Linear Algebra II

Direct Solution of Systems of Linear Equations

Uwe Naumann

Informatik 12:
Software and Tools for Computational Engineering

RWTH Aachen University

Outline

Outline

Objective

- Introduction to direct methods for the solution of systems of linear equations

Learning Outcomes

- You will understand
  - direct linear solvers
    - $A = L \cdot R$
    - $A = L \cdot L^T$
    - $A = Q \cdot R$
  - condition of linear systems
- You will be able to
  - use Eigen for solving linear systems
  - perform error analysis for systems of linear equations

We consider the solution of systems of n linear equations

$$A \cdot \mathbf{x} = \mathbf{b}$$

with invertible system matrix $A \in \boldsymbol{R}^{n \times n}$, right-hand side $\mathbf{b} \in \boldsymbol{R}^n$ and unknown $\mathbf{x} \in \boldsymbol{R}^n$ to be determined such that all n linear equations are satisfied at the same time.

Approaches to the solution of linear systems separate into direct and indirect (iterative) methods. The focus of this module is on direct methods based on factorizations of the system matrix.

Scalar problems yield linear equations $a \cdot x = b$. They are numerically robust and easily solved. Small errors in $b \in \boldsymbol{R}$ imply small (same order) errors in $x \in \boldsymbol{R}$.

Higher-dimensional problems yield systems of linear equations

$$A \cdot \mathbf{x} = \mathbf{b} \, .$$

Small changes in $\mathbf{b} \in \boldsymbol{R}^n$ can yield large changes in $\mathbf{x} \in \boldsymbol{R}^n$ due to poor conditioning of $A \in \boldsymbol{R}^{n \times n}$. For example,

$$\begin{aligned} x + y &= 2 \\ x + 1.001 \cdot y &= 2 \end{aligned} \quad \Rightarrow \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

while

$$\begin{aligned} x + y &= 2 \\ x + 1.001 \cdot y &= 2.001 \end{aligned} \quad \Rightarrow \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \, .$$

We observe $\mathrm{cond}(A) \approx 4004$. (See below for further details.)

Outline

The following statements are equivalent:

1. A is regular.
2. A solution to $A \cdot \mathbf{x} = \mathbf{b}$ exists for any $\mathbf{b}$.
3. A solution to $A \cdot \mathbf{x} = \mathbf{b}$ is unique, if it exists.
4. $\forall \mathbf{x} : A \cdot \mathbf{x} = 0 \Rightarrow \mathbf{x} = \mathbf{0}$
5. the columns (rows) of A are linearly independent
6. the inverse A^{-1} of A exists and $A^{-1} \cdot A = A \cdot A^{-1} = I_n$, where $I_n \in \boldsymbol{R}^{n \times n}$ denotes the identity in $\boldsymbol{R}^n$, i.e, $\forall \mathbf{v} \in \boldsymbol{R}^n : \; I_n \cdot \mathbf{v} = \mathbf{v}$.
7. $\det(A) \neq 0$ (nonzero determinant of A)

Diagonal system matrices yield linear systems the solution of which amounts to the solution of mutually independent linear equations.

All diagonal elements need to be nonzero to ensure regularity.

Inversion becomes particularly simple due to

$$\begin{pmatrix} d_{0,0} & & & \\ & \ddots & & \\ & & d_{i,i} & \\ & & & \ddots \\ & & & & d_{n-1,n-1} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{d_{0,0}} & & & \\ & \ddots & & \\ & & \frac{1}{d_{i,i}} & \\ & & & \ddots \\ & & & & \frac{1}{d_{n-1,n-1}} \end{pmatrix}.$$

Lower / upper triangular system matrices yield linear systems the solution of which amounts to simple forward / backward substitution.

1. Lower triangular system by forward substitution, e.g.

$$\begin{pmatrix} 1 & 0 \\ -\frac{1}{3} & 1 \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} 1 \\ \frac{4}{3} \end{pmatrix}$$

2. Upper triangular system by backward substitution, e.g.

$$\begin{pmatrix} 3 & 1 \\ 0 & \frac{7}{3} \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{4}{3} \end{pmatrix} \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} \frac{1}{7} \\ \frac{4}{7} \end{pmatrix}$$

For orthogonal $A \in \boldsymbol{R}^{n \times n}$ the solution of $A \cdot \mathbf{x} = \mathbf{b}$ simplifies as

$$\mathbf{x} = A^{-1} \cdot \mathbf{b} = A^T \cdot \mathbf{b} \,,$$

for example,

$$\begin{pmatrix} \frac{4}{5} & \frac{3}{5} \\ \frac{3}{5} & -\frac{4}{5} \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

