

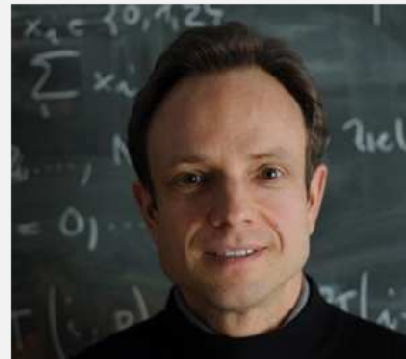
Course Video

Section 2.6: Exploiting Structure when Solving Linear Systems

Prof. R. Hiptmair, SAM, ETH Zurich

Date: September 18, 2020

(C) Seminar für Angewandte Mathematik, ETH Zürich



Prerequisites.

- Proficiency in matrix-vector calculus

Dependencies. Section 2.3, Section 1.4

Duration: 35 minutes

II. Direct Methods for Square Linear Systems of Equations

2.6. Exploiting Structure when Solving Linear Systems

- We have seen *triangular LSE*

Recall: Block GE

regular
↓

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad \begin{matrix} \mathbf{A}_{11} \in \mathbb{K}^{k,k}, \mathbf{A}_{12} \in \mathbb{K}^{k,l}, \mathbf{A}_{21} \in \mathbb{K}^{l,k}, \mathbf{A}_{22} \in \mathbb{K}^{l,l}, \\ \mathbf{x}_1 \in \mathbb{K}^k, \mathbf{x}_2 \in \mathbb{K}^l, \mathbf{b}_1 \in \mathbb{K}^k, \mathbf{b}_2 \in \mathbb{K}^l. \end{matrix} \quad (2.6.0.3)$$

$$\mathbf{x}_1 = \mathbf{A}_{11}^{-1} (\mathbf{b}_1 - \mathbf{A}_{12} \mathbf{x}_2) \quad \text{from 1st row}$$

$$\text{Into 2nd. row} \quad \underbrace{(\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12})}_{\text{Schur complement matrix}} \mathbf{x}_2 = \mathbf{b}_2 - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{b}_1,$$

Schur complement matrix $\in \mathbb{R}^{l \times l}$ Useful, if $\mathbf{A}_{11} \mathbf{x} = \mathbf{c}$ can be solved "easily"

