MATH-329 Nonlinear optimization
Exercise session 6: Truncated conjugate gradient for trust-region

Instructor: Nicolas Boumal
TAs: Guifré Sánchez, Antoine Gonon

Document compiled on October 13, 2025

1. Output of tCG in case of early termination. Consider the definition $v_n \triangleq v_{n-1} + tp_{n-1}$ and the equation $\|v_n\|^2 = \Delta^2$ (where the unknown is t) from the truncated conjugate gradients algorithm in the course.

1. Show that the equation is indeed quadratic in t.
2. In the context above, argue that there is one positive root and one negative root (in particular, they are distinct). Give an expression for the positive root.

It is possible to show that choosing the positive root induces the smaller model value.

2. First iterate of tCG.

1. Verify that the first iterate of tCG, that is, v_1 as produced by the algorithm in lecture notes, is exactly the Cauchy step.
2. What can you conclude regarding the global convergence of the trust-region method with tCG?

3. Improving the TR method with tCG.

1. Implement the truncated conjugate gradients method (tCG) as explained in the lecture notes.
2. Implement the trust-region algorithm using the tCG algorithm to approximate the solution to the subproblem. You can simply modify the implementation you did for exercise session 5.
3. Run the trust-region algorithm with tCG on the multidimensional Rosenbrock function (see the definition at the end of the exercise sheet). We provide files on Moodle to compute the function value, the gradient and the Hessian. You may use $n = 10$, $x_0 =$ `randn(n,1)`.
4. Compare its performance on this problem with trust-region using the Cauchy point as approximate solution to the trust-region problem. Comment on your observations.

Multidimensional Rosenbrock function. We generalize the Rosenbrock function in n dimensions as

$$f(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right].$$

The vector of ones is the unique global minimum (because the function is non-negative and is zero if and only if all entries are ones). The gradient at $x \in \mathbb{R}^n$ is given by

$$\nabla f(x)_i = \begin{cases} -2(1 - x_1) - 400x_1(x_2 - x_1^2) & \text{if } i = 1 \\ 200(x_i - x_{i-1}^2) - 2(1 - x_i) - 400x_i(x_{i+1} - x_i^2) & \text{if } 1 < i < n \\ 200(x_n - x_{n-1}^2) & \text{if } i = n. \end{cases}$$

The Hessian at x is a symmetric tridiagonal $n \times n$ matrix. The main diagonal and the first diagonal above are given by

$$\begin{bmatrix} 2 + 1200x_1^2 - 400x_2 \\ 202 + 1200x_2^2 - 400x_3 \\ \vdots \\ 202 + 1200x_{n-1}^2 - 400x_n \\ 200 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -400x_1 \\ \vdots \\ -400x_{n-1} \end{bmatrix}$$

respectively. In practice we never build the full matrix but solely compute matrix/vector products. This can be done efficiently because the matrix is sparse.