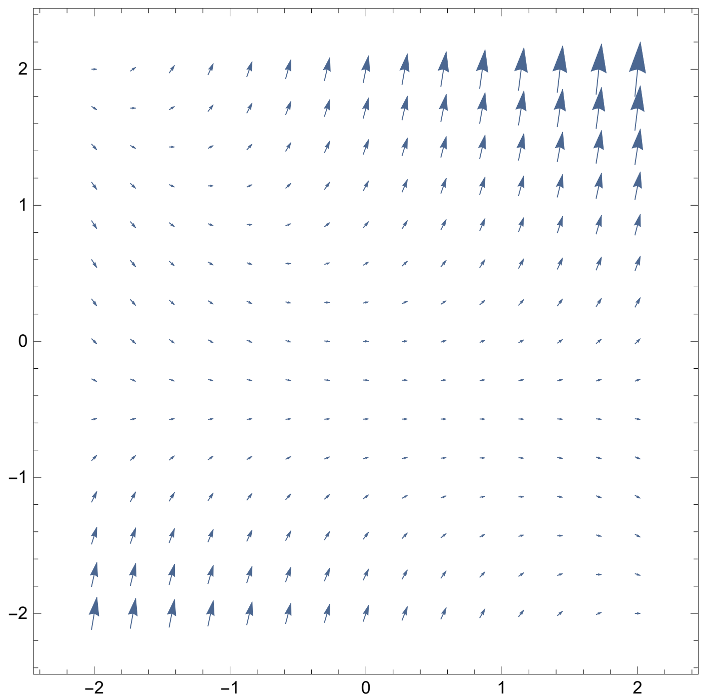
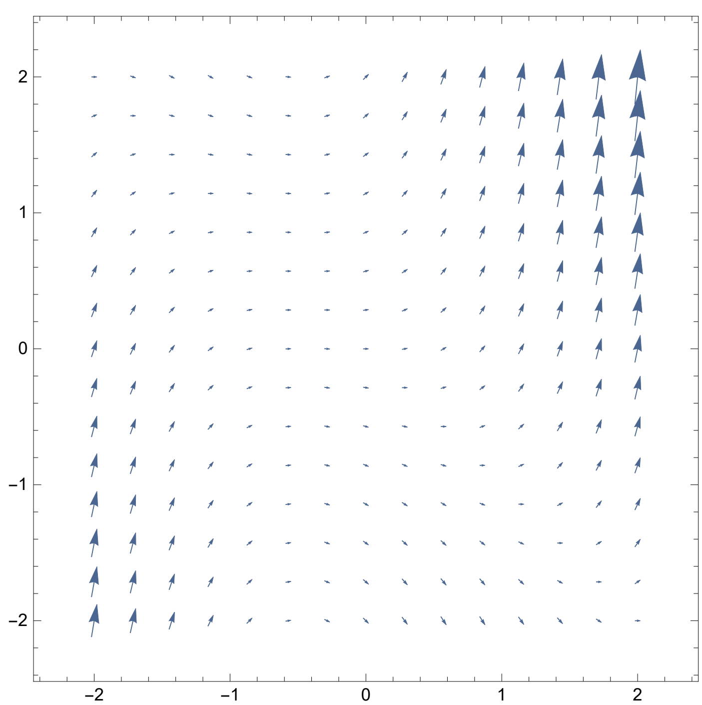
```
f[x_, t_] := (x + 1 / 2) * (x + t)
VectorPlot[{1, f[x, t]}, {t, -2, 2}, {x, -2, 2}]
```

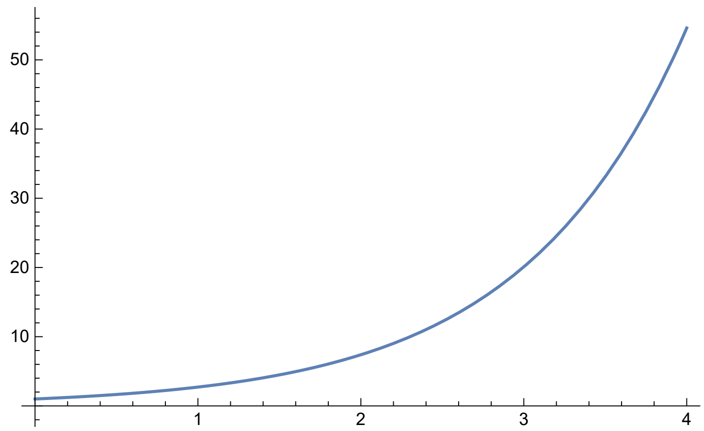


```
f[x_, t_] := (x + 1 / 2) * (x + t)
VectorPlot[{1, f[x, t]}, {x, -2, 2}, {t, -2, 2}]
```



