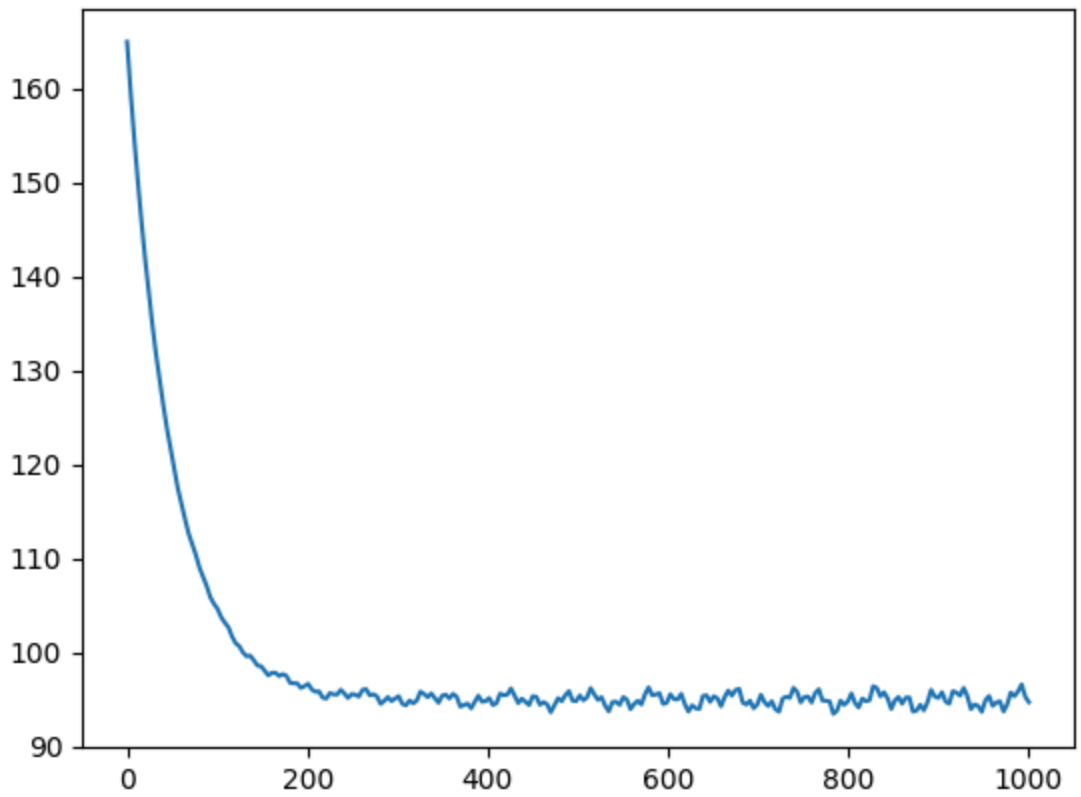
The oscillation-resistant algorithm for determining convergence is detailed here together with an example.

First, identify whether the time series is oscillating by identifying whether it has local extrema; if not, we can simply use the last data points for the termination condition. If it is oscillating, two averages are determined over a larger time span, based on which the termination criterion can be calculated. For a sufficiently large time span covering many periods, this eliminates the oscillating part of the signal.

Since a priori the period is not known, the time span starts out at twice the difference between extrema found and is then increased with each failure to achieve termination. This is limited arbitrarily to a fifth of the time series as not to include the initial part which necessarily shouldn't be oscillating around a steady value; once this is reached, the time span is reset to zero and built up again as the simulation progresses. This ensures that for a function decaying to a steady value overlaid with periodic and noisy data, an approximation to the function's derivative is eventually recovered; or rather within this notebook, an approximation to the steady value.




```
[<matplotlib.lines.Line2D at 0x7f490fd237d0>]
```

Error on converged value without noise: 6.033474164723884e-09

Error on converged value with multiple overlapping periodic noise sources: 0.00941221648260182