuler = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/implicit_euler.txt", "Table"];
symmetric = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/symmetric.txt", "Table"];
adams = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/adams_bashforth.txt", "Table"];
predictor = Import[
   "/Users/ivandybko/Projects/Numerical_methods/Mathematical physics/Lab1
     (Numerical methods for solving ordinary differential
     equations)/data/test3/adams_bashforth_with_predictor_corrector.txt",
   "Table"];
```

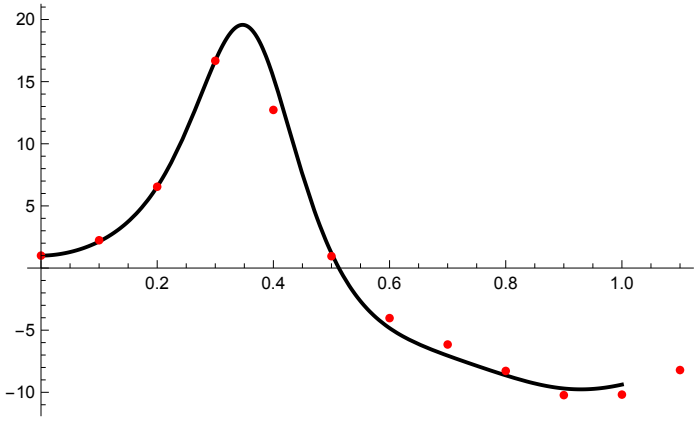
In[3160]:=

```
plotRungeX =
 ListPlot[Transpose[{runge[[All, 1]], runge[[All, 2]]}], PlotStyle → Red];
plotRungeY = ListPlot[Transpose[{runge[[All, 1]], runge[[All, 3]]}],
  PlotStyle → Red];
plotRungeZ =
