Basic exercises in Numerical Linear Algebra

Fred Vermolen

Computational Mathematics Group
Department of Mathematics and Statistics
University of Hasselt, Belgium

In this lab session, we will be acquainted with some elementary solvers for the solution of a linear system of equations. We will see the defect correction scheme, Jacobi, Gauss-Seidel, gradient descent method and the (precondioned) conjugate gradient method.

The defect correction algorithm to solve $A\mathbf{x} = \mathbf{b}$ is given by

$k = 0$;
Choose $\mathbf{x}_0$;
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$;
while $||\mathbf{r}_k|| > \varepsilon$ **do**
  $\mathbf{p}_k = P^{-1}\mathbf{r}_k$;
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$;
  $\mathbf{r}_{k+1} = \mathbf{r}_k - A\mathbf{p}_k$;
  $k = k + 1$;
end

Algorithm 1: The defect correction method to solve $A\mathbf{x} = \mathbf{b}$.

Exercise 1 *We consider some general aspects of the defect correction scheme:*

a Show that if $P = A$ then immediate convergence is obtained.

b Show that the expression for $\mathbf{r}_{k+1}$ is equivalent to

$$\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1}.$$

△

Exercise 2 *Let S be an $n \times n$ matrix with $s_{j,j+1} = 1$, $j = 1, \ldots, n-1$, $s_{jk} = 0$ otherwise and I be the identity matrix. We solve $A\mathbf{x} = \mathbf{f}$, with $A = 2I - S - S^T$ and $f_i = 1$. Use defect correction with $P^{-1} = \frac{1}{2}I$. For the construction of the A-and P(preconditioner)-matrices, you may use the following matlab code:*

```
e = ones(n,1);
A = spdiags([-e 2*e -e],-1:1,n,n);
P = 2*speye(n);
f = ones(n,1);
```

Program the defect correction method in Matlab. As a starting vector, you may use a randomised vector or just the zero vector. Plot the logarithm of $||\mathbf{r}_k||_2$ as a function of iteration number k (you can use the command semilog *in Matlab). Compare the number of iterations for $n = 10$, $n = 100$ and $n = 1000$ to arrive at a residual with $||\mathbf{r}_k||_2 < 10^{-5}$.* △

Exercise 3 *The oldest and probably simplest iterative method to solve $A\mathbf{x} = \mathbf{f}$ is Jacobi's method. Let A be an $n \times n$ 2D Laplace matrix, you may use the following matlab code for the construction of the A–matrix and $f_k = 1$:*

```
e = ones(nx,1);
B = spdiags([-e 2*e -e],-1:1,nx,nx);
A = kron(B,speye(nx)) + kron(speye(nx),B);
b = ones(nx,1);
f = kron(b,b); % you may also use f = ones(nx^2,1) ;
```

We write the matrix A as $A = D - L - U$ where $D = diag(a_{11}, \ldots, a_{nn})$ and $u_{ij} = a_{ij}$ if $j > i$, else $u_{ij} = 0$, $l_{ij} = a_{ij}$ if $i > j$, else $l_{ij} = 0$. We take $P = D$, you may use

*P = 4*speye(nx^2);*

Do the same as in Exercise 2 with $nx = 10$ and $nx = 50$ and $nx = 100$ ($n = nx^2$). △

Exercise 4 *Show that Jacobi's method can be written as*

$$D\mathbf{x}_{k+1} = (L + U)\mathbf{x}_k + \mathbf{f}.$$

△

Exercise 5 *Another classical iterative method to solve $A\mathbf{x} = \mathbf{f}$ is Gauss–Seidel's method. We write the matrix A as $A = D - L - U$ where $D = diag(a_{11}, \ldots, a_{nn})$ and $u_{ij} = a_{ij}$ if $j > i$, else $u_{ij} = 0$, $l_{ij} = a_{ij}$ if $i > j$, else $l_{ij} = 0$. Now we take $P = D - L$, use*

*BP = spdiags([-e 2*e],-1:0,n,n);*
P = kron(BP,speye(n)) + kron(speye(n),BP);

