3X03 Assignment 2

Neelkant Teeluckdharry

November 2024

1 Problem 3

1.1 Part A

Both the Jacobi and Gauss-Seidel are iterative methods in the form:

$$Mx^{k+1} = Nx^k + b$$

This gives

$$M(x^k - x^*) = N(x^{k-1} - x^*)$$

$$Me^k = Ne^{k-1} \implies e^k = M^{-1}Ne^{k-1}$$

$$\left\|e^k\right\| = \left\|M^{-1}Ne^{k-1}\right\| \leq \left\|M^{-1}N\right\| \left\|e^{k-1}\right\| = \|G\| \left\|e^{k-1}\right\|$$

By recursion, this yields

$$= \left\|G^k\right\| \left\|e^{(0)}\right\|$$

In the graph, when the Jacobi Method is applied for 50 iterations, the absolute error obtained is strictly bounded by the graph of $\|G\|^k \left\|e^0\right\|$, where k is the number of iterations. In contrast, when using Gauss-Seidel, the absolute error converges in fewer iterations, although the error is no longer bounded by the theoretical limit above approximately 33 iterations of the method. Both graphs seemingly converge around an absolute error with an order of magnitude of 10^{-15}, which is near machine precision for Float64. At this limit, $FP(x^k) - FP(x^{k-1}) = 0$. Thus, further iterations of the method will not improve the computed solution, causing both graphs to remain flat for the remaining iterations until k_{max}. As a result, it is expected that for more iterations of the Jacobi method, it would no longer obey the theoretical error bound.

1.2 Part B

As discussed above, the floating point representation of the solutions will converge to a value on the same order as machine precision for the Float64 type. This is because improvements to the solution above a certain precision will not be captured in Float64, as the difference between the exact and computed solutions are of order 10^{-16} causing $FP(x^{(k)}) - FP(x^{(k-1)}) = 0$. Consequently, the

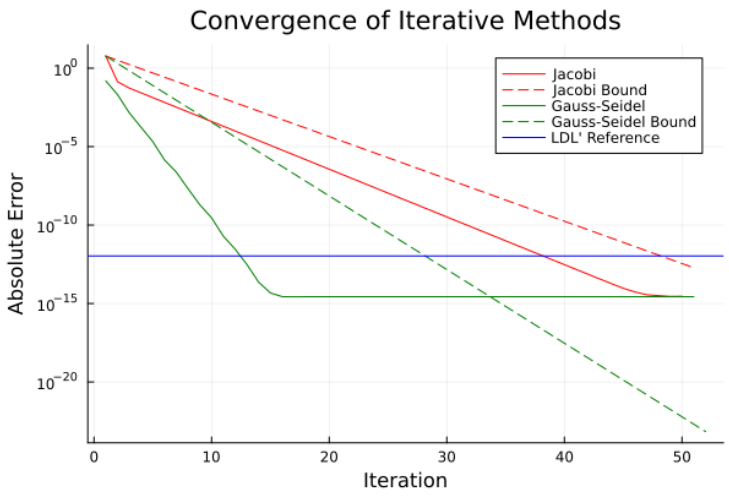

Figure 1: Convergence of Gauss-Seidel vs Jacobi

graph of absolute error will flatten for the remaining iterations of the respective solution methods. The expression for this value is given by:

$$x^{(k)} = M^{-1}Nx^{(k-1)} + M^{-1}b$$

$$= (M^{-1}N)^k x^{(0)} + (M^{-1}N)^{k-1}M^{-1}b + (M^{-1}N)^{k-2}M^{-1}b$$

$$... + (M^{-1}N)M^{-1}b + M^{-1}b$$

$$= (M^{-1}N)^k x^{(0)} + \sum_{i=0}^{k-1}(M^{-1}N)^i M^{-1}b$$

Assuming $\left\|M^{-1}N\right\| < 1$

$$= (M^{-1}N)^k x^{(0)} + (M^{-1}b)\frac{I - (M^{-1}N)^k}{I - (M^{-1}N)}$$

This implies as $k \to \infty$

$$= \boxed{\frac{M^{-1}b}{I - (M^{-1}N)}}$$

This expression can also be obtained using by using $FP(x^k) - FP(x^{k-1}) = 0$.