  ListPlot[Transpose[{runge[[All, 1]], runge[[All, 4]]}], PlotStyle → Red];
plotRungeXstep =
 ListPlot[Transpose[{rungestep[[All, 1]], rungestep[[All, 2]]}], PlotStyle → Red];
plotRungeYstep = ListPlot[
  Transpose[{rungestep[[All, 1]], rungestep[[All, 3]]}], PlotStyle → Red];
plotRungeZstep =
  ListPlot[Transpose[{rungestep[[All, 1]], rungestep[[All, 4]]}], PlotStyle → Red];
plotExpeulerX = ListPlot[Transpose[{grid, expeuler[[All, 1]]}], PlotStyle → Red];
plotExpeulerY = ListPlot[Transpose[{grid, expeuler[[All, 2]]}], PlotStyle → Red];
plotExpeulerZ = ListPlot[Transpose[{grid, expeuler[[All, 3]]}], PlotStyle → Red];
plotImpeulerX = ListPlot[Transpose[{grid, impeuler[[All, 1]]}], PlotStyle → Red];
plotImpeulerY = ListPlot[Transpose[{grid, impeuler[[All, 2]]}], PlotStyle → Red];
plotImpeulerZ = ListPlot[Transpose[{grid, impeuler[[All, 3]]}], PlotStyle → Red];
plotSymmetricX = ListPlot[Transpose[{grid, symmetric[[All, 1]]}], PlotStyle → Red];
plotSymmetricY =
 ListPlot[Transpose[{grid, symmetric[[All, 2]]}], PlotStyle → Red];
plotSymmetricZ =
  ListPlot[Transpose[{grid, symmetric[[All, 3]]}], PlotStyle → Red];
plotAdamsX = ListPlot[Transpose[{grid, adams[[All, 1]]}], PlotStyle → Red];
plotAdamsY = ListPlot[Transpose[{grid, adams[[All, 2]]}], PlotStyle → Red];
plotAdamsZ = ListPlot[Transpose[{grid, adams[[All, 3]]}], PlotStyle → Red];
plotPredictorX = ListPlot[Transpose[{grid, predictor[[All, 1]]}], PlotStyle → Red];
plotPredictorY =
 ListPlot[Transpose[{grid, predictor[[All, 2]]}], PlotStyle → Red];
plotPredictorZ =
