AMSC 660 Homework 10
Due soon
By Marvyn Bailly

Problem 1

(a) Let $\langle ., . \rangle$ be an inner product defined on a vector space V . Prove the Cauchy-Schwarz inequality.

$$\left|\langle u, v\rangle^2\right| \leq \langle u,u\rangle\langle v,v\rangle,\ \forall u,v \in V.$$

(b) Let B be a real symmetric positive definite $n \times n$ matrix. Use the Cauchy-Schwarz inequality to prove that

$$(g^\top Bg)(g^\top B^{-1}g) \geq (g^\top g)^2,\ \forall g \in \mathbb{R}^n.$$

Solution

Proof. (a) Let $u, v \in V$, where V is a vector space. Let $t \in \mathbb{C}$. Then, if we let

$$p(t) = \langle u + tv, u + tv\rangle \geq 0, \forall t$$

we get

$$\begin{aligned} p(t) &= \langle u,u\rangle + \overline{t\langle v,u\rangle} + t\langle v,u\rangle + t\bar{t}\langle v,v\rangle \\ &= \langle u,u\rangle + \overline{t\langle v,u\rangle} + t\langle v,u\rangle + |t|^2\langle v,v\rangle \\ &= \langle u,u\rangle + 2\mathrm{Re}(t\langle v,u\rangle) + |t|^2\langle v,v\rangle \\ &\leq \langle u,u\rangle + 2|t||\langle v,u\rangle| + |t|^2\langle v,v\rangle. \end{aligned}$$

Since all of the terms in $p(t)$ are nonnegative, we have that $p(t) \geq 0$. Thus the discriminant

$$(2|\langle u,v\rangle|)^2 + 4(\langle u,u\rangle\langle v,v\rangle) \geq 0 \implies \left|\langle u,v\rangle^2\right| \leq \langle u,u\rangle\langle v,v\rangle,$$

which proves the Cauchy-Schwarz inequality.

(b) Let B be a real SPD $n \times n$ matrix. We wish to show

$$(g^\top Bg)(g^\top B^{-1}g) \geq (g^\top g)^2,\ \forall g \in \mathbb{R}^n.$$

Let $u = B^{-1}g$ and $v = g$. Then using $\langle u,v\rangle_B = u^\top Bv$ we have that

$$\|u\|^2 = g^\top B^{-1}BB^{-1}g = g^\top B^{-1}g,$$

and

$$\|v\|^2 = g^\top Bg.$$

Then

$$(g^\top Bg)(g^\top B^{-1}g) = \|u\|^2\|v\|^2 \geq \langle u, v\rangle^2 = ((B^{-1}g)^\top Bg)^2 = (g^\top g)^2.$$

Thus we have that

$$(g^\top Bg)(g^\top B^{-1}g) \geq (g^\top g)^2,\ \forall g \in \mathbb{R}^n.$$

□

Problem 2

Consider Newton's algorithm for solving the trust-region subproblem ([NW], Algorithm 4.3, page 87). Prove that Eq. (4.43) is equivalent to Eq. (4.44) in [NW], i.e., that for

$$\phi(\lambda) = \frac{1}{\Delta} - \left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^2} \right)^{-1/2},$$

where (q_j, λ_j) are the eigenpairs of B, the Newton iteration

$$\lambda^{(l+1)} = \lambda^{(l)} - \frac{\phi(\lambda^{(l)})}{\phi'(\lambda^{(l)})},$$

is given by

$$\lambda^{(l+1)} = \lambda^{(l)} + \left(\frac{\|p_l\|}{\|z_l\|} \right)^2 \frac{\|(\|p_l) - \Delta}{\Delta},$$

where $z_l = L^{-1} p_l, p_l = -(B + \lambda^{(l)} I)^{-1} g$, and L is the Cholesky factor of $B + \lambda^{(l)} I$, i.e., $B + \lambda^{(l)} = LL^\top$. Note: $R = L^\top$ in Algorithm 4.3.