Examples for orthogonal matrices are

- permutation matrices obtained by permuting rows and/or columns of the identity
- rotation obtained by embedding 2x2 rotation into identity (Givens)
- reflection (Householder)

Direct methods for the solution of linear systems aim to represent A as a product of diagonal, triangular and/or orthogonal matrices, e.g,

$A = L \cdot R$ with lower triangular $L \in \boldsymbol{R}^{n \times n}$ and upper triangular $R \in \boldsymbol{R}^{n \times n}$

$$\Rightarrow \quad L \cdot \mathbf{v} = \mathbf{b} \; ; \quad R \cdot \mathbf{x} = \mathbf{v}$$

$A = L \cdot L^T$ with lower triangular $L \in \boldsymbol{R}^{n \times n}$

$$\Rightarrow \quad L \cdot \mathbf{v} = \mathbf{b} \; ; \quad L^T \cdot \mathbf{x} = \mathbf{v}$$

$A = L \cdot D \cdot L^T$ with diagonal $D \in \boldsymbol{R}^{n \times n}$ and lower triangular $L \in \boldsymbol{R}^{n \times n}$

$$\Rightarrow \quad L \cdot \mathbf{v} = \mathbf{b} \; ; \quad D \cdot L^T \cdot \mathbf{x} = \mathbf{v}$$

$A = Q \cdot R$ with orthogonal $Q \in \boldsymbol{R}^{n \times n}$ and upper triangular $R \in \boldsymbol{R}^{n \times n}$

$$\Rightarrow \quad \mathbf{v} = Q^T \cdot \mathbf{b} \; ; \quad R \cdot \mathbf{x} = \mathbf{v}$$

Outline

From

$$\begin{pmatrix} \alpha & \mathbf{b}^T \\ \mathbf{a} & A_* \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \mathbf{a}_1 & L_* \end{pmatrix} \cdot \begin{pmatrix} \alpha & \mathbf{b}^T \\ 0 & R_* \end{pmatrix}$$

with $\alpha \in \boldsymbol{R}$, $\mathbf{a}, \mathbf{b} \in \boldsymbol{R}^{n-1}$ and lower / upper triangular matrices $L_*, R_* \in \boldsymbol{R}^{(n-1)\times(n-1)}$ follows

$$\mathbf{a} = \alpha \cdot \mathbf{a}_1$$
$$A_* = \mathbf{a}_1 \cdot \mathbf{b}^T + L_* \cdot R_*$$

and hence

$$\mathbf{a}_1 = \frac{\mathbf{a}}{\alpha}$$
$$L_* \cdot R_* = A_* - \mathbf{a}_1 \cdot \mathbf{b}^T \ .$$

$R = L^{-1} \cdot A$ is generated as

$$R = L_{n-1}^{-1} \cdot \ldots \cdot L_0^{-1} \cdot A \,,$$

where $L_i^{-1} \cdot A$ generates zeros in the i-th column of A below the diagonal, i.e,

$$L_i^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\frac{a_{i+1}}{a_i} & 1 & \\ & & \vdots & & \ddots \\ & & -\frac{a_{n-1}}{a_i} & & 1 \end{pmatrix}$$

The L_i turn out to be atomic lower triangular (I_n and off-diagonal nonzeros in single column) yielding

$$L_i = (L_i^{-1})^{-1} = 2 \cdot I_n - L_i^{-1} \,,$$

i.e, inversion amounts to sign switch for off-diagonal entries.

Division by values of diagonal entries close to machine precision causes trouble.

```
A=
1e-07 4 -2
    4 9 -3
   -2 -3 7

L_0^{-1}=
     1 0 0
-4e+07 1 0
 2e+07 0 1

L_0^{-1}*A=
   1e-07 4 -2
       0 -1.6e+08 8e+07
       0 8e+07 -4e+07
```

```
L_0=
      1 0 0
 4e+07 1 0
-2e+07 0 1

L_0*L_0^{-1}*A=
1e-07 4 -2
    4 16 0
   -2 0 8
```

It can be avoided by permuting rows (partial pivoting) and, optionally, columns (full pivoting) of A such that the entry with the respective maximum absolute value appears on the diagonal.

... are obtained from identity matrices by permuting columns

... switch rows [columns] in A when multiplied from left [right]

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} = \begin{pmatrix} 4 & 5 \\ 2 & 3 \end{pmatrix} \quad \left[\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 5 & 4 \end{pmatrix} \right]$$

... are orthogonal (not necessarily symmetric), i.e. $P^{-1} = P^T$.

