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:*

$$T_N = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \tag{1}$$

Proof. Finding matrix L_N is equivalent to performing a Cholesky decomposition on SPD matix T_N. The elements of matrix L_N in terms of elements of matrix T_N are, in general, given as [1]:

$$l_{11} = \sqrt{t_{11}} \tag{2}$$

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

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

This results in a banded matrix with the following elements:

$$l_{ij} = \begin{cases} \sqrt{1 + \dfrac{1}{i}} & i = j \\ \left(\dfrac{1}{i} - 1\right)\sqrt{1 + \dfrac{1}{i-1}} & j = i-1 \\ 0 & \textit{Otherwise} \end{cases} \tag{5}$$

■

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,

$$T_N v = \lambda v \tag{6}$$

where, λ denotes the vector of eigenvalues of T_N.

The eigenvector v is defined as:

$$v = \begin{bmatrix} \sin(\alpha) & \sin(2\alpha) & \cdots & \sin(N\alpha) \end{bmatrix}^T \tag{7}$$

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

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

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

$$(T_N v)_1 = 2\sin(\alpha) - \sin(2\alpha) \tag{9}$$
$$(T_N v)_k = -\sin((k-1)\alpha) + 2\sin(\alpha) - \sin((k+1)\alpha) \qquad k = 2, \ldots, N-1 \tag{10}$$
$$(T_N v)_N = 2\sin(N\alpha) - \sin((N-1)\alpha) \tag{11}$$
$$\tag{12}$$

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

$$(T_N v)_k = (2 - 2cos(k\alpha))\, v \tag{13}$$

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:

$$v = \begin{bmatrix} \sin(\alpha) & \sin(2\alpha) & \cdots & \sin(N\alpha) \end{bmatrix}^T \tag{14}$$
$$\lambda = \begin{bmatrix} (2 - 2\cos(\alpha)) & (2 - 2\cos(2\alpha)) & \cdots & (2 - 2\cos(N\alpha)) \end{bmatrix}^T \tag{15}$$

■

Question 2

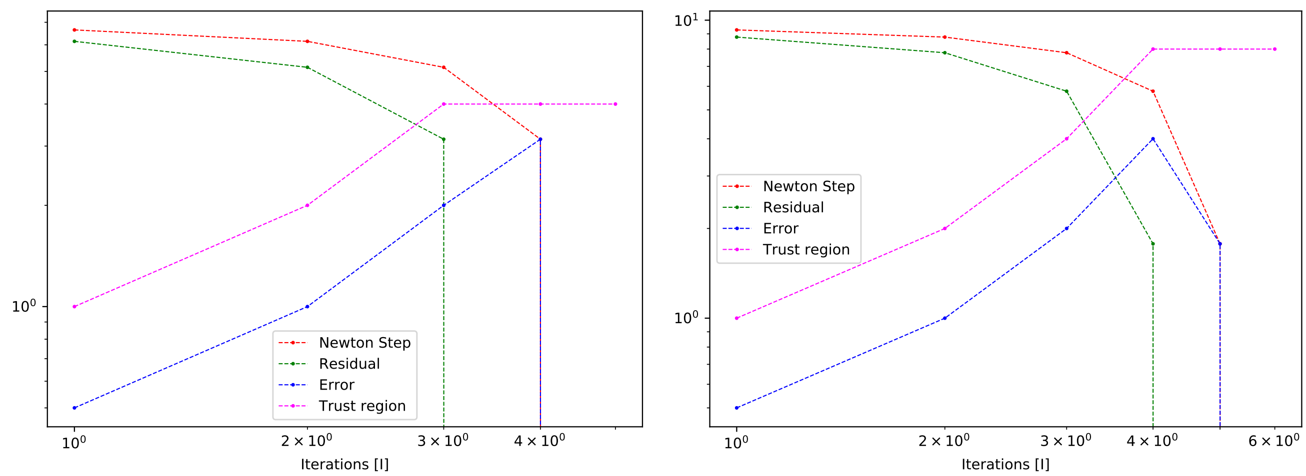
This exercise is aimed at employing the Newton-Hook method to find the root of the following function:

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

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. 2. As the number of equations increases, the 2



