Universität Tübingen
Mathematisches Institut
Dr. Yanyan Shi
Dr. Yoann Le Hénaff

Tübingen, June 17, 2025

9. Exercise Sheet for Algorithms in Numerical Mathematics

Exercise 25: Show that for the conjugate gradient (CG) method it holds:

$$\frac{(d_k, r_k)}{(Ad_k, d_k)} = \frac{(r_k, r_k)}{(Ad_k, d_k)}, \qquad \frac{(Ad_k, r_{k+1})}{(Ad_k, d_k)} = -\frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)}.$$

Exercise 26: Let the eigenvalues of A (symmetric and positive definite) be ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Show that with $\kappa' = \lambda_2/\lambda_n$, the error in the CG method satisfies:

$$\|x_k - x\|_A \leq 2 \left(\frac{\sqrt{\kappa'} - 1}{\sqrt{\kappa'} + 1} \right)^{k-1} \|x_0 - x\|_A \quad \text{for } k \geq 2.$$

(If $\lambda_1 \gg \lambda_2$, this bound is much stronger than the similar estimate with $\kappa = \lambda_1/\lambda_n$ from the lecture.)
Hint: $q_k(\lambda) = \tilde{q}_{k-1}(\lambda)(\lambda_1 - \lambda)/\lambda_1$.

Programming Exercise 8: Implement the steepest descent method for solving a linear system $Ax = b$ with symmetric positive definite matrix A.

Test your function using:

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 20 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 20 \\ 1 \end{pmatrix}$$

and plot the first 50 iterates graphically. Hint: The visualization becomes particularly clear when plotting the iterates on top of a contour plot of the (to be minimized) function f. For example:

MATLAB

```
hold on;
[x1,x2] = meshgrid(-20:1:20,-20:1:20);
contour(x1,x2,x1.^2 + 20*x2.^2,[0:1:400].^2);
plot(here should be the iterates,'-*','Linewidth',2)
hold off;
```

Julia

```
Using Plots
x1 = LinRange(0,20,100)
x2 = LinRange(-1,1,100)
z = @. x1'^2 + 20 * x2^2
levels = 10.0 .^ LinRange(-1,log10(400))
p = contourf(x1,x2,z,levels=levels)
plot!(p, traj[:,1], traj[:,2],
  marker=:x,color=:red)
```

Comment: The Julia code is more zoomed in, the MATLAB is more zoomed out, you can adapt to your liking!

Programming Exercise 9: Implement the cg-method for solving a linear equation system $Ax = b$ with symmetric positive definite matrix A. Plot the error $\|Ax_k - b\|$ for all k. Then test your function using the following two matrices.

Matrix 1:

MATLAB Code

```
function A = MatrixGenerator1(N)
n = N^2;
% Main diagonal: -4
main_diag = -4 * ones(n, 1);
% Off-diagonals at ±N: -1
off_diag_N = -ones(N * (N - 1), 1);
% Off-diagonals at ±1 within each row block: -1
off_diag_1 = -ones(n - 1, 1);

for i = 1:N-1
  off_diag_1(i * N) = 0;
end
% Create sparse matrix
A = spdiags([off_diag_1, off_diag_N,
 main_diag, off_diag_N,
  off_diag_1],
   [-1, -N, 0, N, 1], n, n);
end
```

Julia Code

```
using SparseArrays

function MatrixGenerator1(N)
n = N^2
# Main diagonal: -4
main_diag = fill(-4.0, n)
# Off-diagonals at ±N: -1
off_diag_N = fill(-1.0, N * (N - 1))
# Off-diagonals at ±1 within each row block: -1
off_diag_1 = fill(-1.0, n - 1)

off_diag_1[N:N:end] .= 0
# Create sparse matrix
A = spdiagm(-1 => off_diag_1,
-N => off_diag_N, 0 => main_diag,
 N => off_diag_N, 1 => off_diag_1)
return A
end
```

Matrix 2:

MATLAB Code

```
function A = MatrixGenerator2(N)
n = N^2;
A = sprandsym(n, 0.2, 0.1) ...
  + spdiags(2 * ones(n, 1), 0, n, n);
end
```

Julia Code

```
using SparseArrays

function MatrixGenerator2(N)
n = N^2
A = sprandsym(n, 0.2, 0.1) +
   spdiagm(0 => fill(2.0, n))
return A
end
```

Solutions are discussed on Tuesday 24.06.2025.

Tutor: Georgios Vretinaris - if you have question just come to my office (C3P16) or write me an email.