Hint: you will need to compute the derivative of ϕ and express it in terms of $\|p_l\|$ and $\left\|(B + \lambda^{(l)} I)^{-1} g\right\|^2$. Also, you will need to use the fact that the Cholesky factor of any SPD matrix M is related to $M^{1/2}$ via an orthogonal transformation.

Solution

Proof. Consider Newton's algorithm for solving the trust-region subproblem which seeks a solution p that satisfies

$$p(\lambda) = -(B + \lambda I)^{-1} g,$$

for λ sufficiently large that $B + \lambda I$ is positive definite where B is symmetric. As B is symmetric, we can use the spectral decomposition $B = Q \Lambda Q^\top$, where Q and Λ are made from the eigenpairs (q_j, λ_j) of B. Then we have that $B + \lambda I = Q(\Lambda + \lambda I) Q^T$ and thus we can rewrite $p(\lambda)$ as

$$p(\lambda) = -Q(\Lambda + \lambda I)^{-1} Q^\top g = -\sum_{j=1}^{n} \frac{q_j^\top g}{\lambda_j + \lambda} q_j,$$

and thus

$$\|p(\lambda)\|^2 = \sum_{j=1}^{n} \frac{\left(q_j^\top g\right)^2}{(\lambda_j + \lambda)^2}.$$

Now Newton's method uses the Newton iteration of the form

$$\lambda^{(l+1)} = \lambda^{(l)} - \frac{\phi(\lambda^{(l)})}{\phi'(\lambda^{(l)})},$$

where $\phi(\lambda^{(l)})$ is given by

$$\phi(\lambda^{(l)}) = \frac{1}{\Delta} - \left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^2}\right)^{-1/2} = \frac{1}{\Delta} - \|p(\lambda^{(l)})\|^{-1} = \frac{1}{\Delta} - \|p_j\|^{-1}. \tag{1}$$

Next, let's compute $\phi'(\lambda)$ to get

$$\begin{aligned}
\phi'(\lambda_j) &= \mathrm{d}x\left(\frac{1}{\Delta} - \left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^2}\right)^{-1/2}\right) \\
&= \frac{1}{2}\left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^2}\right)^{-3/2} (-2)\left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^3}\right) \\
&= -\|p_j\|^{-3}\left(\sum_{j=1}^{n} \frac{(q_j g)^2}{(\lambda_j - \lambda)^3}\right).
\end{aligned}$$

Now recall that $(B + \lambda I)$ is SPD by construction and since B is symmetric. Thus $(B + \lambda I)$ has a unique Cholesky factorization of the form $(B + \lambda I) = LL^\top$. Now we also have that

$$\begin{aligned}
(B + \lambda I) &= Q(\Lambda + \lambda I)Q^T \\
&= Q(\Lambda + \lambda I)^{1/2}(\Lambda + \lambda I)^{1/2}Q^\top \\
&= Q(\Lambda + \lambda I)^{1/2}\big(Q(\Lambda + \lambda I)^{1/2}\big)^\top \\
&= Q(\Lambda + \lambda I)^{1/2}\big(Q(\Lambda + \lambda I)^{1/2}\big)^\top \\
&= LL^\top,
\end{aligned}$$

so $L = Q(\Lambda + \lambda I)^{1/2} = (B + \lambda I)Q$. Then we have that

$$\begin{aligned}
z_l &= L^{-1}p_l \\
&= -Q(\Lambda + \lambda^{(l)} I)^{-1/2}Q^{-1}Q(\Lambda + \lambda^{(l)} I)^{-1}Q^\top g \\
&= -Q(\Lambda + \lambda^{(l)} I)^{-3/2}Q^\top g \\
&= -(B + \lambda^{(l)} I)^{-3/2} g
\end{aligned}$$