  ListPlot[Transpose[{grid, predictor[[All, 3]]}], PlotStyle → Red];
```

In[3174]:=

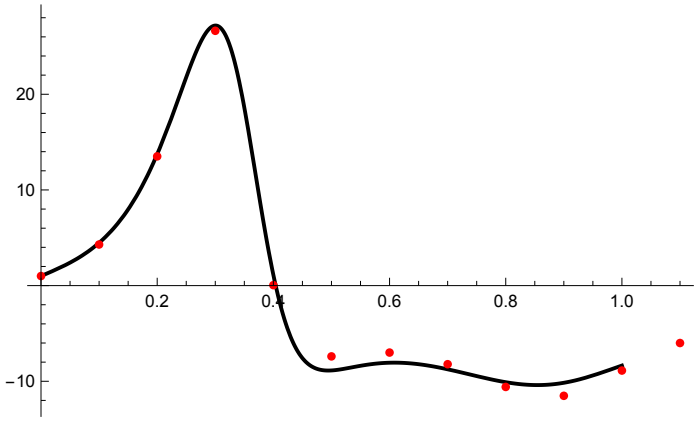
```
Show[plotX, plotRungeX, PlotRange → All]
```

Out[3174]=



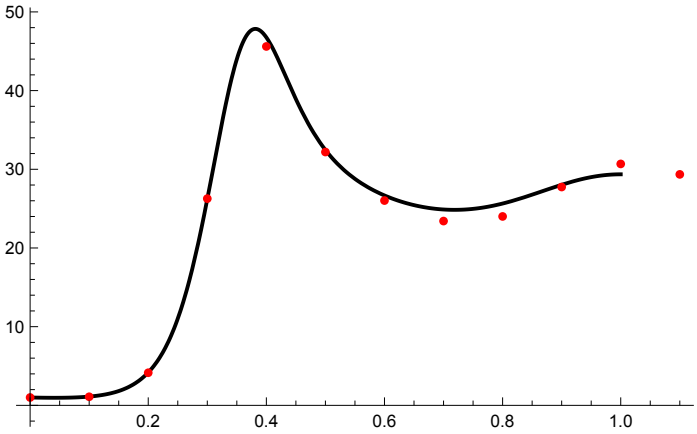
In[3175]:=
```
Show[plotY, plotRungeY, PlotRange → All]
```

Out[3175]=



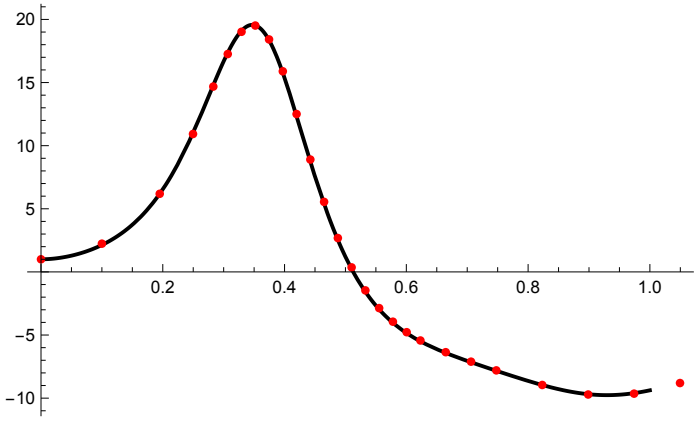
In[3176]:=
```
Show[plotZ, plotRungeZ, PlotRange → All]
```

Out[3176]=

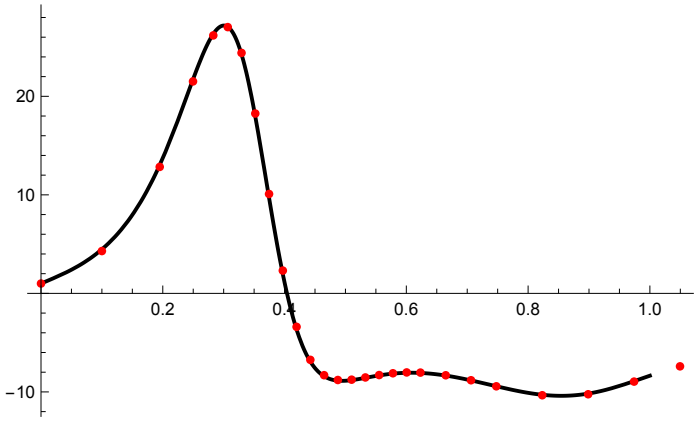


In[3177]:=
```
Show[plotX, plotRungeXstep, PlotRange → All]
```

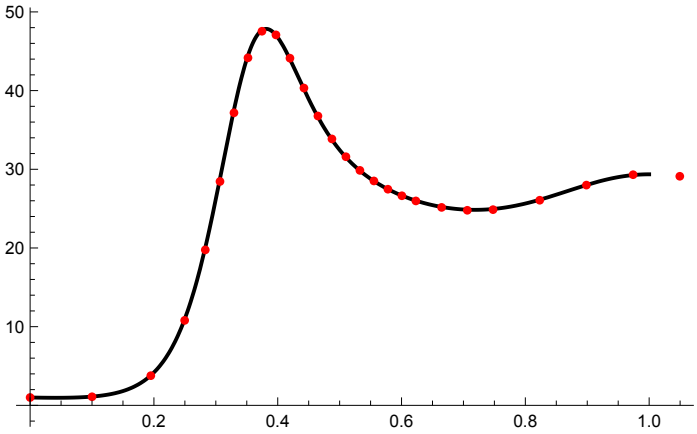
Out[3177]=



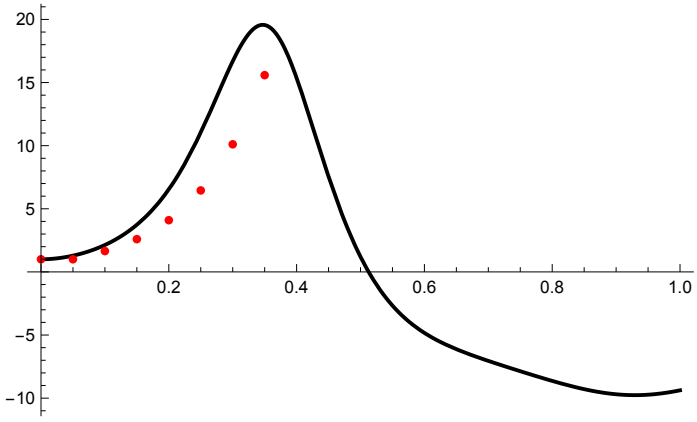
In[3178]:=
```
Show[plotY, plotRungeYstep, PlotRange → All]
```
Out[3178]=



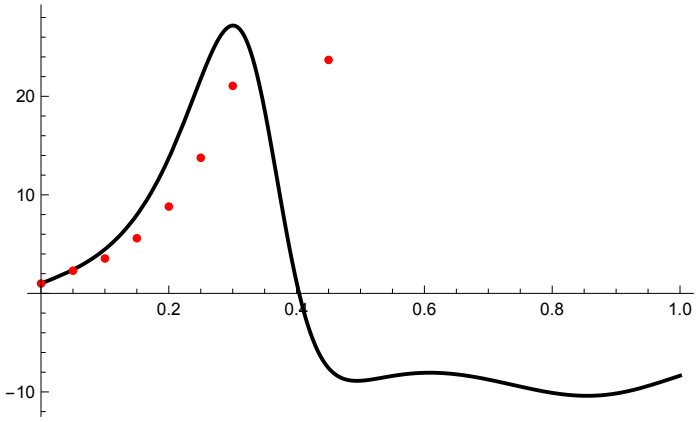
In[3179]:=
```
Show[plotZ, plotRungeZstep, PlotRange → All]