Proof: Let $Q = P^T$ and $R = P \cdot Q$. From $p_{i,j} = 1$ and $p_{i,k} = 0$, $k \neq j$, follows $q_{j,i} = 1$ and $q_{k,i} = 0$, $k \neq j$. Presence of a single unit entry per row of P (column of Q) implies $r_{i,i} = 1$ for all $i = 0, \ldots, n-1$ and, hence, $R = I_n$.

"Eigen is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms."[1]

Eigen provides various dense as well as sparse solvers for linear systems including

- $A = L \cdot R$ ([Partial/Full]PivLU)
- $A = L \cdot L^T$ (LLT for A symmetric positive definite)
- $A = L \cdot D \cdot L^T$ (LDLT for A symmetric positive or negative semi-definite)
- $A = Q \cdot R$ (HouseholderQR, [Col/Full]PivHouseholderQR)

[1] `http://eigen.tuxfamily.org`

$A = L \cdot R$

Solution with Eigen


```
1  #include "Eigen/Dense"
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6    using MT=Eigen::Matrix<float,3,3>;
7    using VT=Eigen::Matrix<float,3,1>;
8    MT A=MT::Random(); cout << "A=" << endl << A << endl;
9    VT b=VT::Random(); cout << "b^T=" << b.transpose() << endl;
10   Eigen::PartialPivLU<MT> LU(A);
11   MT L=LU.matrixLU().triangularView<Eigen::StrictlyLower>();
12   L+=MT::Identity();
13   MT R=LU.matrixLU().triangularView<Eigen::Upper>();
14   MT P=LU.permutationP();
15   cout << "P=" << endl << P << endl;
16   cout << "L=" << endl << L << endl << "R=" << endl << R << endl;
17   cout << "P*L*R=" << endl << P*L*R << endl;
18   VT x=LU.solve(b);
19   cout << "x^T=" << x.transpose() << endl;
20   cout << "(A*x)^T=" << (A*x).transpose() << endl;
21   return 0;
22 }
```

$A = L \cdot R$

Solution with Eigen


```
A=
 0.680375 0.59688 -0.329554
-0.211234 0.823295 0.536459
 0.566198 -0.604897 -0.444451

b^T=
 0.10794 -0.0452059 0.257742

P=
1 0 0
0 0 1
0 1 0

L=
         1 0 0
 0.832185 1 0
-0.310467 -0.915573 1

R=
 0.680375 0.59688 -0.329554
        0 -1.10161 -0.1702
        0 0 0.278313
```

```
P*L*R
 0.680375 0.59688 -0.329554
-0.211234 0.823295 0.536459
 0.566198 -0.604897 -0.444451

x^T=
 0.60876 -0.231281 0.510379

(A*x)^T=
 0.10794 -0.0452059 0.257742
```

Outline

(Semi-)Definiteness

A symmetric matrix $A \in \boldsymbol{R}^{n \times n}$ is

- positive definite (s.p.d.) if

$$\forall \mathbf{v} \neq 0 \in \boldsymbol{R}^n : \quad \mathbf{v}^T \cdot A \cdot \mathbf{v} > 0 \, .$$

- negative definite if

$$\forall \mathbf{v} \neq 0 \in \boldsymbol{R}^n : \quad \mathbf{v}^T \cdot A \cdot \mathbf{v} < 0 \, .$$

- positive semi-definite if

$$\forall \mathbf{v} \neq 0 \in \boldsymbol{R}^n : \quad \mathbf{v}^T \cdot A \cdot \mathbf{v} \geq 0 \, .$$

- negative semi-definite if

$$\forall \mathbf{v} \neq 0 \in \boldsymbol{R}^n : \quad \mathbf{v}^T \cdot A \cdot \mathbf{v} \leq 0 \, .$$

From

$$\begin{pmatrix} \alpha & \mathbf{a}^T \\ \mathbf{a} & A_* \end{pmatrix} = \begin{pmatrix} \rho & 0 \\ \mathbf{r} & R_*^T \end{pmatrix} \begin{pmatrix} \rho & \mathbf{r}^T \\ 0 & R_* \end{pmatrix}$$

follows

$$\alpha = \rho^2; \quad \mathbf{a}^T = \rho \cdot \mathbf{r}^T; \quad A_* = \mathbf{r} \cdot \mathbf{r}^T + R_*^T \cdot R_*$$

and hence

$$\rho = \sqrt{\alpha}$$
$$\mathbf{r} = \frac{\mathbf{a}}{\rho}$$
$$R_*^T \cdot R_* = A_* - \mathbf{r} \cdot \mathbf{r}^T \ .$$

A is s.p.d. as $\alpha \leq 0$ is not permitted.