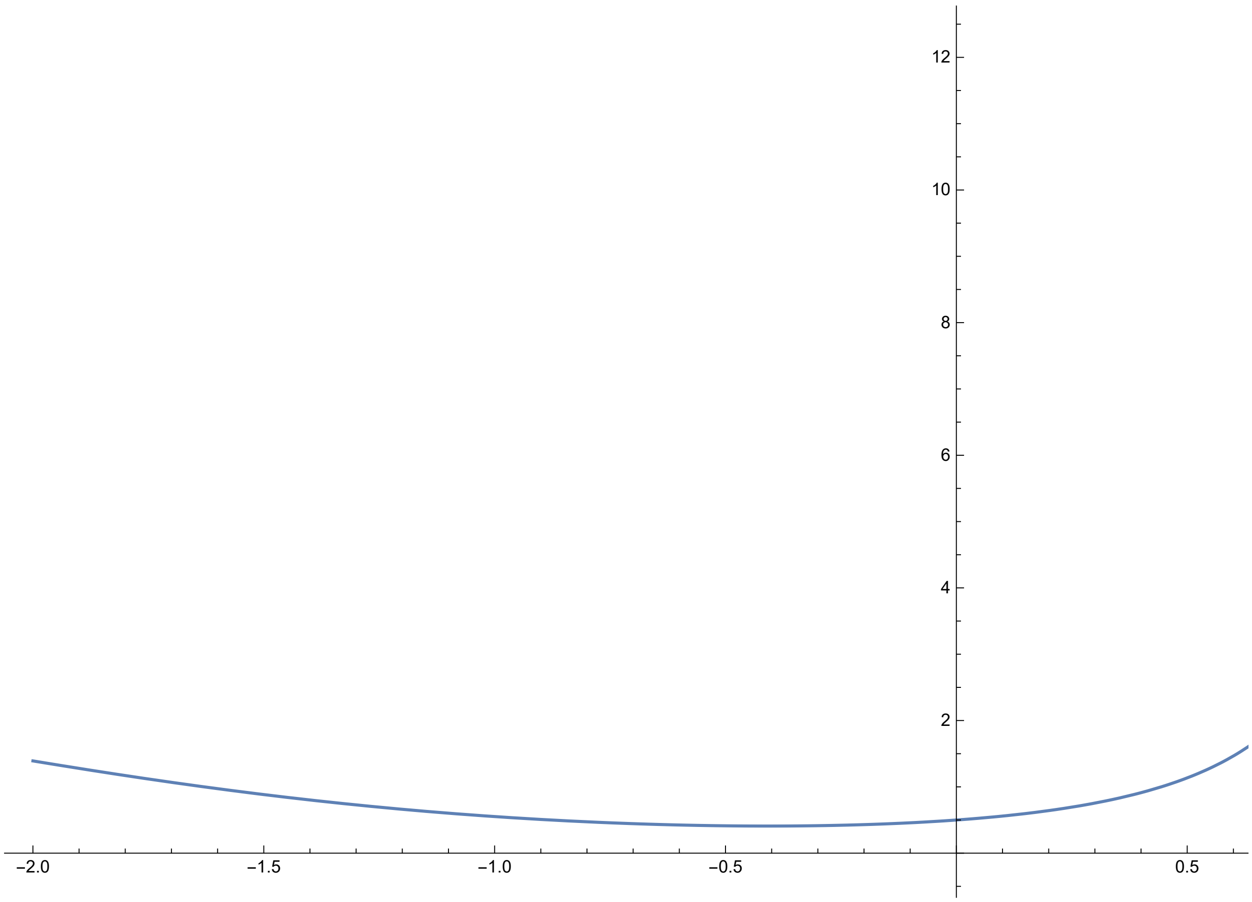
```
g[y_, t_] := y
s = NDSolve[{y'[t] == g[y[t], t], y[0] == 1}, y, {t, 0, 4}]
Plot[Evaluate[y[t] /. s], {t, 0, 4}, PlotRange → All]
```

```
{{y → InterpolatingFunction[ Domain: {{0., 4.}} Output: scalar ]}}
```



```
s = NDSolve[{y'[t] == (y[t] + 0.5) * (y[t] + t), y[0] == 0.5}, y, {t, -2, 1}]
Plot[Evaluate[y[t] /. s], {t, -2, 1}, PlotRange → All]
```