```
Out[3179]=



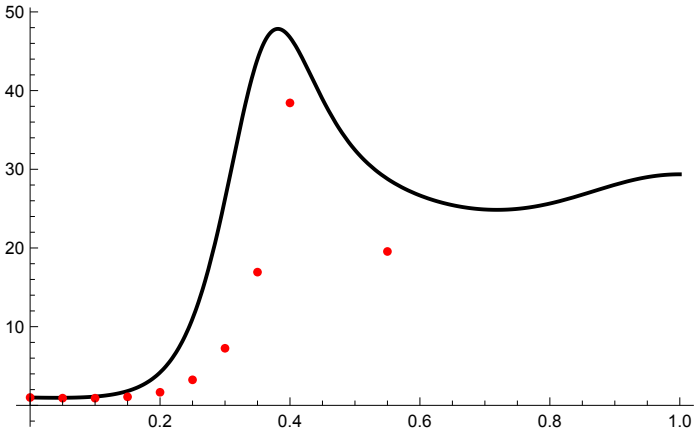
In[3180]:=
```
Show[plotX, plotExpeulerX]
```
Out[3180]=



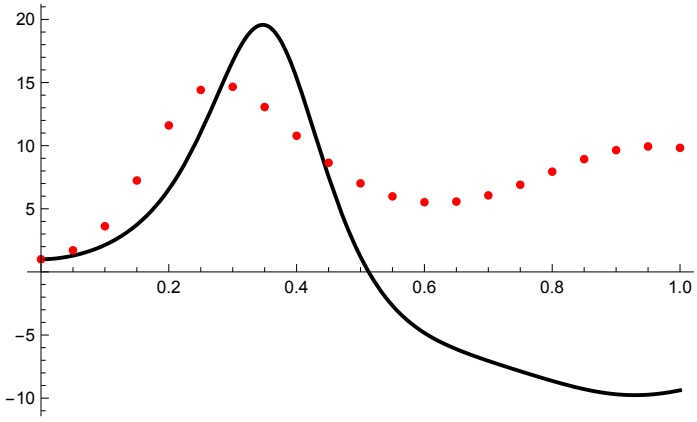
In[3181]:=
```
Show[plotY, plotExpeulerY]
```
Out[3181]=



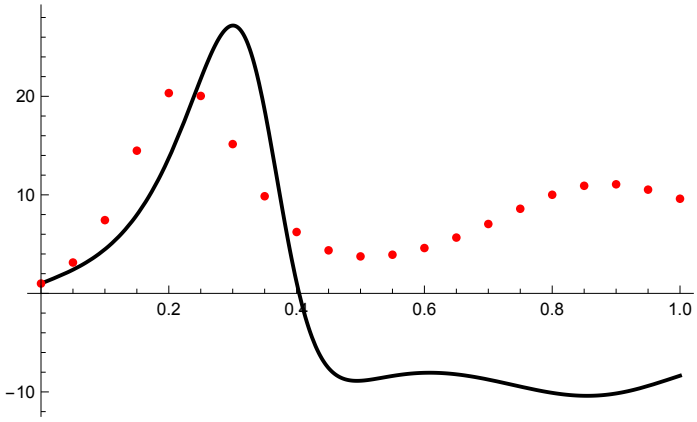
In[3182]:=
```
Show[plotZ, plotExpeulerZ]
```
Out[3182]=



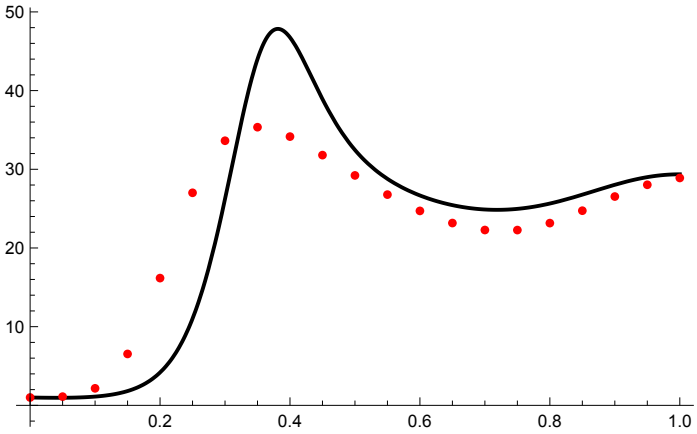
In[3183]:=
```
Show[plotX, plotImpeulerX, PlotRange → All]
```
Out[3183]=



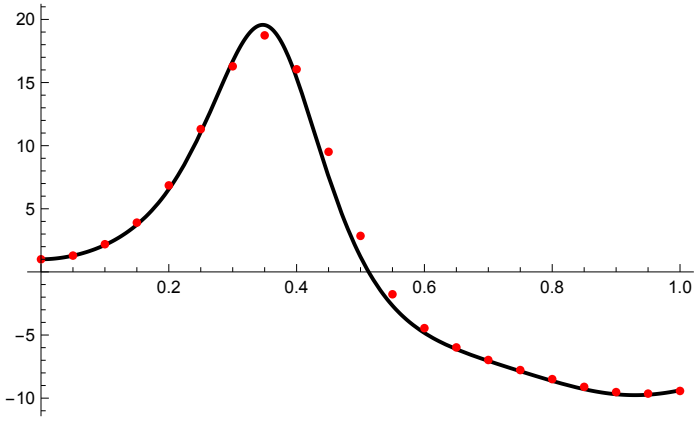
In[3184]:=
```
Show[plotY, plotImpeulerY, PlotRange → All]
```
Out[3184]=



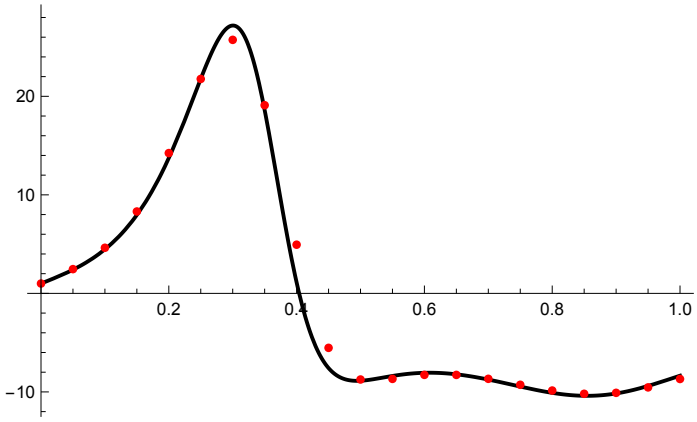
In[3185]:=
```
Show[plotZ, plotImpeulerZ, PlotRange → All]
```
Out[3185]=



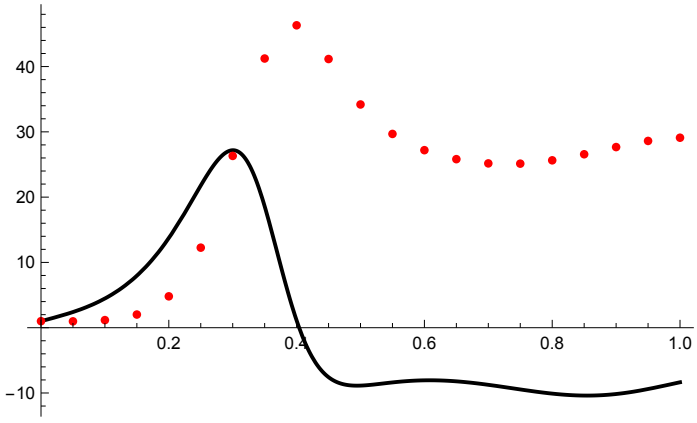
In[3186]:=
```
Show[plotX, plotSymmetricX, PlotRange → All]
```
