Download the template Jupyter notebook `HW7_Template.ipynb` from Canvas and work from that template. You should be able to complete this assignment with only `numpy` and `matplotlib`.

1. **CODING:** In this problem, you will learn how to perform linear regression using the matrix method. The file `Riggs.dat` contains hypothetical particle-physics spectral data (energy vs counts) of a search for a new particle called the *Riggs boson*. The columns are 1) energy E (GeV) and 2) counts N (number of events in that energy bin). The energy bins are independent (i.e., there is no covariance).

```
1.0   277
2.0   254
3.0   252
4.0   265
5.0   266
6.0   266
...
```

A signal of a new particle consists of a gaussian on top of a smooth background. Model the spectrum with a function given by:

$$N(E) = a + bE + cE^2 + Ae^{-(E-E_{\text{Riggs}})^2/(2\sigma_E^2)} \tag{1}$$

where a, b, c, and A are 4 fit parameters. The first three terms make up the background, the last term represents the signal. The term σ_E represents the intrinsic energy width of the particle and is assumed to be 2.6 GeV. Theory also predicts that $E_{\text{Riggs}} = 68.8$ GeV.

(a) (4 pts) Given our data and model, construct the design matrix $\mathbf{G}$, data covariance matrix $\mathbf{S}$ (note that we used $\mathbf{E}$ in class), and response vector $\mathbf{D}$.

(b) (4 pts) Use the matrix operations to solve for the best-fit parameter vector $\hat{\mathbf{A}}$ given by:

$$\hat{\mathbf{A}} = (\mathbf{G}^T\mathbf{S}^{-1}\mathbf{G})^{-\mathbf{1}}\mathbf{G}^T\mathbf{S}^{-1}\mathbf{D} \tag{2}$$

and their uncertainties.

(c) (2 pts) Plot the data and superimpose the best-fit model.

2. **EXTRA CREDIT:** (2 pts) A nice way to visualize the covariance matrix of a fit is to sample values of the parameters that would be consistent with the errors.

(a) First take the Cholesky decomposition of the covariance matrix: $LL^T = \text{Cov}(\hat{A})$

(b) Then make a vector of 4 zero-mean, unit-variance Gaussian random values:
$v = [X_0, X_1, X_2, X_3]^T; \quad X_i \leftarrow \mathcal{N}(0, 1)$

(c) Multiply them together using matrix math: $\delta\hat{\mathbf{A}} = Lv$

This gives you random deviations from the fitted parameter values $\hat{\mathbf{A}}$ according to the appropriate statistics given by the covariance matrix. You can now plot a bunch of models with parameters $\hat{\mathbf{A}} + \delta\hat{\mathbf{A}}$ to see the range of possible solutions that are plausible given the errors. Doing this with a small line width (`lw`) and making them semi-transparent (`alpha` < 1) provides a nice visualization of the set of possible lines.