$A = L \cdot L^T$

Solution with Eigen


```
1  #include "Eigen/Dense"
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6    using MT=Eigen::Matrix<float,3,3>;
7    using VT=Eigen::Matrix<float,3,1>;
8    MT A=MT::Random(); A=A*A.transpose();
9    cout << "A=" << endl << A << endl;
10   VT b=VT::Random();
11   cout << "b^T=" << b.transpose() << endl;
12   Eigen::LLT<MT> LLT(A);
13   MT L=LLT.matrixLLT().triangularView<Eigen::Lower>();
14   cout << "L=" << endl << L << endl;
15   VT x=LLT.solve(b);
16   cout << "x^T=" << x.transpose() << endl;
17   cout << "(A*x)^T=" << (A*x).transpose() << endl;
18   return 0;
19 }
```

Solution with Eigen


```
1  A=
2   0.927783 0.170897 0.170647
3   0.170897 1.01022 −0.856039
4   0.170647 −0.856039 0.884018
5
6  b^T= 0.10794 −0.0452059 0.257742
7
8  L=
9   0.963215 0 0
10  0.177424 0.989315 0
11  0.177164 −0.897057 0.218904
12
13 x^T= −1.18143 3.31816 3.73276
14
15 (A∗x)^T= 0.10794 −0.045206 0.257742
```

Outline

For $A = (a_{i,j}) \in \boldsymbol{R}^{n \times n}$ Householder reflections H^j applied to all columns $j = 0, \ldots, n-1$ yield transformations

$$H^j \cdot \begin{pmatrix} * \\ \vdots \\ * \\ a_{j,j} \\ a_{j+1,j} \\ \vdots \\ a_{n-1,j} \end{pmatrix} \Rightarrow \begin{pmatrix} * \\ \vdots \\ * \\ \sqrt{\sum_{i=j}^{n-1} a_{i,j}^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

($*$-entries are left unchanged) implying $H^{n-1} \cdot \ldots \cdot H^0 \cdot A = R$ and, hence, from $Q \cdot R = A$,

$$Q^{-1} = Q^T = H^{n-1} \cdot \ldots \cdot H^0 \ .$$

See modules on Linear Regression for further details on orthogonalization.

$A = Q \cdot R$

Solution with Eigen


```
1  #include "Eigen/Dense"
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6    using MT=Eigen::Matrix<float,3,3>;
7    using VT=Eigen::Matrix<float,3,1>;
8    MT A=MT::Random(); cout << "A=" << endl << A << endl;
9    VT b=VT::Random(); cout << "b^T=" << b.transpose() << endl;
10   Eigen::HouseholderQR<MT> QR(A);
11   MT Q=QR.householderQ();
12   MT R=QR.matrixQR().triangularView<Eigen::Upper>();
13   cout << "Q=" << endl << Q << endl;
14   cout << "R=" << endl << R << endl;
15   VT x=QR.solve(b);
16   std::cout << "x^T=" << x.transpose() << std::endl;
17   std::cout << "(A*x)^T=" << (A*x).transpose() << std::endl;
18   return 0;
19 }
```

Solution with Eigen


```
A=
 0.680375 0.59688 −0.329554
−0.211234 0.823295 0.536459
 0.566198 −0.604897 −0.444451

b^T= 0.10794 −0.0452059 0.257742

Q=
 −0.74766 −0.58412 −0.315923
 0.232124 −0.675592 0.699782
−0.622192 0.449866 0.640701

R=
−0.910006 0.121205 0.647453
        0 −1.17698 −0.369871
        0 0 0.194758

x^T= 0.60876 −0.231281 0.510379

(A*x)^T= 0.10794 −0.0452058 0.257742
```

Outline

Consider the solution of the linear equation

$$a \cdot y = x \quad \Rightarrow \quad y = a^{-1} \cdot x = \frac{x}{a}$$

with $x, y, a \in \boldsymbol{R}$ and $a \neq 0$ for an absolute error in the right-hand side of Δx yielding

$$a \cdot (y + \Delta y) = a \cdot y + a \cdot \Delta y = x + \Delta x$$

and hence an absolute error in the result of

$$\Delta y = a^{-1} \cdot \Delta x \quad \Rightarrow \quad |\Delta y| = |a^{-1}| \cdot |\Delta x| \quad .$$