Since $z_l = -(B + \lambda^{(l)} I)^{-3/2} g$ is the squared norm of the solution of the second term in ϕ' we get

$$\phi'(\lambda_l) = -\frac{\|z_l\|^2}{\|p_l\|^3}.$$

Plugging the $\phi(\lambda)$ and $\phi'(\lambda)$ into the Newton iteration equation yields

$$\lambda^{(l+1)} = \lambda^{(l)} - \left(\frac{1}{\Delta} - \|p_j\|^{-1}\right)\frac{-\|p_j\|^3}{\|z_l\|^2} = \lambda^{(j)} + \frac{\|p_j\|^2}{\|z_l\|^2}\left(\frac{\|p_l\| - \Delta}{\Delta}\right).$$

□

Problem 3

Consider the problem of finding local energy minima of the LJ_7 as in Problem 3 of HW9. Consider the same set of initial conditions: four initial conditions close to its four local minima, and ten random initial conditions. Implement the BFGS trust-region method with the dogleg subproblem solver. Compare its performance with the trust-region Newton with the exact subproblem solver implemented in the provided code by creating a table with the number of iterations required to achieve convergence and plotting the graph of f and $\|\nabla f\|$ against the iteration number for each test case (the four initial conditions close to the minima and one representative random configuration initial condition). Do it for each of the four initial conditions approximating the four local minima and ten random initial conditions. The set of figures to include is the same as for Problem 3 in HW9. Comment on the performance of trust-region methods compared to the performance of line-search methods.

Solution

Proof. Let's first implement the BFGS method with the following MatLab code:

```
if direction == 1 && rho > eta % update bfgs matrix B if
   accepted
    p = xnew - x;
    y = gnew - g;
    if mod(iter,BFGSReset)==0
        B = eye(length(x));
    else
        B = B-((B*p)*(p'*B'))/(p'*B*p) +(y*y')/(y'*p);
    end
end
```

and the Dogleg subproblem solver with:

```
if direction == 1 % the Dogleg method
    pu = - (norm(g)^2*g)/(g'*B*g);
    pu_norm = norm(pu);
    if pu_norm >= Delta
        %fprintf("pu >= delta with tau:");
        tau = Delta/(pu_norm);
        p = tau * pu;
        flag_boundary = 1;
    else
        pb = -B\g;
        if pb <= Delta
            p = pb;
        else
```

```
            % solve the quadratic in alpha
            a = (pu - pb)'*(pu - pb);
            b = 2*(pu'*(pb - pu));
            c = pu'*pu - Delta^2;

            % get alpha
            %fprintf("pu < delta with alpha:");
            alpha = (-b + sqrt(b^2 - 4*a*c)) / (2*a);
            p = pu + alpha*(pb - pu);

            flag_boundary = 1;
        end
    end
end
```

where we solve the quadratic formula in α given by

$$\begin{aligned}\|p^u + \alpha(p^b - p^u)\|^2 &= (p^u + \alpha(p^b - p^u))^\top (p^u + \alpha(p^b - p^u)) \\ &= (p^u - p^b)^\top (p^u - p^b)\alpha^2 + 2((p^u)^\top (p^b - p^u))\alpha + (p^u)^\top p^u = \Delta^2,\end{aligned}$$

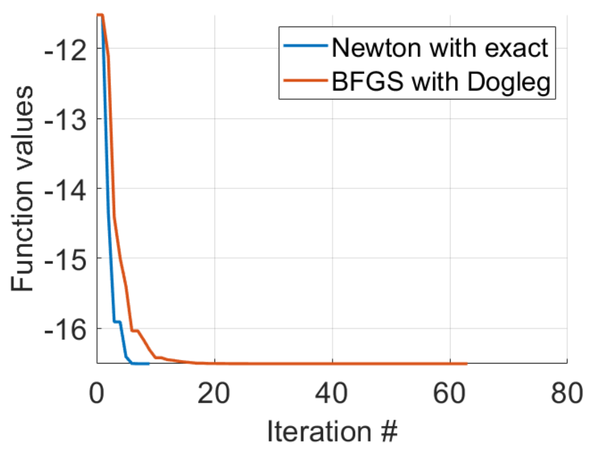
using the positive root of the quadratic. To compare the two methods, we will use a max iteration count of 400, a tolerance of 10^{-6}, and reset the BFGS matrix every 20 iterations. Now we will run both methods for the fourth initial conditions approximating the four local minima and plotting the function value and $\|\nabla f\|$ against the iteration numbers as seen in Figure 1, Figure 2, Figure 3, and Figure 4 for the four respective local minima. From the plots, we can see that the Newton method using the exact subproblem solver converges rapidly to the local minima while the BFGS method takes more iterations to converge to the same local minima. On the other hand, when starting at a random initial condition, we see that the Newton method outperforms the BFGS method in both iteration count and precision as seen in Figure 5. It appears that the BFGS method gets stuck with an approximate solution around 10^{-6} while the Newton method reaches 10^{-15}. Running both methods on ten different random initial conditions, as seen in Table 1, we see that for 8 of the ten trials, both methods converge to the same local minima while in the other 2 trials, the methods converge to different local minima. In all of the trials, the Newton method converges with significantly less iterations than the BFGS method.