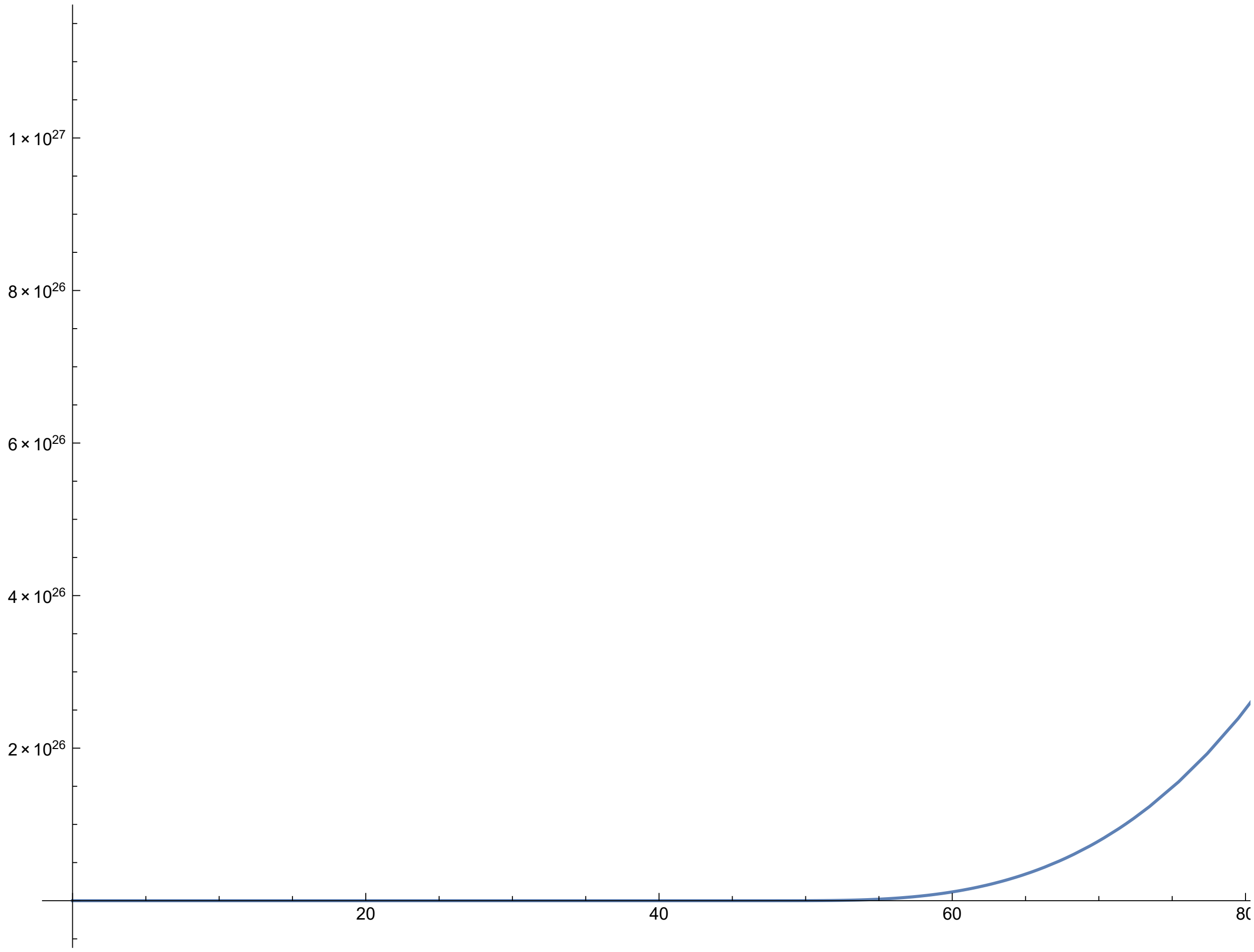
```
{{y → InterpolatingFunction[ Domain: {{-2., 1.}}
                             Output: scalar ]}}
```



```
M = 100
s = NDSolve[{y'[t] == y[t] * (1 - M / y[t]), y[0] == 110}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