Out[3186]=



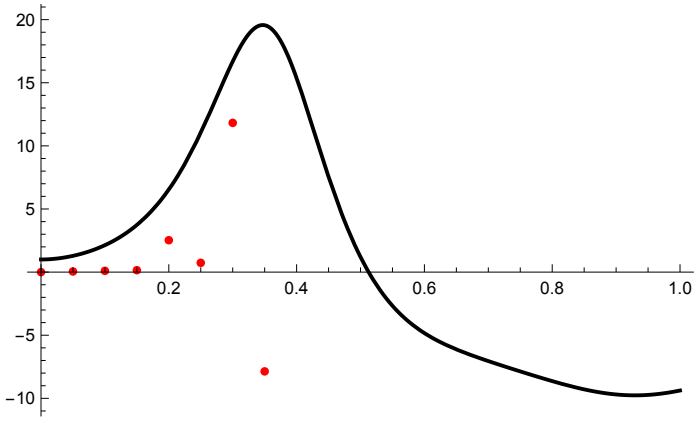
In[3187]:=
```
Show[plotY, plotSymmetricY, PlotRange → All]
```
Out[3187]=



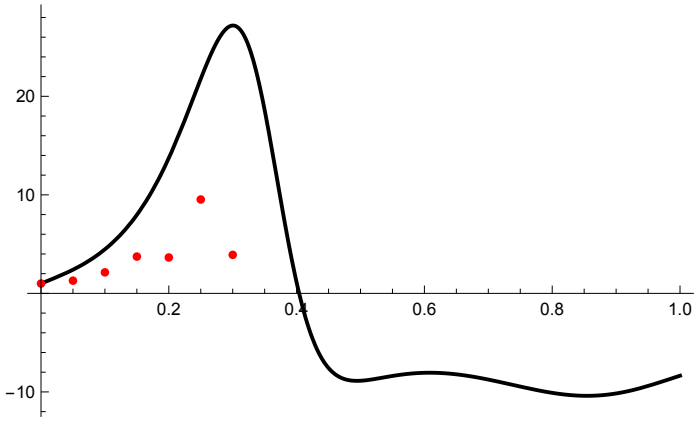
In[3188]:=
```
Show[plotY, plotSymmetricZ, PlotRange → All]
```
Out[3188]=



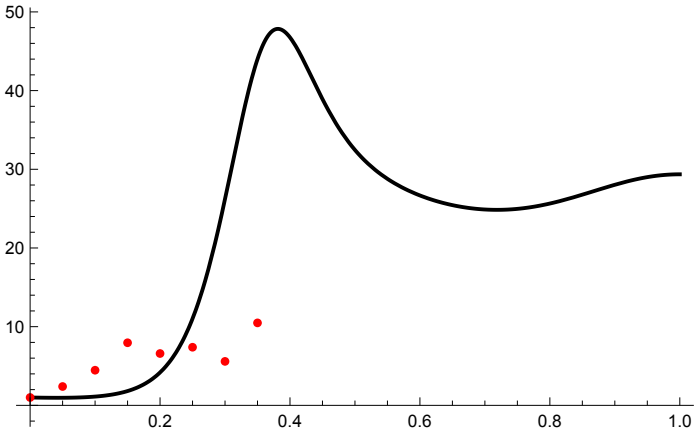
In[3189]:=
```
Show[plotX, plotAdamsX]
```
Out[3189]=



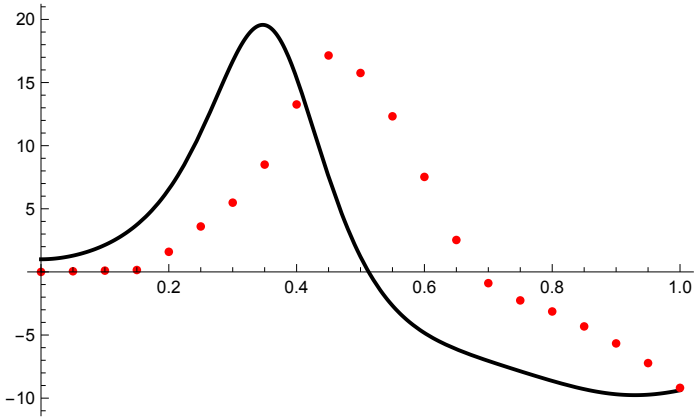
In[3190]:=
```
Show[plotY, plotAdamsY]
```
Out[3190]=



In[3191]:=
```
Show[plotZ, plotAdamsZ]
```
Out[3191]=

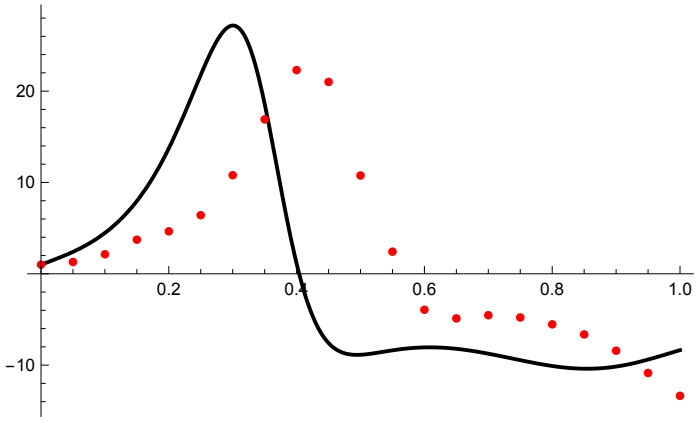


In[3192]:=
```
Show[plotX, plotPredictorX, PlotRange → All]
```
Out[3192]=



In[3193]:=
```
Show[plotY, plotPredictorY, PlotRange → All]
```

Out[3193]=



In[3194]:=
```
Show[plotZ, plotPredictorZ, PlotRange → All]
```

Out[3194]=