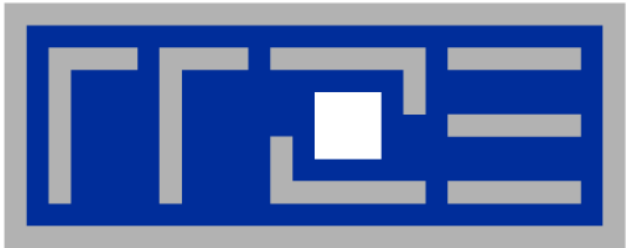


# Elementary Numerical Mathematics

Supporting material for

Iterative Techniques for LSE

Norms of Vectors and Matrices

Jacobi

Gauss-Seidel

Winter Term 2019/20

**Gerhard Wellein**

**Department for Computer Science**

**HPC Services, Regionales Rechenzentrum Erlangen (RRZE)**

# Vector norms – Basic properties (I)

**Definition (III.1.)**

**A vector norm on $\mathbb{R}^n$ is a function $\|\ \ \|$ from $\mathbb{R}^n$ into $\mathbb{R}$ with following properties:**

(1) $\|x\| \geq 0$ **for all** $x \in \mathbb{R}^n$

(2) $\|x\| = 0 \Leftrightarrow x = 0$

(3) $\|\alpha x\| = |\alpha| \, \|x\|$ **for all** $x \in \mathbb{R}^n$ **and** $\alpha \in \mathbb{R}$

(4) $\|x + y\| \leq \|x\| + \|y\|$ **for all** $x, y \in \mathbb{R}^n$

# Vector norms – Basic properties (II)

**Definition (III.2.)**

**The $L_2$ and $L_\infty$norm for the vector $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ are defined by:**

(1) $\|x\|_2 = \left\{\sum_{i=1}^{n} x_i^2\right\}^{1/2}$

(2) $\|x\|_\infty = \max_{i=1,\dots n} |x_i|$

# Vector norms – Basic properties (III)

**Definition (III.3.)**

**The $L_2$ and $L_\infty$ distances between $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ and $y = (y_1, \dots, y_n)^T \in \mathbb{R}^n$ are defined by:**

(1) $$\|x - y\|_2 = \left\{\sum_{i=1}^{n} (x_i - y_i)^2\right\}^{1/2}$$

(2) $$\|x - y\|_\infty = \max_{i=1,\dots n} |x_i - y_i|$$

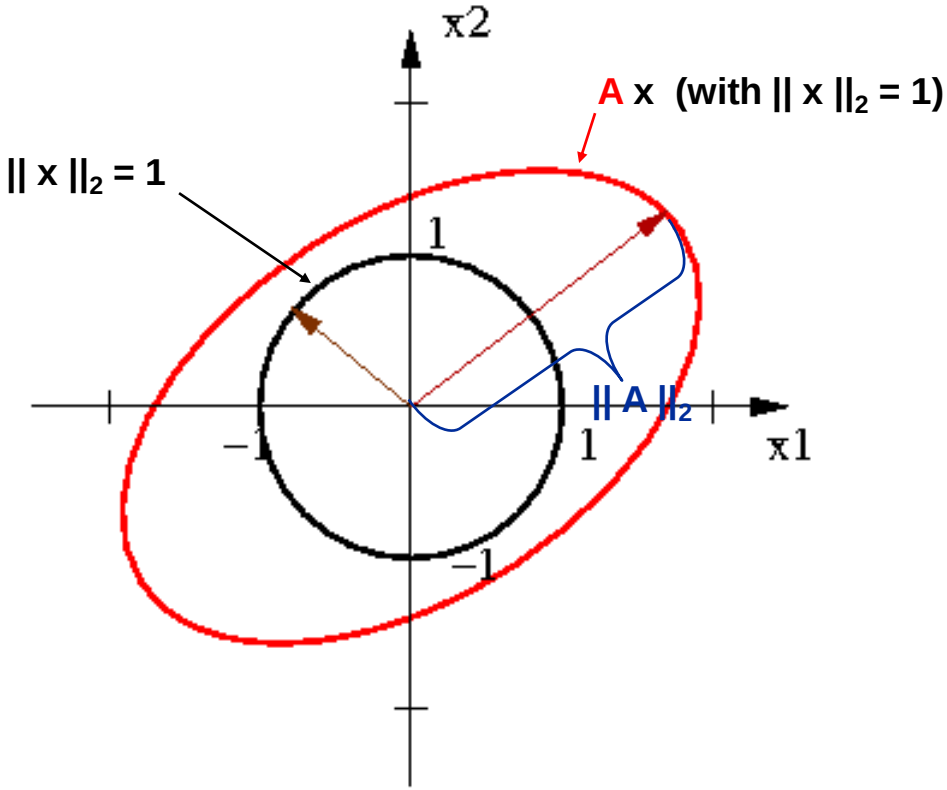
# Vector norms – Basic properties (IV)