```
100
```

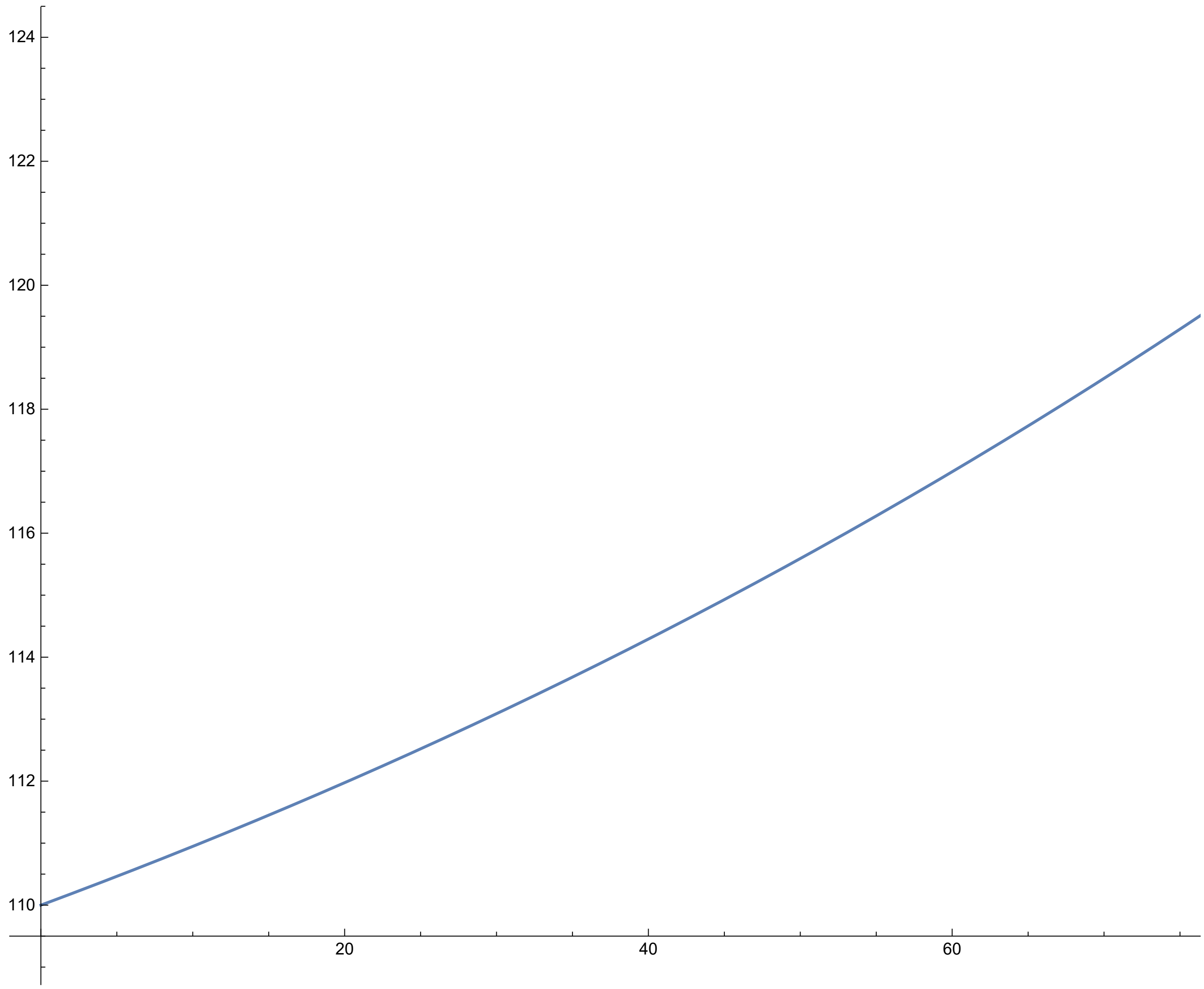
```
{{y → InterpolatingFunction[ Domain: {{0., 50.}} Output: scalar ]}}
```