My code used for this problem can be found at `https://github.com/MarvynBailly/AMSC660/tree/main/homework10`.

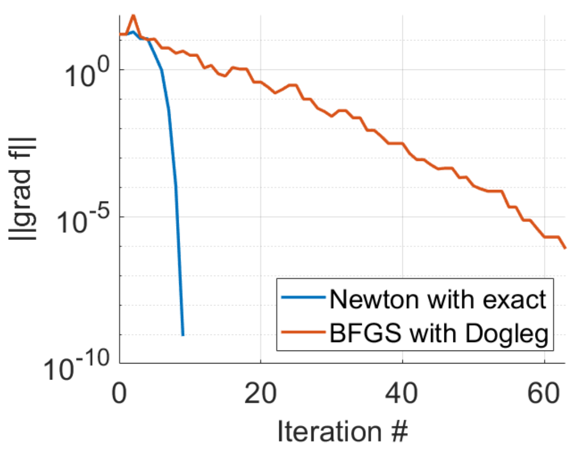
method	iter count	min found
Newton	32	-16.5054
BFGS	178	-16.5054
Newton	31	-16.5054
BFGS	152	-16.5054
Newton	31	-15.5331
BFGS	234	-15.5331
Newton	22	-15.5932
BFGS	151	-15.5331
Newton	32	-16.5054
BFGS	155	-16.5054

method	iter count	min found
Newton	28	-15.5331
BFGS	120	-15.5331
Newton	27	-15.5331
BFGS	396	-15.5331
Newton	147	-15.5331
BFGS	400	-15.5331
Newton	30	-15.5932
BFGS	239	-15.5331
Newton	39	-15.5331
BFGS	392	-15.5331

Table 1: Results of the five algorithms ran on random initial conditions.

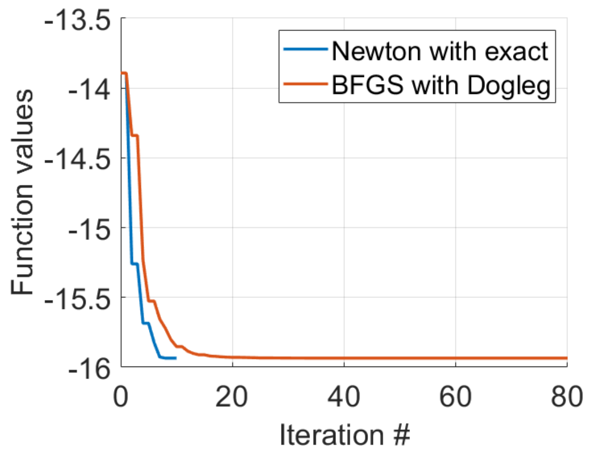


(a)

