

MCSC 6020G - Numerical Analysis

Assignment 3

Parikshit Bajpai

Question 1

0.1 (a) Cholesky Decomposition

To find. The lower triangular matrix L_N such that $L_N L_N^T = T_N$, where $T_N \in \mathbb{R}^{N \times N}$ is a tridiagonal SPD matrix resulting from finite difference discretisation of one-dimensional Poisson equation with homogeneous Dirichlet boundary conditions:

$$T_N = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \quad (1)$$

Proof. Finding matrix L_N is equivalent to performing a Cholesky decomposition on SPD matrix T_N . The elements of matrix L_N in terms of elements of matrix T_N are, in general, given as [?]:

$$l_{11} = \sqrt{t_{11}} \quad (2)$$

$$l_{ij} = \frac{\left(t_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}\right)}{l_{jj}} \quad i = 2, \dots, N; \quad j = 1, \dots, i-1 \quad (3)$$

$$l_{ii} = \left(t_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{1/2} \quad (4)$$

This results in the following banded matrix:

$$\{l_{ij}\}_{\forall ij} = \begin{cases} \sqrt{1 + \frac{1}{i}} & i = j \\ \left(\frac{1}{i} - 1\right) \sqrt{1 + \frac{1}{i-1}} & j = i-1 \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

■

0.2 (b) Eigenvectors and eigenvalues

To find. All eigenvectors and eigenvalues of matrix T_N

Proof. Let v be the eigenvector of T_N . Then,

$$T_N v = \lambda v \quad (6)$$

where, λ denotes the vector of eigenvalues of T_N .

The eigenvector v is defined as:

$$v = [\sin(\alpha) \quad \sin(2\alpha) \quad \cdots \quad \sin(N\alpha)]^T \quad (7)$$

The k^{th} $T_N v$ is:

$$(T_N v)_k = \sum_{l=1}^N t_{k,l} \sin(l\alpha) \quad (8)$$

which takes the following form for the matrix T_N given above:

$$(T_N v)_1 = 2 \sin(\alpha) - \sin(2\alpha) \quad (9)$$

$$(T_N v)_k = -\sin((k-1)\alpha) + 2 \sin(\alpha) - \sin((k+1)\alpha) \quad k = 2, \dots, N-1 \quad (10)$$

$$(T_N v)_N = 2 \sin(N\alpha) - \sin((N-1)\alpha) \quad (11)$$

$$(12)$$

Using eqn. (9) and trigonometric identity $\sin(A+B) = 2 \sin(A) \cos(B)$, eqn. (6) can be rewritten as:

$$(T_N v)_k = (2 - 2 \cos(k\alpha)) v \quad (13)$$

so that the k^{th} eigenvalue is equal to $(2 - 2 \cos(k\alpha))$.

Therefore, the eigenvector, v , and the eigenvalues of matrix T_N can be written as:

$$v = [\sin(\alpha) \quad \sin(2\alpha) \quad \cdots \quad \sin(N\alpha)]^T \quad (14)$$

$$\lambda = [(2 - 2 \cos(\alpha)) \quad (2 - 2 \cos(2\alpha)) \quad \cdots \quad (2 - 2 \cos(N\alpha))]^T \quad (15)$$

■

Question 2

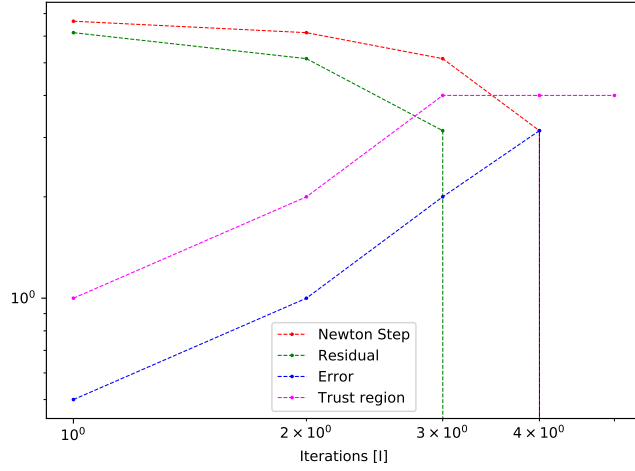
This exercise is aimed at employing the Newton-Hook method to find the root of the following function:

$$f_i(x) = x_i - e^{\lambda \cos(i \sum_{k=1}^N x_k)} \quad (16)$$

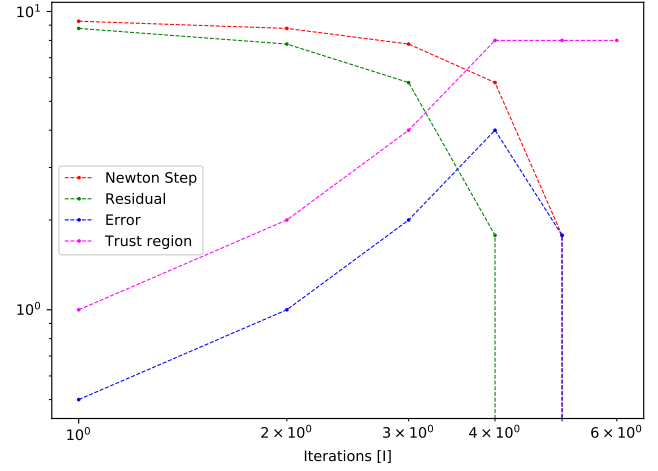
To find the optimum direction vector when the Newton step results in the solution approximation being outside the trust region, method of Lagrange multipliers is used and the resulting equation for the Lagrange multiplier is solved using a Newton-like method which uses a rational approximation to find the Lagrange multiplier. The order of convergence of Newton method is quadratic once the approximation is close enough to the true solution. However, a large amount of time might be spent in arriving close enough to the desired solution. Newton-Hook method is a trust-region based modification of Newton method to reduce the computational effort for arriving close enough to the true solution by rejecting Newton steps which would have lead to an increase in the residual and selecting an alternate residual which minimises the residual at subsequent iteration.

Eqn. (16) has a trivial solution $x = 1$ for $\lambda = 0$. This solution has been used to test the implementation of Newton-Hook algorithm and with the initial estimate as a uniform random vector with values between 1 and 1.5, Newton-Hook method converges rapidly to the trivial solution. The trust-region was halved if the residual within the trust region increased and every time a optimal direction vector was selected, the trust region was doubled. The results shown in fig. 1 show that the optimisation process results in a rapid decrease of the residuals and convergence is achieved quadratically once the approximation gets close to the true solution.

The rate of convergence was tested for a range of number of equations with the same parameter values as for the trivial case. and the results are shown in fig. ???. As the number of equations increases, the 2 norm of the residuals increases (since it's proportional of \sqrt{N}), which explains the plateau as the number of equations increases. The plateau denotes the iterations which a used for arriving at a close enough approximation and beyond that the convergence is quadratic.

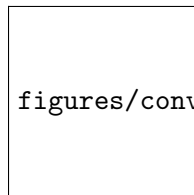


(a) 500 equations



(b) 1000 equations

Figure 1: Newton step size, residual, error, and trust region size vs. number of iterations for $\lambda = 0$.



figures/convergence.pdf

Figure 2: Convergence rate for the Newton-Hook method.