The relative error in the result becomes equal to

$$|\delta y| \equiv \frac{|\Delta y|}{|y|} = |a| \cdot |a^{-1}| \cdot \frac{|\Delta x|}{|x|} = \frac{|\Delta x|}{|x|} = |\delta x| \quad .$$

The above can be formulated in terms of the condition of

$$y = f(x) = a^{-1} \cdot x = \frac{x}{a} .$$

Absolute condition

$$\mathcal{K}_f(x) = \lim_{\Delta x \to 0} \frac{|\Delta y|}{|\Delta x|} = \lim_{\Delta x \to 0} \left| \frac{\Delta y}{\Delta x} \right| = \lim_{\Delta x \to 0} \left| \frac{\Delta x}{a \cdot \Delta x} \right| = \frac{1}{|a|} = f'(x) .$$

(Relative) Condition

$$\begin{aligned} \kappa_f(x) &= \lim_{\Delta x \to 0} \frac{|\delta y|}{|\delta x|} = \lim_{\Delta x \to 0} \left| \frac{\delta y}{\delta x} \right| \\ &= \lim_{\Delta x \to 0} \left| \frac{\Delta y}{\Delta x} \cdot \frac{|x|}{|y|} \right| = \lim_{\Delta x \to 0} \left| \frac{\Delta x}{a \cdot \Delta x} \frac{|a| \cdot |x|}{|x|} \right| = \left| \frac{|a|}{a} \right| = 1 . \end{aligned}$$

Consider the solution of the system of linear equations

$$A \cdot \mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = A^{-1} \cdot \mathbf{b}$$

with invertible system matrix $A \in \boldsymbol{R}^{n \times n}$ for an absolute error in the right-hand side of $\Delta\mathbf{b}$ yielding $A \cdot (\mathbf{x} + \Delta\mathbf{x}) = A \cdot \mathbf{x} + A \cdot \Delta\mathbf{x} = \mathbf{b} + \Delta\mathbf{b}$ and hence an absolute error in the result of

$$\Delta\mathbf{x} = A^{-1} \cdot \Delta\mathbf{b} \quad \Rightarrow \quad \|\Delta\mathbf{x}\| = \|A^{-1} \cdot \Delta\mathbf{b}\| \leq \|A^{-1}\| \cdot \|\Delta\mathbf{b}\| = \left\| \frac{d\mathbf{x}}{d\mathbf{b}} \right\| \cdot \|\Delta\mathbf{b}\|$$

for any vector-induced (hence, submultiplicative) matrix norm $\|\cdot\|$.
From $\|\mathbf{b}\| = \|A \cdot \mathbf{x}\| \leq \|A\| \cdot \|\mathbf{x}\|$ follows $\frac{\|\mathbf{b}\|}{\|A\|} \leq \|\mathbf{x}\|$ and hence the following estimate of the relative error:

$$\|\delta\mathbf{x}\| = \frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} = \|A\| \cdot \|A^{-1}\| \cdot \|\delta\mathbf{b}\| = \mathrm{cond}(A) \cdot \|\delta\mathbf{b}\| \,.$$

The numerical stability of a system of linear equations $A \cdot \mathbf{x} = \mathbf{b}$ wrt. perturbations $\Delta\mathbf{b}$ in $\mathbf{b}$ depends on the condition of its system matrix.

For

$$\mathbf{x} = A^{-1} \cdot \mathbf{b}$$

the above error analysis yields an absolute condition of

$$\mathcal{K}_{\mathbf{x}}(\mathbf{b}) = \lim_{\Delta\mathbf{b}\to 0} \frac{\|\Delta\mathbf{x}\|}{\|\Delta\mathbf{b}\|} \leq \|A^{-1}\| = \left\|\frac{d\mathbf{x}}{d\mathbf{b}}\right\|$$

resulting in a (relative) condition of

$$\kappa_{\mathbf{x}}(\mathbf{b}) = \lim_{\Delta\mathbf{b}\to 0} \frac{\|\delta\mathbf{x}\|}{\|\delta\mathbf{b}\|} \leq \|A\| \cdot \|A^{-1}\| \equiv \mathrm{cond}(A)$$

Outline

Summary

- direct linear solvers
  - $A = L \cdot R$
  - $A = L \cdot L^T$
  - $A = Q \cdot R$
- condition of linear systems

Next Steps

- Inspect sample code.
- Practice use of Eigen.
- Continue the course to find out more ...