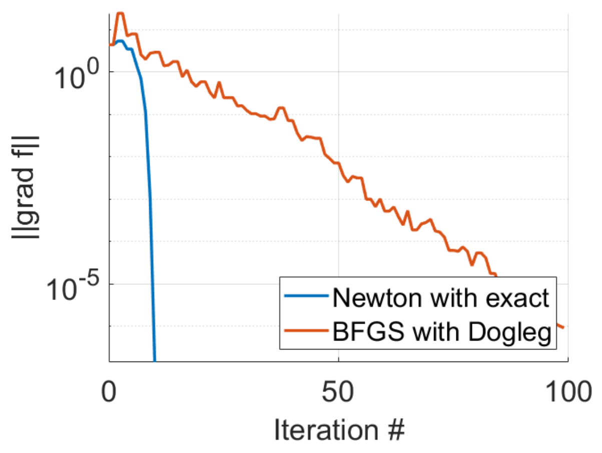


(b)

Figure 1: Function values of the Newton method with the exact solver and the BFGS method using the dogleg method at each iteration are shown in (a) while the norm of the gradient of the function value is shown in (b) when starting at the first initial condition.

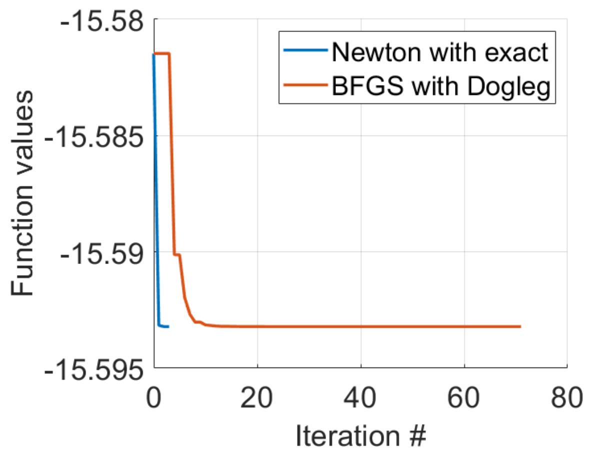


(a)

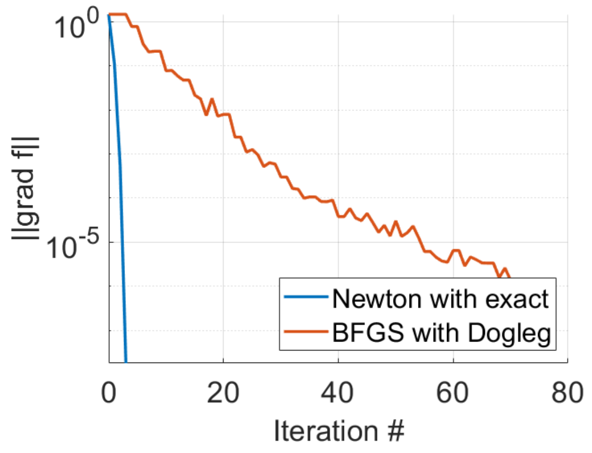


(b)

Figure 2: Function values of the Newton method with the exact solver and the BFGS method using the dogleg method at each iteration are shown in (a) while the norm of the gradient of the function value is shown in (b) when starting at the second initial condition.

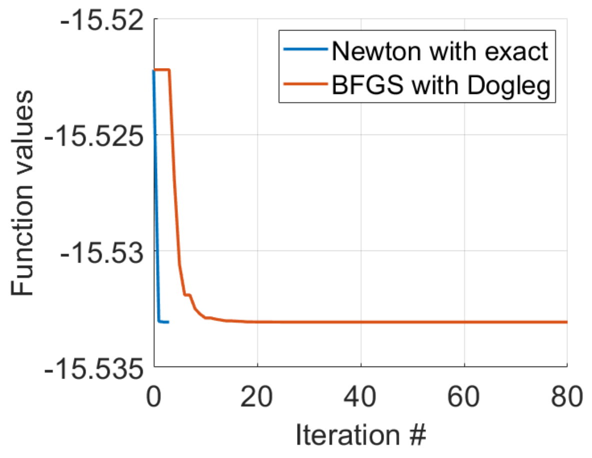


(a)