(a) 500 equations

(b) 1000 equations

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

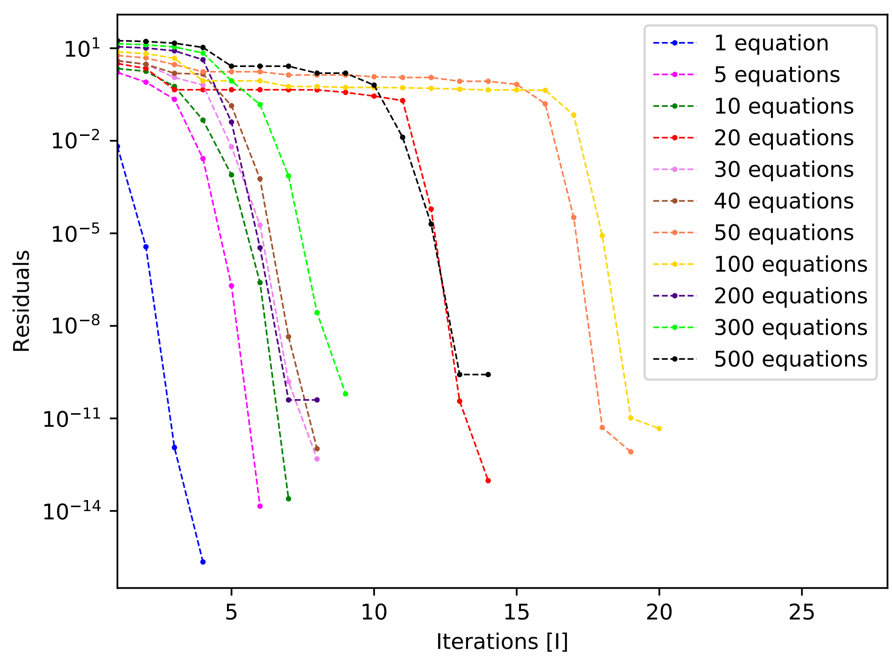


Figure 2: Convergence rate for the Newton-Hook method.

Table 1: Parameters used in Newton-Hook and Newton-like methods.

$\boldsymbol{\lambda}$	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Initial trust-region**	**Initial estimate of solution**	**Number of equations**
1	0.5	2.0	0.5	Uniformly distributed random array	25

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 beyong that the convergence is quadratic.

For the other convergence tests, the parameters shown in tab. 1 were used:

The parameter λ affects the order of convergence of the Newton-Hook method. The number of iterations required becomes lower as λ approaches 0. Moreover, the trust region needs to be smaller as λ becomes larger and larger. This can lead to convergence issues as the number of equations becomes large. In the tests, Newton-Hook converged consistently for number of equations up to 25 but for larger systems, Newton-Hook failed to converge as the value of parameter λ was changed from zero to one. The results for different values of lambda are shown in fig. 4

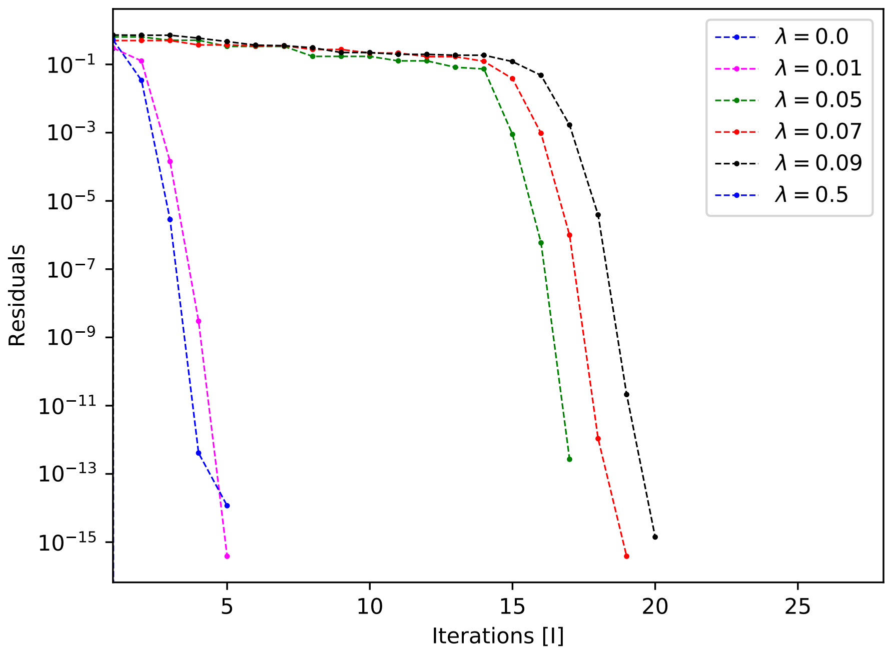


Figure 3: Convergence rate for different values of parameter λ.

The initial trust region also plays an important role in the number of iterations required for convergence. A very small initial estimate slows the progress of the code since larger Newton steps might still lead to a lower residual while a very large initial trust region can lead to increase in residuals for full Newton steps. Four different initial sizes were tested and the results have been shown in fig. **??**. As expected, for a very large trust region, the number of iterations spent in getting close to the true solution is large. The effect of an extremely small trust region was not observed as the smallest trust region was sufficiently large.

