a) Load data from `cw1a.mat`. Train a GP with a squared exponential covariance function, `covSEiso`. Start the log hyper-parameters at `hyp.cov = [-1 0]; hyp.lik = 0;` and minimize the negative log marginal likelihood. Show the 95% predictive error bars. Comment on the predictive error bars and the optimized hyperparameters.

b) Show that by initializing the hyperparameters differently, you can find a different local optimum for the hyperparameters. Try a range of values. Show the fit. Explain what is going on. Which fit is best, and why?

c) Train instead a GP with a periodic covariance function. Show the fit. Comment on the behaviour of the error-bars, compared to your fit from a). Do you think the data generating mechanism (apart from the noise) was really strictly periodic? Why, why not?

d) Generate 200 (essentially) noise free data points at `x = linspace(-5,5,200)';` from a GP with the following covariance function: `{@covProd, {@covPeriodic, @covSEiso}}`, with covariance hyperparameters `hyp.cov = [-0.5 0 0 2 0]`. In order to apply the Cholesky decomposition to the covariance matrix, you may have to add a small diagonal matrix, for example `1e-6*eye(200)`, why? Plot some sample functions. Explain their behaviour.

e) Load `cw1e.mat`. This data has 2-D input and scalar output. Visualise the data, for example using `mesh(reshape(x(:,1),11,11),reshape(x(:,2),11,11),reshape(y,11,11));` Rotate the data, to get a good feel for it. Compare two GP models of the data, one with `covSEard` covariance and the other with `{@covSum, {@covSEard, @covSEard}}` covariance. For the second model be sure to break symmetry with the initial hyperparameters (eg by using `hyp.cov = 0.1*randn(6,1);`).

Compare the models: How do the data fits compare? How do the marginal likelihoods compare? What is your interpretation? Which of the two is better?