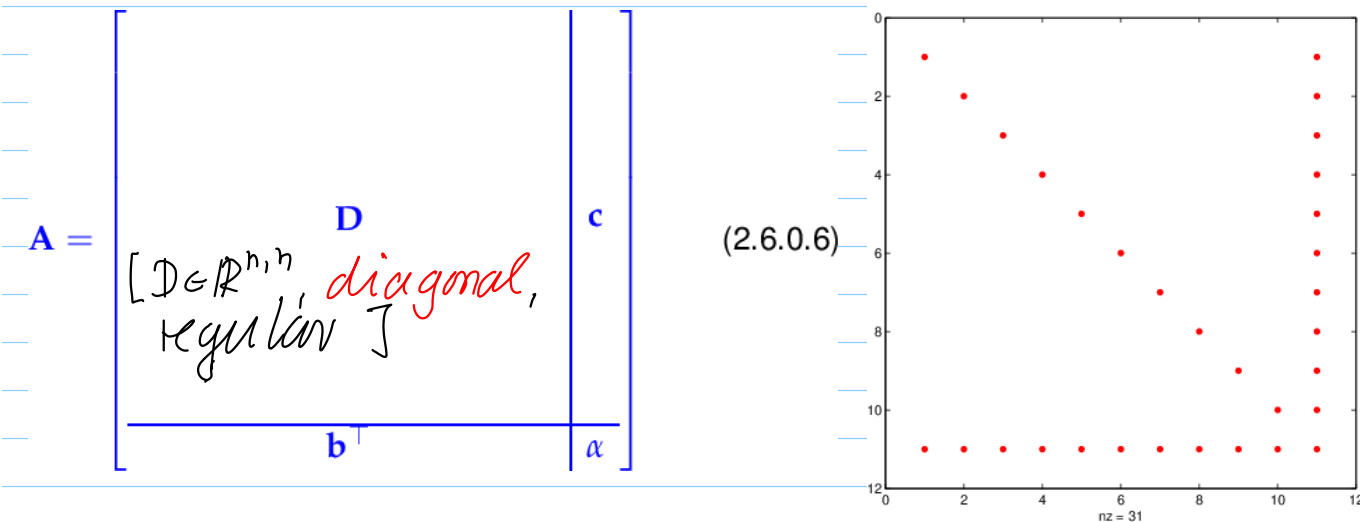
Rem 2.3.2.19: Block LU-decomposition

With

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}}_{\text{block LU-factorization}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{21} \mathbf{A}_{11}^{-1} & \mathbf{I} \end{bmatrix}}_{\text{with Schur complement}} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}, \quad \mathbf{S} := \mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12}. \quad (2.3.2.20)$$

(2)

Ex 2.6.0.5 : LSE with arrow matrix



C++ code 2.6.0.9: Dense Gaussian elimination applied to arrow system → GITLAB

```

2 VectorXd arrowsys_slow(const VectorXd &d, const VectorXd &c, const VectorXd &b,
3                       const double alpha, const VectorXd &y) {
4     int n = d.size();
5     MatrixXd A(n + 1, n + 1); // Empty dense matrix
6     A.setZero(); // Initialize with all zeros.
7     A.diagonal().head(n) = d; // Initializee matrix diagonal from a vector.
8     A.col(n).head(n) = c; // Set rightmost column c.
9     A.row(n).head(n) = b; // Set bottom row b^T.
10    A(n, n) = alpha; // Set bottom-right entry alpha.
11    return A.lu().solve(y); // Gaussian elimination
12 }

```

→ Cast = $O(n^3)$

Block GE :

$$Ax = \left[\begin{array}{c|c} D & c \\ \hline b^T & \alpha \end{array} \right] \begin{bmatrix} x_1 \\ \xi \end{bmatrix} = y := \begin{bmatrix} y_1 \\ \eta \end{bmatrix}, \quad (2.6.0.7)$$

$$\xi = \frac{\eta - b^T D^{-1} y_1}{\alpha - b^T D^{-1} c} \rightarrow \neq 0 \text{ required!} \quad (2.6.0.8)$$

$$x_1 = D^{-1} (y_1 - \xi c).$$

→ $O(n)$ cast

C++ code 2.6.0.10: Solving an arrow system according to (2.6.0.8) → GITLAB

```

2 VectorXd arrowsys_fast(const VectorXd &d, const VectorXd &c, const VectorXd &b,
3                       const double alpha, const VectorXd &y) {
4     int n = d.size();
5     VectorXd z = c.array() / d.array(); // z = D^{-1}c
6     VectorXd w = y.head(n).array() / d.array(); // w = D^{-1}y_1
7     const double den = alpha - b.dot(z); // denominator in (2.6.0.8)
8     // Check for (relatively!) small denominator
9     if (std::abs(den) <
10         std::numeric_limits<double>::epsilon() * (b.norm() + std::abs(alpha))) {
11         throw std::runtime_error("Nearly singular system");
12     }
13     const double xi = (y(n) - b.dot(w)) / den;
14     return (VectorXd(n + 1) << w - xi * z, xi).finished();
15 }

```

→ safe test == 0.0

▽
0 Fast code more vulnerable to round-off : instability

③

2.6.0.12: LSE subject to low-rank modification

Assume: $Ax = \underline{b}$ is easy to solve, because(i) A special structure

(ii) LU-dec. available

Sought: $\tilde{x}$: $\tilde{A}\tilde{x} = \underline{b}$, where $\tilde{A}$ arises from A by changing a single entry $(A)_{i^*, j^*}$

$$A, \tilde{A} \in \mathbb{K}^{n,n}: \tilde{a}_{ij} = \begin{cases} a_{ij} & , \text{if } (i,j) \neq (i^*, j^*), \\ z + a_{ij} & , \text{if } (i,j) = (i^*, j^*), \end{cases} \quad i^*, j^* \in \{1, \dots, n\}. \quad (2.6.0.13)$$