Finally, the initial guess plays a very important role in achieving convergence. The initial estimates were selected to be either equal to 1 which is the solution for trivial case or were randomnly selected from a uniform random distribution in [1,1.5]. Since the solution goes away from 1 as parameter λ goes away from 0, selecting 1 as an initial estimate was only useful for the cases with λ close to 0. In other cases, the uniform random distribution provided faster convergence. However, the uniform distribution is not guaranteed to provide an estimate which can ensure convergence and therefore, during testing, the method failed to converged a number of times when the random sampling was used. However, the probability of convergence with random was still more than that with 1 as the initial guess. The results are shown in fig. 5

In summary, selecting the optimal value of a number of parameters used in the implementation can significantly affect the rate of convergence of Newton-Hook method. The initial estimate of the solution is the most significant parameter and appropriate selection can ensure convergence. The initial trust region

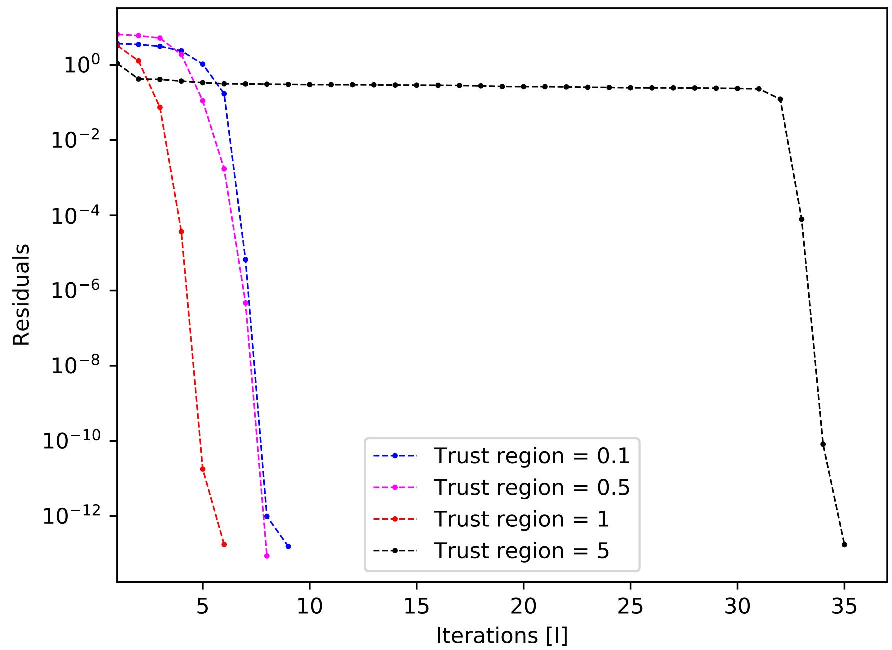
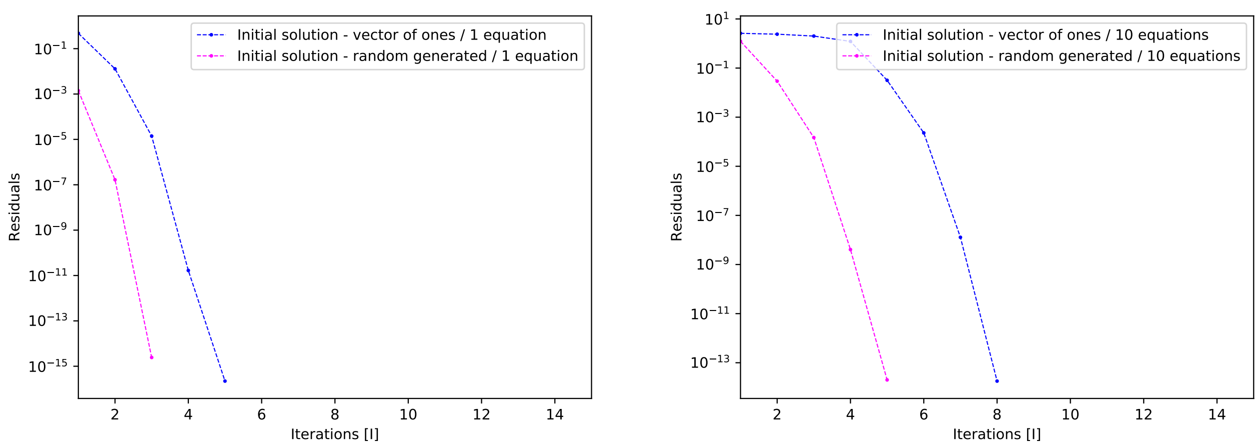


Figure 4: Convergence rate for different sizes of initial trust region.



(a) 1 equation

(b) 10 equations

Figure 5: Convergence rate for different initial estimates.

size and scaling parameters for the trust region can also affect the performance of the method and must be carefully studied.

Collaboration: This assignment was completed in collaboration with Marcos Machado. Discussions with other classmates were also quite useful in handling some of the issues.

References

[1] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics.* Texts in Applied Mathematics, Springer New York, 2017.