```
M = 100
s = NDSolve[{y'[t] == (1 - M / y[t]), y[0] == 110}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

```
100
```

```
{{y → InterpolatingFunction[ Domain: {{0., 50.}} Output: scalar ]}}
```



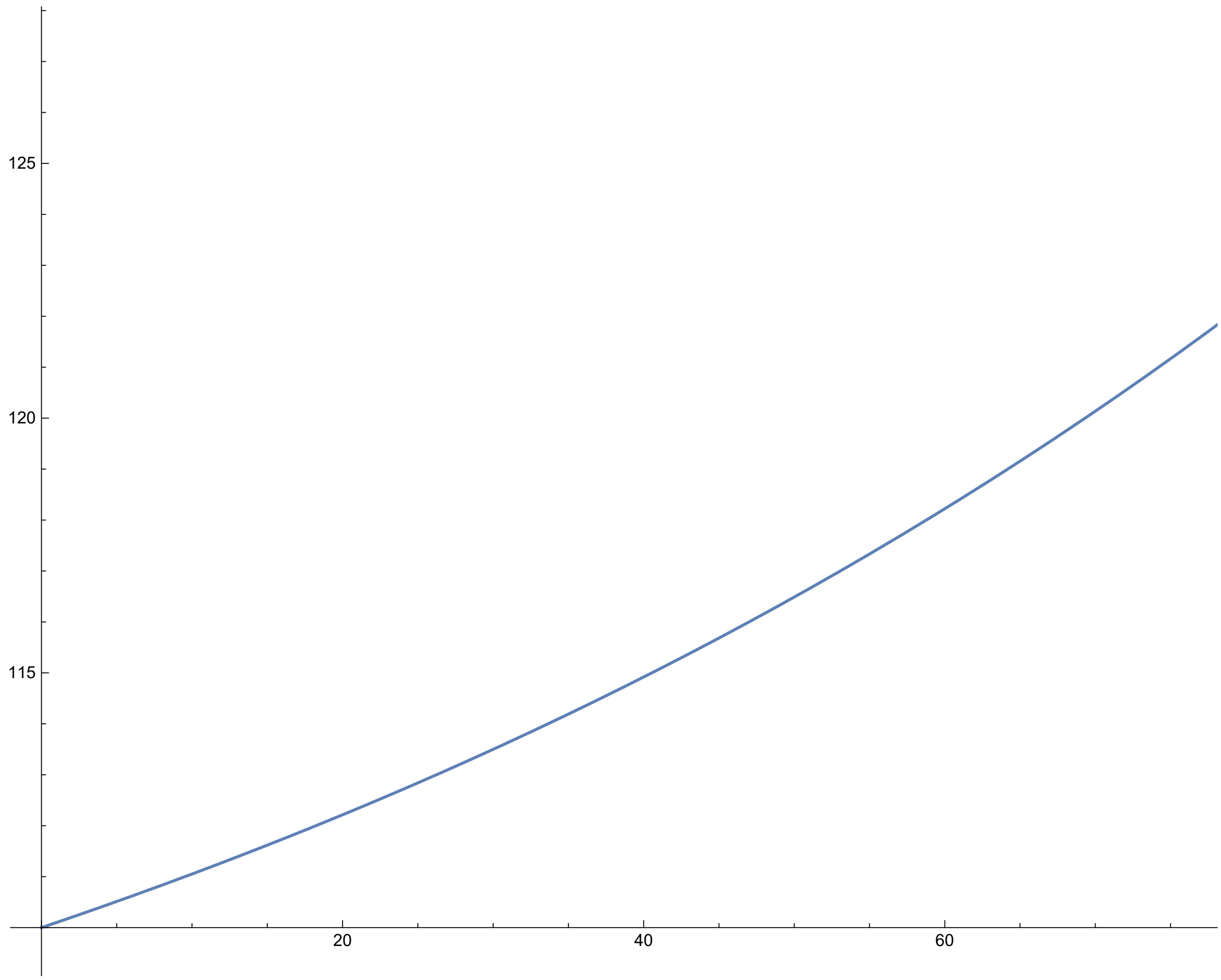
NDSolve::deqn :