**Definition (III.4.) „Cauchy-Bunyakovsky-Schwarz“**

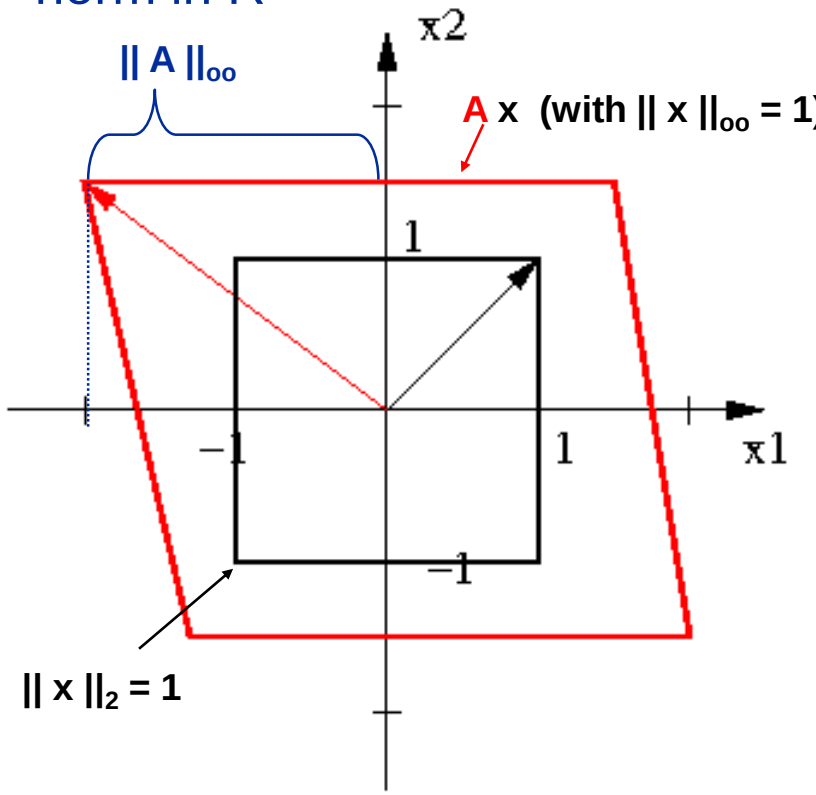
**For each $x \in \mathbb{R}^n$ and y $\in \mathbb{R}^n$ the following inequality holds:**

$$x^T y = \sum_{i=1}^{n} x_i y_i \leq \left\{\sum_{i=1}^{n} x_i^2\right\}^{\frac{1}{2}} \left\{\sum_{i=1}^{n} y_i^2\right\}^{\frac{1}{2}} = \|x\|_2 \ \ y\|_2$$

## Matrix norm induced by $L_2$ vector norm in $R^2$




## Matrix norm induced by $L_{oo}$ vector norm in $R^2$



# Iterative Techniques for LSE

## Motivation: Jacobi method

E1: $10\,x_1 - x_2 + 2\,x_3 = 6$

E2: $-x_1 + 11\,x_2 - x_3 + 3\,x_4 = 25$

E3: $2\,x_1 - x_2 + 10\,x_3 - x_4 = -11$

E4: $3\,x_2 - x_3 + 8\,x_4 = 15$

$$
\begin{aligned}
x_1 &= (1/10) * \{ \quad x_2 - 2\,x_3 \quad \} &&+ (1/10) * 6 \\
x_2 &= (1/11) * \{ x_1 + x_3 - 3\,x_4 \} &&+ (1/11) * 25 \\
x_3 &= (1/10) * \{-2\,x_1 + x_2 + x_4 \} &&- (1/10) * 11 \\
x_4 &= (1/8) * \{ \quad -3\,x_2 + x_3 \quad \} &&+ (1/8) * 15
\end{aligned}
$$