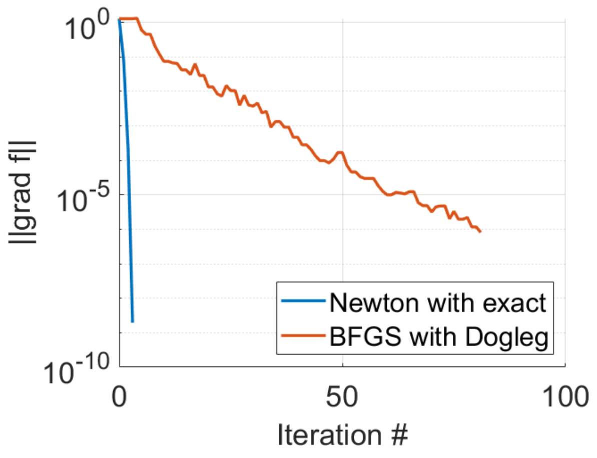


(b)

Figure 3: Function values of the Newton method with the exact solver and the BFGS method using the dogleg method at each iteration are shown in (a) while the norm of the gradient of the function value is shown in (b) when starting at the third initial condition.

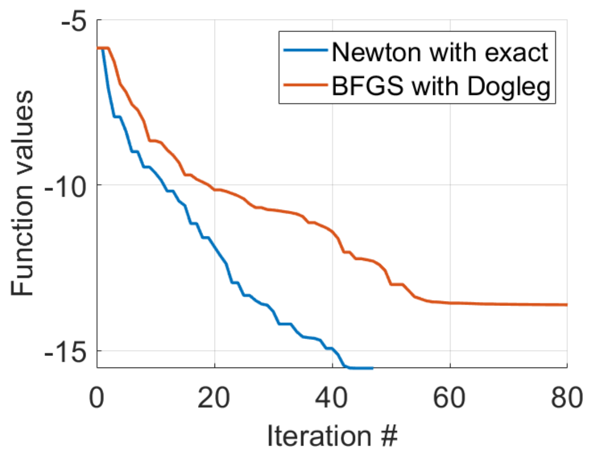


(a)

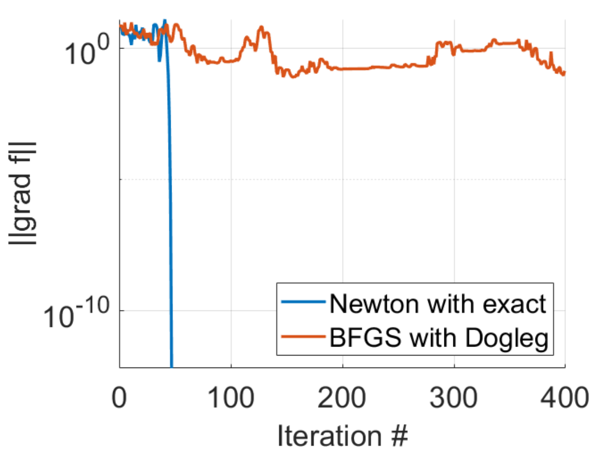


(b)

Figure 4: Function values of the Newton method with the exact solver and the BFGS method using the dogleg method at each iteration are shown in (a) while the norm of the gradient of the function value is shown in (b) when starting at the fourth initial condition.



(a)



(b)

Figure 5: Function values of the Newton method with the exact solver and the BFGS method using the dogleg method at each iteration are shown in (a) while the norm of the gradient of the function value is shown in (b) when starting at a random initial condition.

□

Problem 4

(Approx. Problem 3.1 from [NW]) Write a code that applies the two algorithms from the previous problem (the trust-region BFGS with the dogleg solver and the trust-region Newton with the exact subspace solver) to the Rosenbrock function as in Problem 4 of HW9:

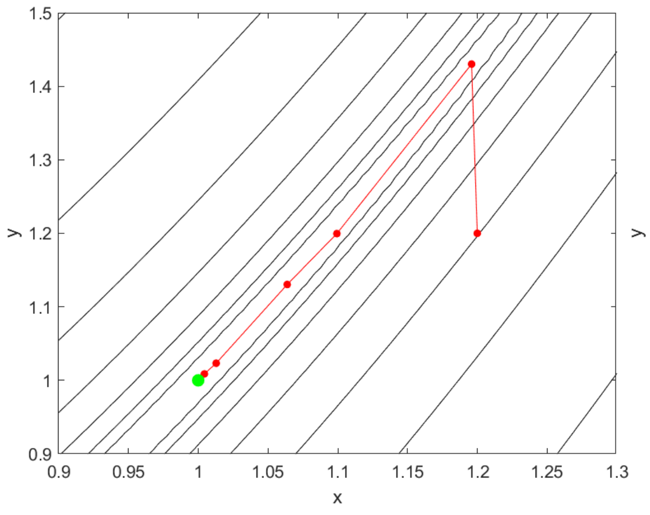
$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$

Experiment with the same two initial conditions: $(1.2, 1.2)$ and $(-1.2, 1)$. Plot the level sets of the Rosenbrock function using the command contour and plot the iterations for each method over it. Plot $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis for each method. Compare the performance of the methods.

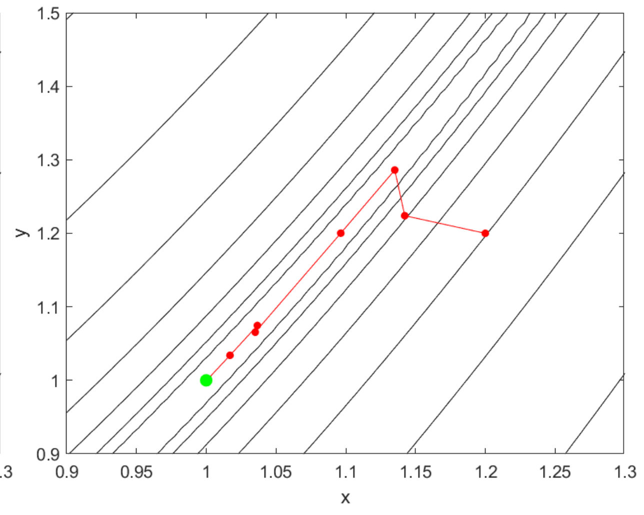
Solution

Proof. We begin by modifying the code we used above to find the minima of the Rosenbrock function. We begin the search from the initial conditions $(1.2, 1.2)$ and $(-1.2, 1)$ and plot the contour and the iterates upon it seen in Figures 6 and 7. Next, we plot the $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis as seen in Figures 8. We observe that Newton with the exact solver convergences in fewer iterations than the BFGS method using the dogleg solver. But both methods are able to identify the local minima.

My code used for this problem can be found at `https://github.com/MarvynBailly/AMSC660/tree/main/homework10`.

