Return m-files before 14.5.2013
Email the solution to :
`fysp120(at)gmail.com` Subject-line: `FNM exercise3`

1. Program `opti.m` uses Newton's method to optimize a function. The next point is $x' = x + d$, where the displacement d is solved from the linear set of equations

   $$Hd = -\nabla f$$

   where H is the Hessian and ∇f the gradient of the function f we optimize. As you see from the example, the problem with this method is that if the Hessian H is not positive definite, the next point x' may actually be worse, $f(x') > f(x)$, and `opti.m` turns out to maximize the function.
   Use the Levenberg-Marquardt method to stabilize the routine. Compute the eigenvalues E of the Hessian (`E=eig(H)`). Start with constant $a \geq 0$, solve

   $$(H + a\ diag(E))d = -\nabla f\ ,$$

   and check if the point $x' = x + d$ is better, that is, $f(x') < f(x)$. If not, increase a by about 0.1 and get a new x' to try. If x' is better, decrease a. Iterate to convergence, remember to keep $a \geq 0$.
   Summary:
   small $a \rightarrow$ fast downhill- may get lost or find a saddle point
   large $a \rightarrow$ slowly, but surely downhill.
   Idea behind the method:
   Think of a slightly modified method, replace $diag(E)$ with the unit matrix I. Since the operation $H + aI$ shifts the eigenvalues of H up by the constant a, this method ensures that with large enough a the iteration goes downhill. Here constant a introduces mixing of steepest descent (large a) and Newton's method ($a = 0$). The logic behind using eigenvalues E is to keep *some* information about the Hessian also for large a, so that the progress is better than that of a pure steepest descent: long steps to slowly decreasing directions, short to steeply descending ones. As you notice, it's a good idea to check that all eigenvalues E are positive.

   CONTINUES ON THE NEXT PAGE!

2. Solve numerically the differential equation

$$5y'' + 4y' + 5y = 0$$

with the boundary conditions $y'(0) = 10$ and $y(30) = 0$ in the range $x \in [0, 30]$. As result, plot the solution $y(x)$. You may freely choose the solution method; both self made and Matlab built-in methods are fine.

HINTS

Optimization:

- As such, `opti.m` fails to find the minimum - it finds a saddle point.
- Your task is to implement the Levenberg-Marquardt improvement, and add a `while`-loop, where you increase `a` and solve a new `p` until $f(x + p) < f(x)$. Note: Solve `E=eig(H)` *before* you start this `while`- loop, no need to solve it over and over again.
- Starting from point (-1.5,1.5), the minimum

  $$f(-1.72764202, 1.72764202) = 0$$

  should be reached in about 8 steps.
- The variable `a` appears already in `opti.m`, though not used yet.

Differential equation (DE):

- The DE is second order, so you need to break it to two coupled first order DE's. See the lecture notes how this is done.
- You have two boundary values for the solution.
  - option 1 Use Matlab function `bvp4c`, which uses a "multiple-shooting method" - take a good look at Example 1 in Matlab help (type `doc bvp4c`).
  - option 2 Use a more common solver, such as `ode45` . Write a loop that tries to find a starting point $y(0)$ so that $y(30) = 0$. First, find two values of $y(0)$, say a and b, that give positive and negative $y(30)$, Then you can be sure that $a \leq y(0) \leq b$ will have $y(30) = 0$. Use Newton's bisection method to pinpoint the solution more accurately.
- The solution should have $y(0) \approx 19$.