Define: $\mathbf{x}^{(1)} = \mathbf{T}\,\mathbf{x}^{(0)} + \mathbf{c}$

with: $\mathbf{c} = (6/10, 25/11, -11/10, 15/8)$

$$
\mathbf{T} = \begin{pmatrix} 0 & 1/10 & -2/10 & 0 \\ 1/11 & 0 & 1/11 & -3/11 \\ -2/10 & 1/10 & 0 & 1/10 \\ 0 & -3/8 & 1/8 & 0 \end{pmatrix}
$$

# Iterative Techniques for LSE
## Jacobi algorithm

```
Input:N;A := matrix(n,n); x:= vector(n);b:= vector(n);X0:= vector(n)
Output: x := vector(n);          % approximation of solution Ax=b
                                 % after k-Jacobi steps
k=1
while (k <= N) do
    for i= 1 to n do   x_i = 1/a_ii ( - Σ_{j=1, j≠i}^{n} (a_ij XO_j) + b_i )

    If (|| x- XO || ) < TOL  then STOP %(Solution reached;)
    k= k + 1

        for i= 1 to n do   XO_i = x_i

    end do
  end do
end while
```

|| ||$_{oo}$ is frequently used in this context

# Iterative Techniques for LSE

## Motivation: Gauss-Seidel method

E1: $10\,x_1 - x_2 + 2\,x_3 = 6$

E2: $-x_1 + 11\,x_2 - x_3 + 3\,x_4 = 25$

E3: $2\,x_1 - x_2 + 10\,x_3 - x_4 = -11$

E4: $3\,x_2 - x_3 + 8\,x_4 = 15$

Use coefficients from the previous iteration

$$\mathbf{x_1^{(k)}} = (1/10) * \{ x_2^{(k-1)} - 2\,x_3^{(k-1)} + 6 \}$$

$$\mathbf{x_2^{(k)}} = (1/11) * \{ \mathbf{x_1^{(k)}} + x_3^{(k-1)} - 3\,x_4^{(k-1)} + 25 \}$$

$$\mathbf{x_3^{(k)}} = (1/10) * \{ -2\,\mathbf{x_1^{(k)}} + \mathbf{x_2^{(k)}} + x_4^{(k-1)} - 11 \}$$

$$\mathbf{x_4^{(k)}} = (1/8) * \{ -3\,\mathbf{x_2^{(k)}} + \mathbf{x_3^{(k)}} + 15 \}$$

Use coefficients updated in the same iteration

**$\{ x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)} \}$ must be computed successively starting with $x_1^{(k)}$**

```
Input:N; A:= matrix(n,n);x:= vector(n);b:= vector(n); X0:= vector(n)
Output: x := vector(n);          % approximation of solution Ax=b
                                 % after k Gauss-Seidel steps

k=1
while (k <= N) do
    for i= 1 to n do   x_i = 1/a_ii ( - Σ_{j=1}^{i-1} a_ij x_j - Σ_{j=i+1}^{n} a_ij XO_j + b_i )

    If (|| x- XO || ) < TOL  then STOP (Solution reached)
    k= k + 1

        for i= 1 to n do   XO_i = x_i

    end do
  end do
end while
```

