Homework nr. 4

Let n be the system's size, ϵ - the computation error, $A \in \mathbb{R}^{n\times n}$ - a real squared matrix, k_{max} - the maximum number of computing steps.

1. Using all the iterative methods described below, approximate the inverse of matrix A. Use formula (5), (6) for choosing the initial matrix V_0.

2. For each method, at the end of the algorithm display the number of computed steps. If the method converged, display the norm:

$$\||A * A^{-1}_{\text{approx}} - I_n\|_1.$$

3. Consider the matrix:

$$A = \begin{pmatrix} 1 & 2 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 2 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

   Deduce inductively the general form of the inverse of matrix A, by successively running your program using different values for n,the size of the matrix A. Denote by $A_{\text{exact}^{-1}}$ the exact inverse (computed with the deduced formula) and $A_{\text{approx}^{-1}}$ the inverse approximated using the iterative methods. Display the norm:

$$\|A^{-1}_{\text{approx}} - A^{-1}_{\text{exact}}\|.$$

Bonus 15 pt.: Adapt one of the iterative algorithms for approximating the inverse of a matrix, for a non-square matrix, $A \in \mathbb{R}^{m\times n}$, $m \neq n$.

Iterative Methods for Approximating the Inverse of a Matrix

The inverse of a nonsingular matrix A is approximated by computing a sequence of matrices $\{V_k\,;\,k \geq 0\}$ that converges to A^{-1}..

The Schultz method computes the sequence $\{V_k\,;\,k \geq 0\}$ in the following way:

$$V_{k+1} = V_k\big(2I_n - AV_k\big)\ ,\ k = 0, 1, 2, \ldots \quad , \quad V_0 - \text{ given} \tag{1}$$

This method is also known as the Hotelling-Bodewig algorithm ot the hyper-power iterative method.

Li and Li proposed the following two iterative methods for approximating the inverse of a matrix:

$$V_{k+1} = V_k\Big[3I_n - AV_k\big(3I_n - AV_k\big)\Big]\ ,\ k = 0, 1, 2, \ldots \tag{2}$$

$$V_{k+1} = \Big[I_n + \frac{1}{4}\big(I_n - V_kA\big)\big(3I_n - V_kA\big)^2\Big]V_k\ ,\ k = 0, 1, 2, \ldots \tag{3}$$

To ensure the convergence of the sequence $\{V_k\,;\,k \geq 0\}$, the first matrix of the sequence, V_0, must be chosen such that the following relation holds:

$$||AV_0 - I_n|| < 1. \tag{4}$$

Ways of choosing V_0 that ensure the sequence's convergence

1.

$$V_0 = \frac{A^T}{\|A\|_1\|A\|_\infty} \tag{5}$$

$$\begin{aligned} \|A\|_1 &= \max\left\{\sum_{i=1}^{n}|a_{ij}|\,;\,j = 1, 2, \ldots, n\right\}, \\ \|A\|_\infty &= \max\left\{\sum_{j=1}^{n}|a_{ij}|\,;\,i = 1, 2, \ldots, n\right\}. \end{aligned} \tag{6}$$

2. If matrix A is row diagonally dominant or column diagonally dominant:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad , \quad \forall i = 1,\dots,n \; - \; \text{row diagonally dominant}$$

$$|a_{ii}| > \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}| \quad , \quad \forall i = 1,\dots,n \; - \; \text{column diagonally dominant}$$

the formula that ensure the sequence's convergence is:

$$V_0 = \mathbf{diag}\Big(\frac{1}{a_{11}}, \frac{1}{a_{22}}, \dots, \frac{1}{a_{nn}}\Big) \tag{7}$$

3. If matric A is symmetric an positive definite, a good choice of V_0 is:

$$V_0 = \frac{1}{\|A\|_F} I_n \quad , \quad \|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}| \right)^{\frac{1}{2}} \tag{8}$$

4. $$V_0 = \alpha\, A^T \quad , \quad 0 < \alpha < \frac{2}{\|A\|_2^2} \quad , \quad \|A\|_2 = \sqrt{\rho(A^T A)} \tag{9}$$

5. $$V_0 = \alpha\, I_n \quad , \quad \alpha \in \mathbb{R} \quad , \quad \max\Big\{|1 - \lambda_i \alpha|; \lambda_i \text{ eigenvalue for } A\Big\} < 1 \tag{10}$$

The algorithm stops if:

1) $\|V_k - V_{k-1}\| < \epsilon$ (or $\|I_n - V_k A|\| < \epsilon$) - in this case matrix V_k approximates the inverse, $V_k \approx A^{-1}$; (one can use any norm described in (6))

2) $k > k_{max}$ - the maximum number of iterations has been exceeded, the approximation of the inverse with the desired precision failed ;

3) $\|V_k - V_{k-1}\| > 10^{10}$- the computed sequence does not converge, the algorithm failed to approximate the inverse of matrix A.

Not all the elements of the sequence $\{V_k\}$ must be stored in order to obtain an approximation for A^{-1}, it suffice to use only two matrices, one for storing V_k and the other for V_{k+1}.

For economically computing a matrix of type $C = aI_n - AV$ (where a is a real value, $a \in \mathbb{R}$), one computes only once in the program the matrix $B = (-A)$, then the matrix $C = B * V$ is calculated. The required $aI - AV$ matrix is obtained from C by adding the constant a to all the diagonal elements of matrix C ($c_{ii} = c_{ii} + a,\ \forall i$).

A possible approximation scheme for computing an approximation for the inverse of a matrix is the following:

$V0 = V1 = \dfrac{A^T}{\|A\|_1 \|A\|_\infty}$;
$k = 0$;
$k_{max} = 10000$; // a possible value
do
{
$V0 = V1$; // deep copy
compute $V1$ using $V0$ and one of the formulae (1), (2), (3);
compute $\Delta V = \|V1 - V0\|$;
$k = k + 1$;
}
while ($\Delta V \geq \epsilon$ and $k \leq k_{max}$ and $\Delta V \leq 10^{10}$)
if ($\Delta V < \epsilon$) $V1 \approx A^{-1}$;
else *divergence* ; // the algorithm fails to approximate the inverse