Equation or list of equations expected instead of $\left(\frac{1}{2} + y[t]\right)(t + y[t])$ in the first argument $\left\{\left(\frac{1}{2} + y[t]\right)(t + y[t]), y[0] == \frac{1}{2}\right\}$. >>

```
M = 100
s = NDSolve[{y'[t] == (y[t] / M - 1), y[0] == 110}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

100

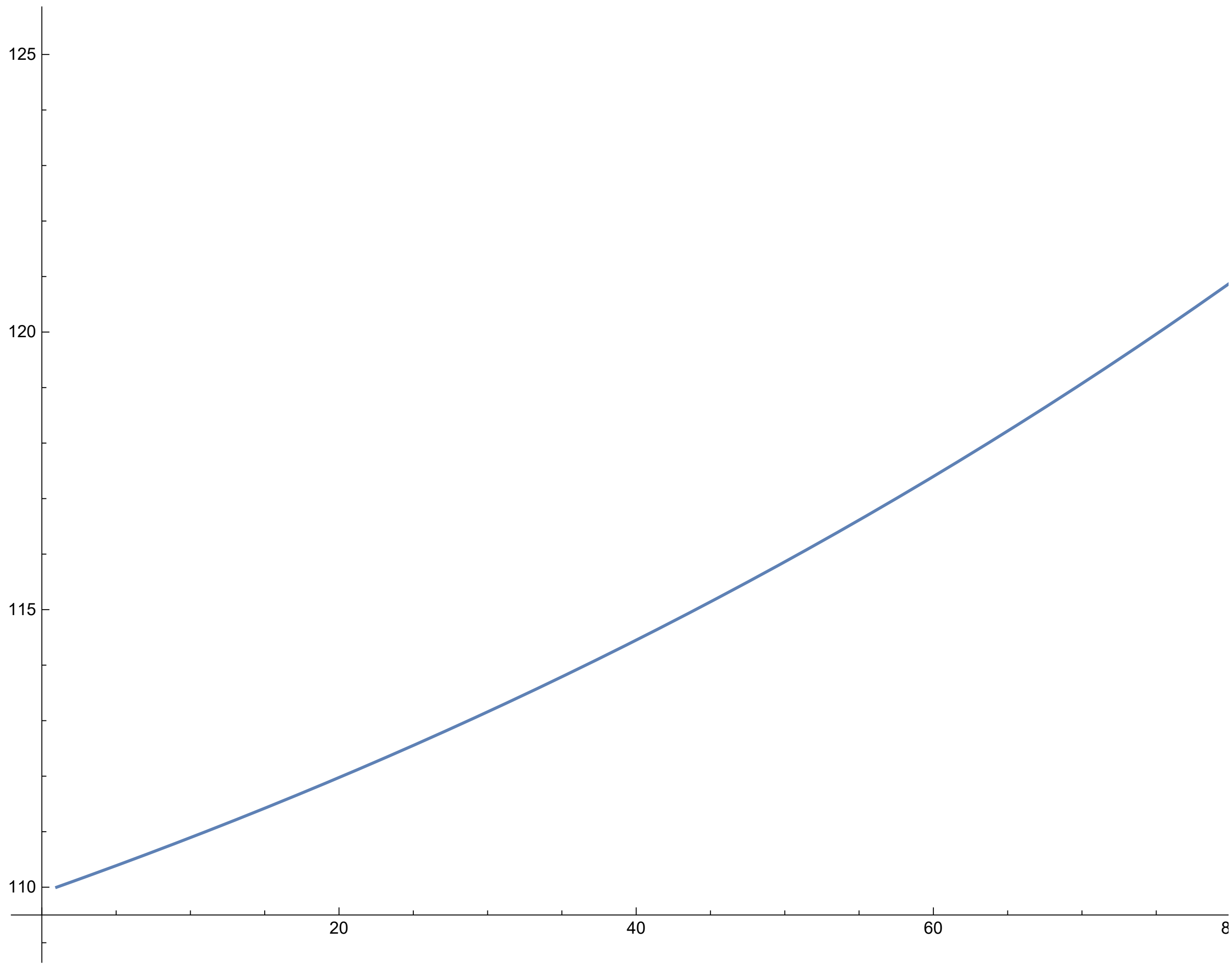
{{y → InterpolatingFunction[Domain: {{0., 50.}} Output: scalar]}}