# Iterative Techniques for LSE

## Jacobi – approximating the solution

| | Exact solution | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|---|
| | | 1,000000 | 2,000000 | -1,000000 | 1,000000 |
| *Initial guess* | *0* | *0* | *0* | *0* | *0* |
| Solution after k Jacobi steps | 1 | 0,600000 | 2,272727 | -1,100000 | 1,875000 |
| | 2 | 1,047273 | 1,715909 | -0,805227 | 0,885227 |
| | 3 | 0,932636 | 2,053306 | -1,049341 | 1,130881 |
| | 4 | 1,015199 | 1,953696 | -0,968109 | 0,973843 |
| | 5 | 0,988991 | 2,011415 | -1,010286 | 1,021351 |
| | 6 | 1,003199 | 1,992241 | -0,994522 | 0,994434 |
| | 7 | 0,998128 | 2,002307 | -1,001972 | 1,003594 |
| | 8 | 1,000625 | 1,998670 | -0,999036 | 0,998888 |
| | 9 | 0,999674 | 2,000448 | -1,000369 | 1,000619 |
| | 10 | 1,000119 | 1,999768 | -0,999828 | 0,999786 |
| | 11 | 0,999942 | 2,000085 | -1,000068 | 1,000109 |
| | 12 | 1,000022 | 1,999959 | -0,999969 | 0,999960 |
| | 13 | 0,999990 | 2,000016 | -1,000013 | 1,000019 |

# Iterative Techniques for LSE

## Gauss-Seidel – approximating the solution

| | Exact solution | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|---|
| | | **1,000000** | **2,000000** | **-1,000000** | **1,000000** |
| ***Initial guess*** | | *0* | *0* | *0* | *0* |
| **Solution after k Gauss-Seidel steps** | 1 | 0,600000 | 2,327273 | -0,987273 | 0,878864 |
| | 2 | 1,030182 | 2,036938 | -1,014456 | 0,984341 |
| | 3 | 1,006585 | 2,003555 | -1,002527 | 0,998351 |
| | 4 | 1,000861 | 2,000298 | -1,000307 | 0,999850 |
| | 5 | 1,000091 | 2,000021 | -1,000031 | 0,999988 |
| | 6 | 1,000008 | 2,000001 | -1,000003 | 0,999999 |
| | 7 | 1,000001 | 2,000000 | -1,000000 | 1,000000 |
| | 8 | 1,000000 | 2,000000 | -1,000000 | 1,000000 |

# Iterative Techniques for LSE
## SOR algorithm

```
Input:  N; TOL,w
        A:= matrix(n,n);x:= vector(n);b:= vector(n); X0:= vector(n)
Output: x := vector(n);          % approximation of solution Ax=b
                                 % after k Gauss-Seidel steps

k=1
while (k <= N) do
```

for i= 1 to n do $x_i = (1-\omega)XO_j + \frac{\omega}{a_{ii}}\left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right)$

```
    If (|| x- XO || ) < TOL  then STOP (Solution reached)
    k= k + 1
```

for i= 1 to n do $XO_i = x_i$

```
end while
```

# Refresh: Properties of inner products

## Theorem III.37.

For any $x, y, z \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ we have

1. $< x , y > = < y , x >$
2. $< \alpha x , y > = < x , \alpha y > = \alpha < x , y >$
3. $< x + z , y > = < x , y > + < z , y >$
4. $< x , x > \geq 0$
5. $< x , x > = 0 \Leftrightarrow x = 0$

## Example:

$$< x, y > := \sum_{i=1}^{n} x_i \; y_i$$

1. Choose initial approximation $x^{(0)}$
2. Compute: $r^{(0)} = b - A\,x^{(0)}$ and $v^{(1)} = r^{(0)}$
3. For $k = 1,2,\ldots,n$ do