Do the same as in Exercise 3. △

Exercise 6 *Show that Gauss-Seidel's method can be written as*

$$(D - L)\mathbf{x}_{k+1} = U\mathbf{x}_k + \mathbf{f}.$$

△

The gradient descent method reads as

$k = 0$;
Choose $\mathbf{x}_0$;
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$;
$\mathbf{p}_0 = \mathbf{r}_0$;
while $||\mathbf{r}_k|| > \varepsilon$ **do**
  $\alpha_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{p}_k, A\mathbf{p}_k)}$;
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$;   *% update solution* ;
  $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$;   *% update residual*;
  $\mathbf{p}_{k+1} = \mathbf{r}_{k+1}$;   *% find new orthogonal search direction*;
  $k = k + 1$;
end

Algorithm 2: The gradient descent method to solve $A\mathbf{x} = \mathbf{b}$.

Exercise 7 *Next we use the gradient descent method. Program the gradient descent method in Matlab. Repeat the steps from Exercise 3 with the same matrix and right-hand side vector as in Exercise 3. Show the logarithm of the 2-norm of the residual $||\mathbf{r}_k||_2$ as a function of the iteration number k. Further, show the logarithm of the A-norm of the error $||\varepsilon_k||_A = ||\mathbf{x}_k - \mathbf{x}||_A$, where $\mathbf{x}$ is the exact solution, that can be obtained by $x = A\backslash f$ in Matlab.* △

Next, we consider the conjugate gradient method:

$k = 0$;
Choose $\mathbf{x}_0$;
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$;
$\mathbf{p}_0 = \mathbf{r}_0$;
while $||\mathbf{r}_k|| > \varepsilon$ **do**

$\alpha_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{p}_k, A\mathbf{p}_k)}$;
$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$; *% update solution*;
$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$; *% update residual*;
$\beta_k = \frac{(\mathbf{r}_{k+1}, \mathbf{r}_{k+1})}{(\mathbf{r}_k, \mathbf{r}_k)}$;
$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$; *% find new A–orthogonal search direction*;
$k = k + 1$;

end

Algorithm 3: The conjugate gradient method to solve $A\mathbf{x} = \mathbf{b}$.

Exercise 8 *Next we use the conjugate gradient method. Program the conjugatevgradient method in Matlab. Repeat the steps from Exercise 3 with the same matrix and right-hand side vector as in Exercise 3. Show the logarithm of the 2-norm of the residual $||\mathbf{r}_k||_2$ as a function of the iteration number k. Further, show the logarithm of the A-norm of the error $||\varepsilon_k||_A = ||\mathbf{x}_k - \mathbf{x}||_A$, where $\mathbf{x}$ is the exact solution, that can be obtained by $x = A\backslash f$ in Matlab.*
Compare your results to the convergence bound from Luenberger, given by

$$||\mathbf{x}_k - \mathbf{x}||_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k ||\mathbf{x}_0 - \mathbf{x}||_A, \text{ where } \kappa = \frac{\lambda_1}{\lambda_n},$$

with λ_1 and λ_n, respectively, the largest and smallest eigenvalue of matrix A (which is symmetric, positive definite). △

Preconditioning of the system $A\mathbf{x} = \mathbf{f}$ leads to a more favourable condition of the matrix. Preconditioning can be done in several ways (see lecture notes). We take the simplest preconditioner, P, namely $P = diag(a_{11}, \ldots, a_{nn})$ (the so-called diagonal preconditioner). Then, we solve

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{f}.$$

Exercise 9 *Adjust your conjugate gradient method that you implemented in Exercise 8 so that the above system is solved. Repeat the steps in Exercise 8.* △

Remark: *The current preconditioning approach is only useful for the case that $P^{-1}A$ symmetric positive definite. For the current diagonal preconditioner, this symmetry and positive definiteness is certainly satisfied. For generic preconditioners, this condition is not satisfied, even though both A and P are both symmetric positive definite. Then, the preconditioning step needs a treatment that is a little more sophisticated. The interested reader is referred to the lecture notes, or to Segal, Van Kan and Vermolen. Numerical Methods in Scientific Computing. Further, non-symmetric matrices need alternative Krylov subspace methods, such as GMRES, BiCG STAB.*