 $z \in \mathbb{R}$ $\underline{e}_i =$ Cartesian basis vector

$$\tilde{A} = A + z \cdot \underline{e}_{i^*} \underline{e}_{j^*}^T \quad \leftarrow \begin{array}{l} n \times n \text{ tensor-product} \\ \text{matrix, rank} = 1 \end{array} \quad (2.6.0.14)$$

$$\begin{bmatrix} 0 & \text{---} & 0 \\ | & & | \\ 0 & \text{---} & 0 \end{bmatrix} \quad \begin{array}{l} \leftarrow i^* \\ \uparrow j^* \end{array}$$

More general: rank-1 modification

$$\tilde{A} = A + \underline{u} \underline{v}^T, \quad \underline{u}, \underline{v} \in \mathbb{R}^n$$

Trick: Block elimination

$$\begin{bmatrix} A & \underline{u} \\ \underline{v}^T & -1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \underline{\zeta} \end{bmatrix} = \begin{bmatrix} \underline{b} \\ 0 \end{bmatrix}$$

$$\text{block row II} \Rightarrow \underline{\zeta} = \underline{v}^T \tilde{x} \xRightarrow{I} \underbrace{(A + \underline{u} \underline{v}^T)}_{\tilde{A}} \tilde{x} = \underline{b}$$

$$\begin{aligned} \text{block row I} &\Rightarrow \tilde{x} = A^{-1}(\underline{b} - \underline{\zeta} \underline{u}) \\ &\xRightarrow{II} \underline{\zeta} = \underline{v}^T A^{-1}(\underline{b} - \underline{\zeta} \underline{u}) \\ &\Rightarrow \underline{\zeta} = \frac{\underline{v}^T A^{-1} \underline{b}}{1 + \underline{v}^T A^{-1} \underline{u}} \end{aligned}$$

$$\tilde{x} = A^{-1} \underline{b} - \frac{A^{-1} \underline{u} (\underline{v}^T (A^{-1} \underline{b}))}{1 + \underline{v}^T (A^{-1} \underline{u})} \quad \leftarrow \neq 0! \quad (2.6.0.22)$$

 $\rightarrow$ have to solve two LSE $Ax = \dots$ C++ code 2.6.0.23: Solving a rank-1 modified LSE $\rightarrow$ GITLAB

```

2 // Solving rank-1 updated LSE based on (2.6.0.22)
3 template <class LUDec>
4 Eigen::VectorXd smw(const LUDec &lu, const Eigen::VectorXd &u,
5                     const Eigen::VectorXd &v, const Eigen::VectorXd &b) {
6     const Eigen::VectorXd z = lu.solve(b); // z = A^{-1}b
7     const Eigen::VectorXd w = lu.solve(u); // w = A^{-1}u
8     double alpha = 1.0 + v.dot(w); // Compute denominator of (2.6.0.22)
9     double beta = v.dot(z); // Factor for numerator of (2.6.0.22)
10    if (std::abs(alpha) < std::numeric_limits<double>::epsilon() * std::abs(beta)) {
11        throw std::runtime_error("A nearly singular");
12    }
13    else {
14        return (z - w * beta / alpha); // see (2.6.0.22)
15    }
16 }

```

safe check $\neq 0$

$\left. \begin{array}{l} \text{cast } O(n^2), \text{ if LU-dec.} \\ O(n) \text{ cast} \end{array} \right\}$

4 Generalization of (2.6.0.22) :

Lemma 2.6.0.21. Sherman-Morrison-Woodbury formula

For regular $\mathbf{A} \in \mathbb{K}^{n,n}$, and $\mathbf{U}, \mathbf{V} \in \mathbb{K}^{n,k}$, $n, k \in \mathbb{N}$, $k \leq n$, holds

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^H)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^H\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^H\mathbf{A}^{-1},$$

if $\mathbf{I} + \mathbf{V}^H\mathbf{A}^{-1}\mathbf{U}$ is regular.

General rank k -matrix

5

7