Faculty of Arts and Sciences
Department of Computer Science
CMPS 351 – Numerical Optimization
Assignment 3 – Due Feb 11, 2019

Reading Material:

- Chapters 6 and 10 of Ref. 1 (Nocedal and Wright)

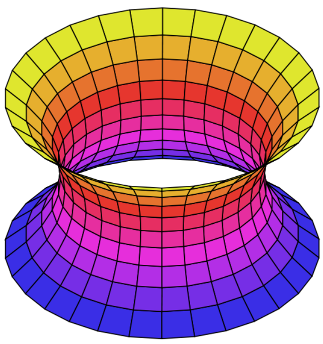
1. Minimal Surface. Consider the problem of generating a shape with minimal surface area. The shape is a surface of revolution about a vertical axis of length l (see figure on back page). Because of circular symmetry the problem reduces to finding the radii of horizontal cross sections. The radii of the cross sections are fixed on the top and bottom at r_0. We can discretize the problem so the unknowns x represent the radii at n locations equally spaced along the vertical axis at a distance $h = l/(n+1)$. With this discretization the surface area may be approximated by the following function:

$$f(x) = 2\pi h x_1 \left[1 + \left(\frac{x_1 - r_0}{h}\right)^2\right]^{1/2} + 2\pi h \sum_{i=1}^{n-1} x_i \left[1 + \left(\frac{x_{i+1} - x_i}{h}\right)^2\right]^{1/2} + 2\pi h x_n \left[1 + \left(\frac{r_0 - x_n}{h}\right)^2\right]^{1/2}$$

Note the structure of the function: $f(x) = f_1(x_1) + f_2(x_1, x_2) + f_3(x_2, x_3) + \cdots f_n(x_{n-1}, x_n) + f_{n+1}(x_n)$.

Use Newton's method with a line search to find the optimal shape for $r_0 = 1$, $l = 0.75$, and $n = 20$.

- write a routine that computes the objective function.
- write a routine that computes the gradient.
- write a routine that computes the Hessian. Notice that the objective function consists of the sum of terms where each term contains only a pair of adjacent variables. This gives rise to a Hessian that has a tridiagonal structure.
- use your Newton routine with a backtracking line search to find the optimal shape starting from a cylindrical initial shape (i.e. $x_i = r_0$ for all i).
- plot the convergence behavior.



2. Gauss-Newton. Nonlinear Least Squares (NLLS) problems arise in a large number of practical scientific and engineering contexts, and represent an important class of numerical optimization problems. NLLS problems commonly arise when trying to fit a model to a measured data set in a way that minimizes the discrepancy between model predictions and measured data. The simplest objective to minimize is the sum of the squares of the discrepancies at the measured data points.

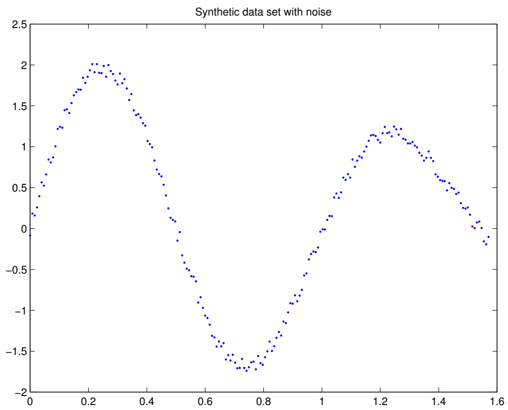
Consider the NLLS problem of fitting a model with four parameters ($n = 4$) of the form

$$\phi(a;\ t) = a_1 \sin(2\pi a_2 t) + a_3 \sin(2\pi a_4 t)$$

to a given data set (t_i, y_i) such as the one shown below ($m = 200$).

```python
m = 200
t = np.linspace(0, np.pi/2, m)
p = lambda t: 0.8*np.sin(2*np.pi*1.15*t) + 1.2*np.sin(2*np.pi*0.9*t)
noise = 0.2 * np.random.uniform(-0.5, 0.5, m)
y = p(t) + noise       # this is a comment α
plt.scatter(t, y, s=3)
plt.title('Synthetic data set with noise')
plt.show()
```

- Write a routine that returns the objective function $f(a) = \frac{1}{2}r(a)^t r(a)$ where $r(a)$ is an $m \times 1$ residual vector with entries $r_i(a) = \phi(a;\ t_i) - y_i$.
- Write a routine that returns the gradient of the objective function $\nabla f(a) = J(a)^t\, r(a)$ where $J_{m\times n}$ is the Jacobian of r with entries $J_{ij} = \frac{\partial f_i}{\partial a_j}$
- Write a routine that returns $H(a) = J(a)^t J(a)$, the Gauss-Newton approximation of the Hessian $\nabla^2 f(a)$ of the objective function.
- Use a Gauss-Newton method to fit the data set above using the starting point $a^0 = [1.0\ 1.0\ 1.0\ 1.0]^t$.



3. Quasi-Newton Methods. When the Hessian of the objective is not available, is too expensive, or is too cumbersome to compute, an approximation of it may be obtained by a low-rank update at every iteration. The resulting methods are known as quasi-Newton methods. Among the many quasi-Newton methods, the BFGS method is perhaps the most popular. It is a rank 2 update which may be written as:

$$B^{k+1} = B^k - \frac{y^k (y^k)^t}{(y^k)^t s^k} - \frac{B^k s^k (s^k)^t B^k}{(s^k)^t B^k s^k}$$

where where $s^k = x^{k+1} - x^k$ and $y^k = \nabla f^{\,k+1} - \nabla f^{\,k}$.

- Describe very briefly the insight into this formula and how we might derive it.
- From a computational perspective, it is often convenient to store and update the inverse (or the Cholesky factors) of this approximate Hessian since this is what is used in computing the quasi-Newton direction. What kind of computational savings can we obtain as a result? Describe (in a few words) how this formula is obtained.

$$(B^{k+1})^{-1} = (B^{\,k})^{-1} + \frac{\left((s^k)^t y^k + (y^k)^t (B^k)^{-1} y^k\right) s^k (s^k)^t}{((s^k)^t y^k)^2} - \frac{(B^k)^{-1} y^k (s^k)^t + s^k (y^k)^t (B^k)^{-1}}{(s^k)^t y^k}$$

- Implement a BFGS with a backtracking line search and compare its performance to that of Newton's direction on the Rosenbrock function of the previous homework.
- Use the last three iterates to estimate the rate of convergence of BFGS on this problem. Does the result make sense?