2 Problem 5

Let $v = [a_1, a_2...a_n]$, then $V = vv^T = [a_1v, a_2v, ..., a_nv]$. Thus, the columns are linearly dependent, and the rank of the matrix V is 1.
By the rank-nullity theorem, there are $n-1$ eigenvectors of the matrix V, each

having eigenvalue of 0.
We claim that the remaining eigenvalue is 1 with eigenvector v, and all other eigenvectors are orthogonal to v.
Consider

$$Vv = vv^T v = v \implies \lambda = 1$$

And some u orthogonal to v.

$$Vu = vv^T u = 0 \implies \lambda = 0$$

Thus, the only two eigenvalues are 0 and 1, with eigenvectors v and u.

3 Problem 7

Each function $f_i(x)$ can be expressed as

$$f_i(x) = \left\| \begin{bmatrix} x_1 - p_{i1} \\ x_2 - p_{i2} \\ \vdots \\ x_n - p_{in} \end{bmatrix} \right\| - d_i$$

$$= \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2 + \cdots + (x_n - p_{in})^2} - d_i$$

$$= (\sum_{k=1}^{n} (x_k - p_{ik})^2)^{1/2} - d_i$$

This implies

$$J(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j} = \frac{1}{2}(\sum_{k=1}^{n} (x_k - p_{ik})^2)^{-1/2} \cdot 2(x_j - p_{ij})$$

$$= (\sum_{k=1}^{n} (x_k - p_{ik})^2)^{-1/2} \cdot (x_j - p_{ij})$$

$$= \frac{x_j - p_{ij}}{\|x - p_{ij}\|}$$

Thus, the Jacobian Matrix is given by:

$$\implies J(x) = \begin{bmatrix} \frac{x_1 - p_{11}}{\|x - p_1\|} & \frac{x_2 - p_{12}}{\|x - p_1\|} & \cdots & \frac{x_n - p_{1n}}{\|x - p_1\|} \\ \frac{x_1 - p_{21}}{\|x - p_2\|} & \frac{x_2 - p_{22}}{\|x - p_2\|} & \cdots & \frac{x_n - p_{2n}}{\|x - p_2\|} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_1 - p_{n1}}{\|x - p_n\|} & \frac{x_2 - p_{n2}}{\|x - p_n\|} & \cdots & \frac{x_n - p_{nn}}{\|x - p_n\|} \end{bmatrix}.$$

4 Problem 9

4.1 Gradient

$$f(x) \triangleq \sum_{i=1}^{m} |f_i(x)|^2$$

$$= \sum_{i=1}^{m} (\|x - p_i\| - d_i)^2$$

$$= \sum_{i=1}^{m} \|x - p_i\|^2 - 2\sum_{i=1}^{m} d_i \|x - p_i\| + \sum_{i=1}^{m} d_i^2$$

$$= \sum_{i=1}^{m} (x - p_i)^T (x - p_i) - 2\sum_{i=1}^{m} d_i \|x - p_i\| + \sum_{i=1}^{m} d_i^2$$

Since $x^T p_i = p_i^T x$.

$$= \sum_{i=1}^{m} x^T x - 2x^T p_i + p_i^T p_i - 2\sum_{i=1}^{m} d_i \|x - p_i\| + \sum_{i=1}^{m} d_i^2$$

Applying the gradient operator

$$(\nabla f)_j = 2\sum_{i=1}^{m} (x_j - p_{ji}) - 2\sum_{i=1}^{m} d_i \frac{x_j - p_{ji}}{\|x - p_i\|}$$

$$\implies \nabla f = 2\sum_{i=1}^{m} (\|x - p_i\| - d_i) \frac{x - p_i}{\|x - p_i\|} = 2\sum_{i=1}^{m} f_i(x) \frac{x - p_i}{\|x - p_i\|}$$

4.2 Hessian

$$\nabla^2 f(x) = \frac{\partial}{\partial x} \nabla f(x) = \frac{\partial}{\partial x} (2\sum_{i=1}^{m} f_i(x) \frac{x - p_i}{\|x - p_i\|})$$

