Otto-von-Guericke-University Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods for Systems and Control Theory

Dr. Martin Köhler, Dr. Pavan Laxmipathy Veluvali
Website: http://www.mpi-magdeburg.mpg.de/scicomp2425

Winterterm 2024/2025

Scientific Computing 1
Handout 9
October 16, 2024

Projection Methods and Conjugate Gradients

- **Projection Method:**
  - Let $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, K_m, L_m m-dimensional subspaces of $\mathbb{R}^n$. $V_m, W_m \in \mathbb{R}^{n \times m}$ with full column rank, containing bases of K_m, L_m respectively. $x_0 \in \mathbb{R}^n$ an initial vector.
  - A **projection method** for $Ax = b$ searches for a solution (approximation) $x_m \in x_0 + K_m$ that satisfies
  $$b - Ax_m \perp L_m. \tag{1}$$
  - If $K_m = L_m$ (1) is called **(Ritz-)Galerkin condition** and the method is called **orthogonal projection method**.
  - If $K_m \neq L_m$ (1) is called **Petrov-Galerkin condition** and the method is called **oblique projection method**.

- **Krylov Subspaces and Krylov Subspace Methods:**
  Let $y \in \mathbb{R}^n$ be an arbitrary vector.
  $$K_m(A, y) = \operatorname{span}\{\, y,\, Ay,\, \ldots,\, A^{m-1}y \,\}$$
  is called the m-th Krylov subspace for A and y. A projection method with $K_m = K_m(A, y)$ is called **Krylov subspace (projection) method.**

- **A prototype Krylov Subspace Method:**

Algorithmus 1 Conjugate Gradient Method

Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$
Output: $x = A^{-1}b$
$p_0 = r_0 = b - Ax_0$, $\alpha_0 = \|r_0\|^2$
for $m := 0, \dots, n-1$ **do**
  if $\alpha_m \neq 0$ **then**
    $v_m = Ap_m$
    $\lambda_m = \frac{\alpha_m}{(v_m, p_m)}$
    $x_{m+1} = x_m + \lambda_m p_m$
    $r_{m+1} = r_m - \lambda_m v_m$
    $\alpha_{m+1} = \|r_{m+1}\|^2$
    $p_{m+1} = r_{m+1} + \frac{\alpha_{m+1}}{\alpha_m} p_m$
  else
    `STOP`
  end if
end for

Algorithmus 2 Preconditioned Conjugate Gradient Method

Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $A^{-1} \approx P \in \mathbb{R}^{n\times n}$
Output: $x = A^{-1}b$
$r_0 = b - Ax_0$, $p_0 = Pr_0$, $\alpha_0 = (r_0, p_0)$
for $m := 0, \dots, n-1$ **do**
  if $\alpha_m \neq 0$ **then**
    $v_m = Ap_m$
    $\lambda_m = \frac{\alpha_m}{(v_m, p_m)_2}$
    $x_{m+1} = x_m + \lambda_m p_m$
    $r_{m+1} = r_m - \lambda_m v_m$
    $z_{m+1} = Pr_{m+1}$
    $\alpha_{m+1} = (r_{m+1}, z_{m+1})_2$
    $p_{m+1} = z_{m+1} + \frac{\alpha_{m+1}}{\alpha_m} p_m$
  else
    `STOP`
  end if
end for