# Nonlinear Curve Fitting

Model Parameter Estimation with Least Squares

Joseph Barton

February 2026

**Abstract**

This document demonstrates nonlinear least-squares curve fitting using `scipy.optimize.curve_fit` to estimate parameters of physical models from noisy data. Example models include exponential decay, Gaussian peaks, and custom functions relevant to physics experiments. The code computes best-fit parameters, uncertainties, covariance matrix, and goodness-of-fit statistics ($\chi^2$, reduced $\chi^2$, $R^2$). Visualizations show data, fit, and residuals. Source available in the repository. Link at end of paper.

## 1 Introduction

Curve fitting extracts meaningful physical parameters (decay rates, amplitudes, centers, etc.) from experimental or simulated data. When the relationship is nonlinear in the parameters, analytic solutions do not exist, so we use numerical optimization to minimize the sum of squared residuals.

This project accompanies a Python implementation that fits one or more model functions to synthetic or loaded data, reports parameter uncertainties via the covariance matrix, and assesses fit quality.

## 2 Theoretical Background

Given data points $(x_i, y_i)$ with uncertainties $\sigma_i$ (often assumed constant or estimated), we seek parameters $\vec{\beta}$ that minimize the weighted sum of squared residuals:

$$\chi^2(\vec{\beta}) = \sum_{i=1}^{N} \left( \frac{y_i - f(x_i; \vec{\beta})}{\sigma_i} \right)^2, \tag{1}$$

where $f(x; \vec{\beta})$ is the model function.

For unweighted fits ($\sigma_i = 1$), this reduces to ordinary least squares. The Levenberg–Marquardt algorithm (used by `curve_fit`) efficiently solves this nonlinear optimization problem and returns: - Best-fit parameters $\vec{\beta}$, - Covariance matrix $\mathrm{Cov}(\vec{\beta}) \to$ standard errors $\sigma_{\beta_j} = \sqrt{\mathrm{Cov}_{jj}}$, - Residuals and $\chi^2_{\mathrm{red}} = \chi^2/(N-M)$ (M = number of parameters; $\approx 1$ indicates good fit assuming correct model and uncertainties).

Common models demonstrated:

- Exponential decay: $f(t) = Ae^{-t/\tau} + C$,
- Gaussian: $f(x) = A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + B$,
- Damped oscillator or custom physics functions.

## 3 Numerical Implementation

The fitting is performed with

```
popt, pcov = curve_fit(model_func, xdata, ydata, p0=initial_guesses,
                        sigma=uncertainties, absolute_sigma=True)
```

Initial guesses (`p0`) are crucial for convergence in nonlinear cases. Parameter uncertainties and correlations are extracted from `pcov`.

Key outputs and diagnostics: - Best-fit parameters with $1\sigma$ errors, - Reduced chi-squared ($\chi^2_{\mathrm{red}}$), - Coefficient of determination ($\mathrm{R}^2$), - Residual plot (should be random, no structure), - Confidence bands using covariance propagation.

## 4 Results and Interpretation

Fits to synthetic data (with added Gaussian noise) recover true parameters within uncertainties. For example: - Exponential decay recovers lifetime $\tau$ accurately, - Gaussian fits yield precise peak position $\mu$ and width $\sigma$, - Residuals confirm model adequacy or highlight systematic deviations.

This approach is directly applicable to real experimental data (radioactive decay, spectroscopy, cooling curves, etc.).

## 5 Conclusion

Nonlinear curve fitting bridges data and physical insight, requiring careful model selection, initial guesses, uncertainty handling, and diagnostic checks. This project showcases robust use of SciPy optimization tools, statistical interpretation, and publication-quality plotting — essential skills for scientific computing and data-driven physics.

**Repository:** `https://github.com/jbarton727/Portfolio`