```
M = 100
s = NDSolve[{y'[t] == Log[y[t] / M], y[1] == 110}, y, {t, 1, 50}]
Plot[Evaluate[y[t] /. s], {t, 1, 100}, PlotRange → All]
```

```
100
```

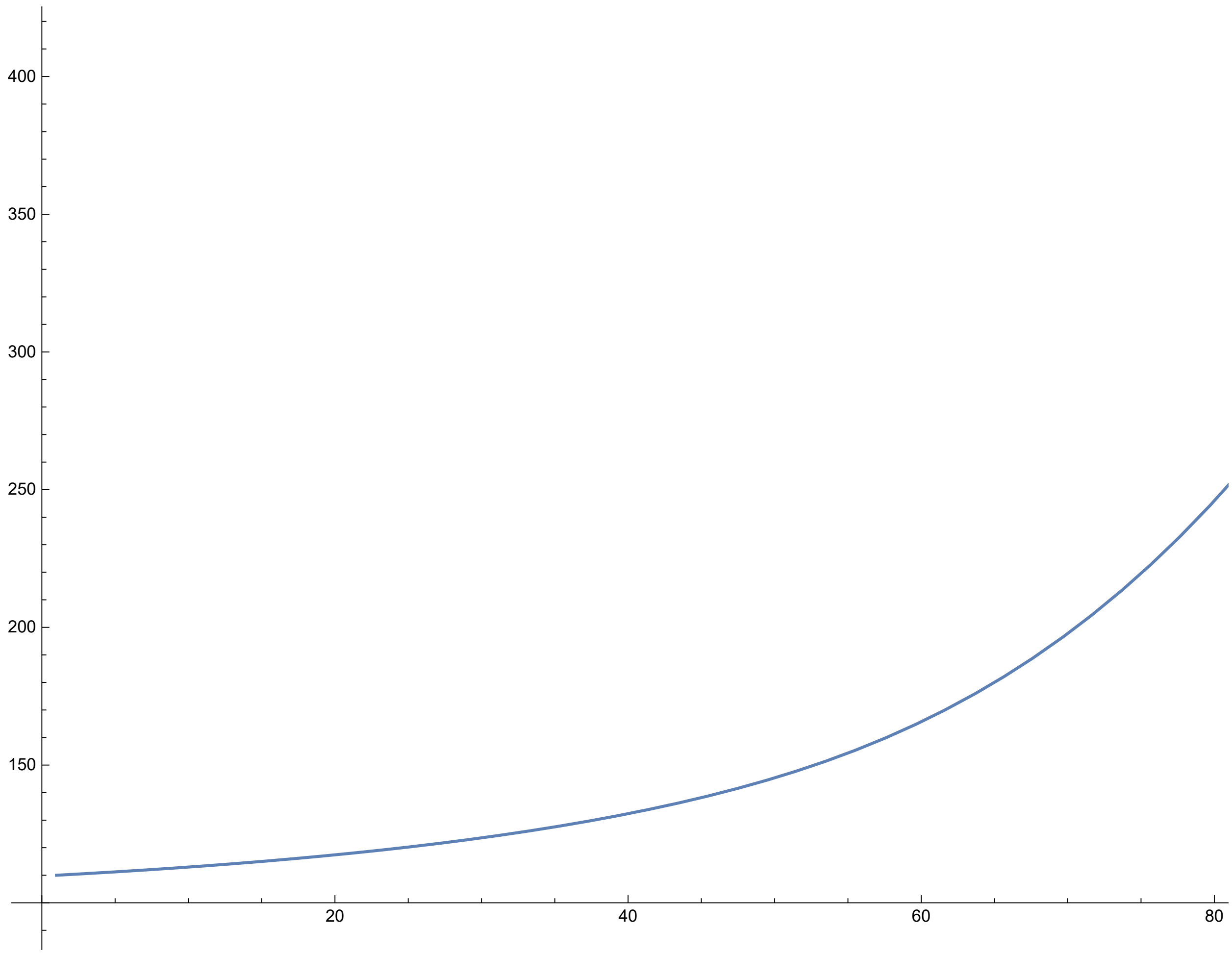
{{y → InterpolatingFunction[Domain: {{1., 50.}} Output: scalar]}}