To derive this, the multivariate product rule is applied. It is important to know that

$$\frac{\partial}{\partial x} (\frac{x_k - p_i}{\|x - p_i\|}) = \frac{\|x - p_i\| \cdot I - (x - p_i) \cdot \frac{(x - p_i)}{\|x - p_i\|}}{\|x - p_i\|^2}$$

$$= \frac{I}{\|x - p_i\|} - \frac{(x - p_i)^T (x - p_i)}{\|x - p_i\|^3}$$

And,

$$\frac{\partial}{\partial x} f_i(x) = \frac{x - p_i}{\|x - p_i\|}$$

Therefore

$$\nabla^2 f(x) = 2\sum_{i=1}^{m}\left[\left(\frac{\partial f_i(x)}{\partial x}\right)\frac{x-p_i)^T}{\|x-p_i\|} + f_i(x)\frac{\partial}{\partial x}\left(\frac{x-p_i}{\|x-p_i\|}\right)\right]$$

$$\implies \nabla^2 f(x) = 2\sum_{i=1}^{m}\frac{(x-p_i)(x-p_i)^T}{\|x-p_i\|^2} + f_i(x)\frac{\|x-p_i\|\cdot I - (x-p_i)\frac{(x-p_i)^T}{\|x-p_i\|}}{\|x-p_i\|^2}$$

$$= 2\sum_{i=1}^{m}\frac{(x-p_i)(x-p_i)^T}{\|x-p_i\|^2} + f_i(x)\left(\frac{I}{\|x-p_i\|} - \frac{(x-p_i)(x-p_i)^T}{\|x-p_i\|^3}\right).$$

5 Problem 10

5.1 Plot

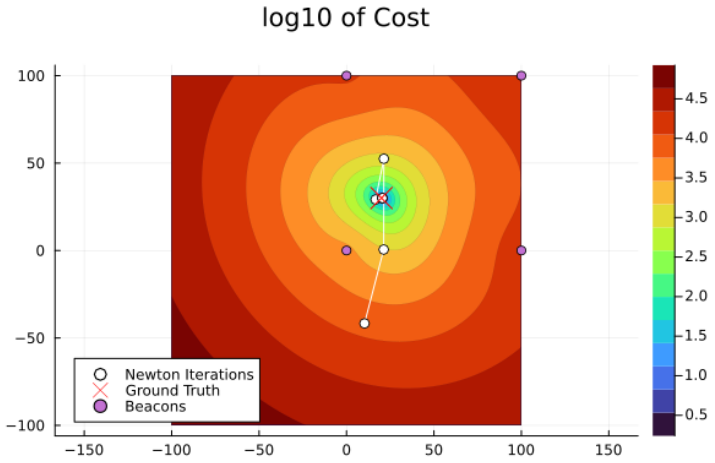

Figure 2: Decreasing cost as newton iterations converge towards critical point.

5.2 Part A

As shown in Figures 3 and 4, the value of $\nabla f(x^*) \approx 0$, which is in accordance with the derived first order optimality condition.

5.3 Part B

As show in Figures 3 and 4, the smallest eigenvalue $\lambda_{min} \approx 3.20699$, which is positive. This implies that all eigenvalues are also positive, which means the hessian $\nabla^2 f(x^*)$ is positive definite, as expected at the optimal cost.

```
x = copy(x0);

grad_f = ones(n);

while k <= max_iters && norm(grad_f) >= tol

        grad_f = compute_gradf(x,P,d);

        grad_gradf = compute_hessian(x,P,d);


        s = grad_gradf \ -grad_f;

        x += s;

        push!(x_trace, copy(x));


        k +=1;

end

println("grad_f(x*) is:", compute_gradf(x, P,d));
println("grad_gradf(x*) is:", compute_hessian(x,P,d));
println("Eigenvalues of grad_gradf(x*) are:", eigvals(compute_hessian(x,P,d)));

return x_trace
```

Figure 3: Modified script showing that $\nabla f(x^*)$ and $\nabla^2 f(x^*)$ are computed.

```
dhilan@dhilan-pc:~/julia_ws/Scientific-Computation/handout_a2$ julia plot_newton.jl
grad_f(x*) is:[-1.2628786905111156e-14, -2.2912227670701668e-14]
grad_gradf(x*) is:[3.7076561202200113 0.7582709760963003; 0.7582709760963003 4.355417291160314]
Eigenvalues of grad_gradf(x*) are:[3.206992281881158, 4.8560811294991675]
```

Figure 4: Results of computing $\nabla f(x^*)$ and $\nabla^2 f(x^*)$.