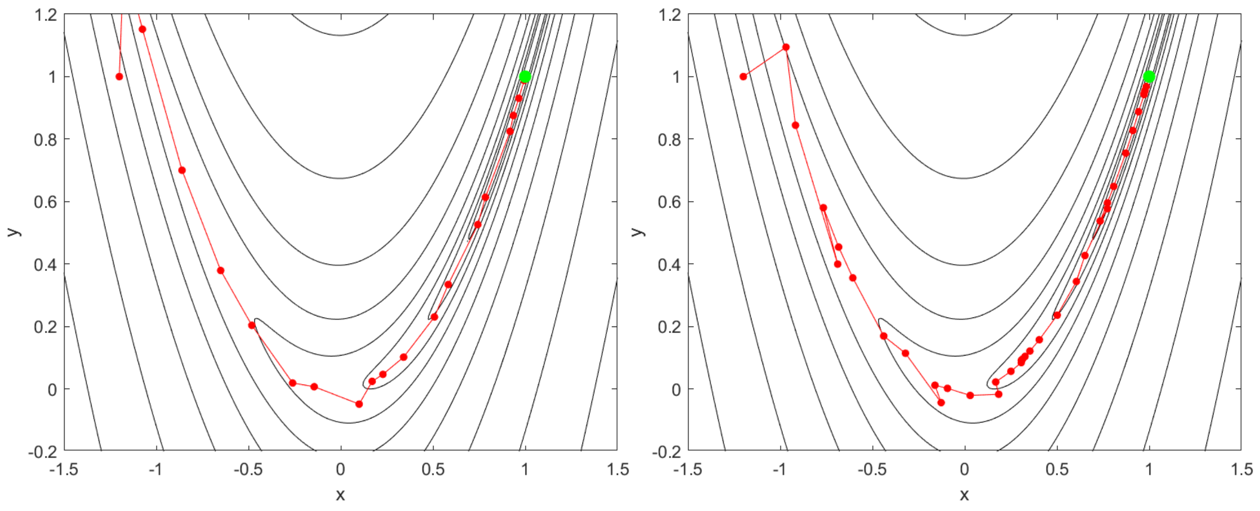


(a) Newton with exact subproblem solver.



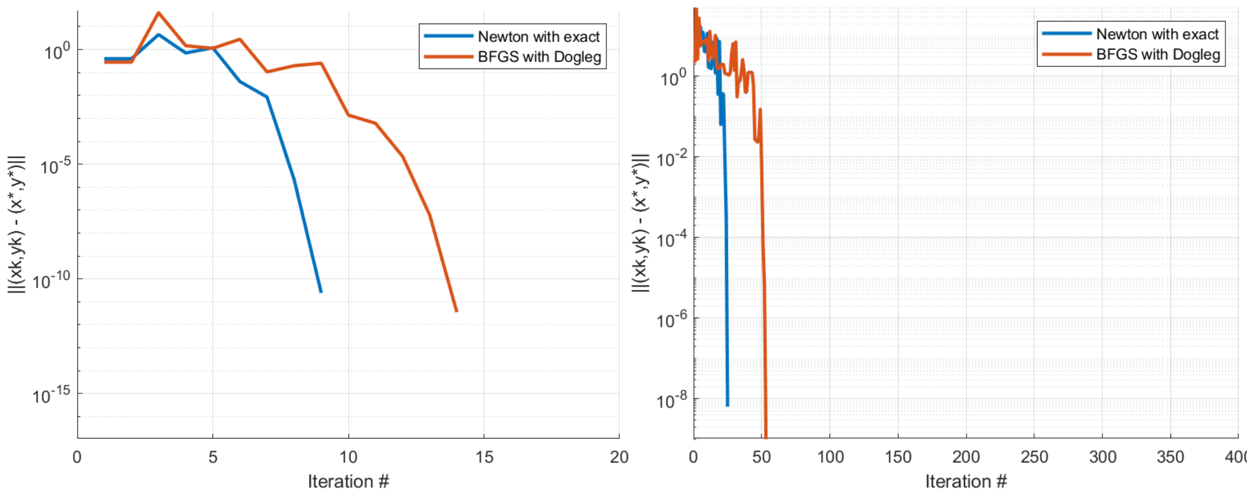
(b) BFGS method with Dogleg subproblem solver.

Figure 6: A contour plot of the Rosenbrock function with iterates of the Newton method using the exact subproblem solver (seen in (a)) compared to the BFGS method using the Dogleg subproblem solver (seen in (b)) from the initial condition $(1.2, 1.2)$



(a) Newton with exact subproblem solver. (b) BFGS method with Dogleg subproblem solver.

Figure 7: A contour plot of the Rosenbrock function with iterates of the Newton method using the exact subproblem solver (seen in (a)) compared to the BFGS method using the Dogleg subproblem solver (seen in (b)) from the initial condition $(-1.2, 1)$.



(a) From initial condition $(1.2, 1.2)$. (b) From initial condition $(-1.2, 1)$.

Figure 8: Plot of $\|(x_k, y_k) - (x^*, y^*)\|$ versus k in the logarithmic scale along the y-axis. The initial condition $(1.2, 1.2)$ is shown in (a) while the initial condition $(-1.2, 1)$ is shown in (b).

$\square$