```
M = 100
s = NDSolve[{y'[t] == Exp[y[t] / M] * Log[y[t] / M], y[1] == 110}, y, {t, 1, 50}]
Plot[Evaluate[y[t] /. s], {t, 1, 100}, PlotRange → All]
```

100

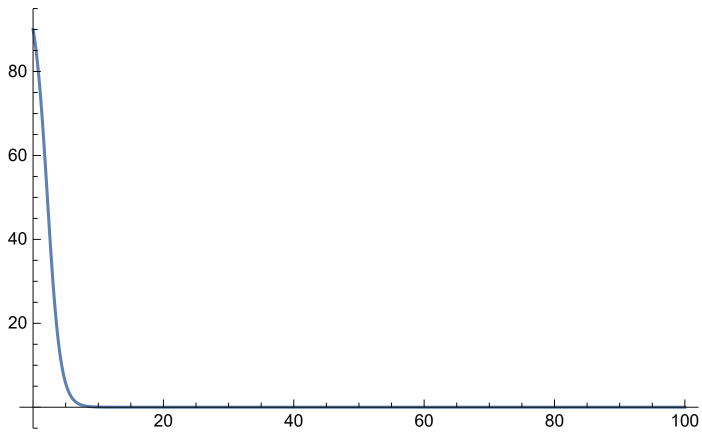
{{y → InterpolatingFunction[Domain: {{1., 50.}} Output: scalar]}}



```
M = 100
s = NDSolve[{y'[t] == y[t] * (y[t] / M - 1), y[0] == 90}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

```
100
```

```
{{y → InterpolatingFunction[ Domain: {{0., 50.}}
                             Output: scalar ]}}
```

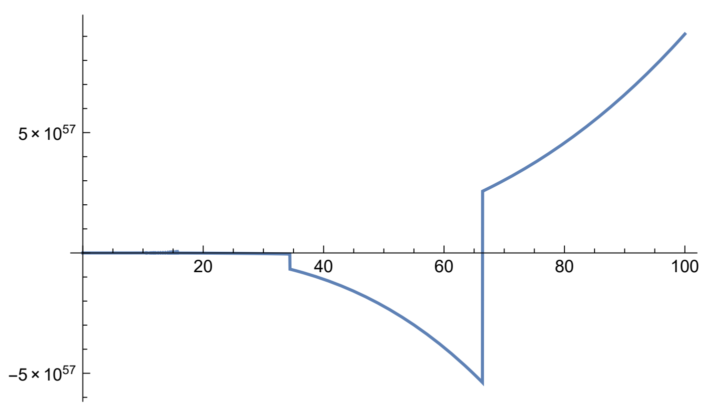



```
M = 100
s = NDSolve[{y'[t] == y[t] * (y[t] / M - 1), y[0] == 110}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

```
100
```

NDSolve::ndsz : At t == 2.397894560990232`, step size is effectively zero; singularity or stiff system suspected. ≫

```
{{y → InterpolatingFunction[ Domain: {{0., 2.4}}
                             Output: scalar ]}}
```



```
s = NDSolve[
  {y'[t] == 0.5 * y[t] * (1 - y[t] / 500) * (y[t] / 100 - 1), y[0] == 1100}, y, {t, 0, 50}]
Plot[Evaluate[y[t] /. s], {t, 0, 100}, PlotRange → All]
```

```
{{y → InterpolatingFunction[ Domain: {{0., 50.}} Output: scalar ]}}
```