a) $t_k = \frac{<r^{(k-1)},r^{(k-1)}>}{<v^{(k)},A\,v^{(k)}>}$

b) $x^{(k)} = x^{(k-1)} + t_k\,v^{(k)}$ ! Calc. new approximation

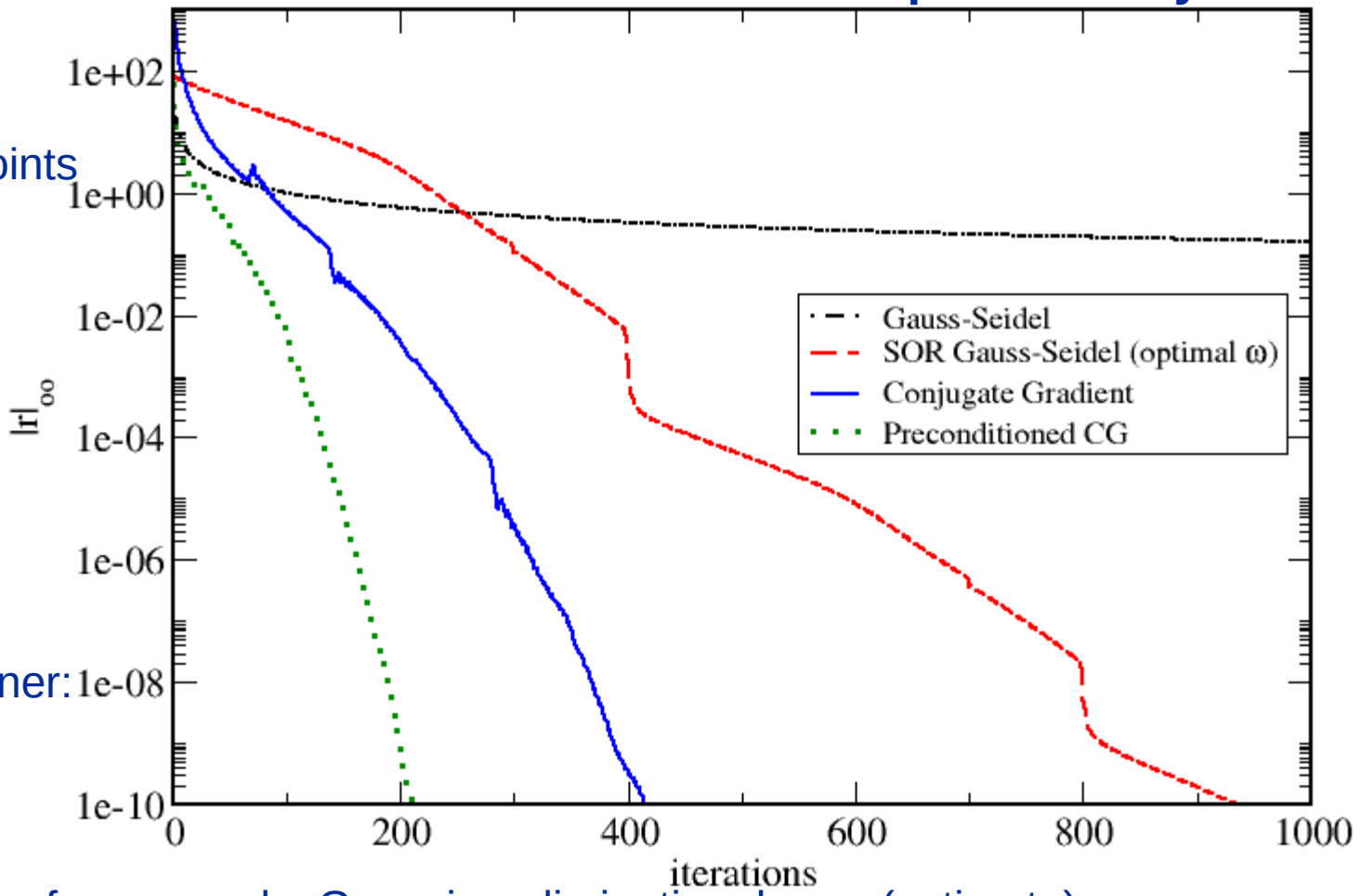
c) $r^{(k)} = r^{(k-1)} - t_k\,A\,v^{(k)}$ ! Calc. current residual

d) $s_k = \frac{<r^{(k)},r^{(k)}>}{<r^{(k-1)},r^{(k-1)}>}$

e) $v^{(k+1)} = r^{(k)} + s_k\,v^{(k)}$ ! Calc. Next search direction

f) Check for convergence of $x^{(k)}$

- **Solve 2D Poisson equation ( $\Delta u(x,y)=f(x,y)$ ) using finite difference method. Coefficient matrix has 5 nonzero elements per row only!**
- 200*100 grid points → n=20.000
- CG Preconditioner: Symmetr. GS




- Compute times ~ few seconds; Gaussian elimination: hours (estimate)

# Lemma (III.17.a.)

**If the spectral radius $\rho(T)$ satisfies $\rho(T) < 1$,**

**then $(I - T)^{-1}$ exists and**

$$(I - T)^{-1} = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j$$

# Lectures: Changes

- **On travel Nov 16th – 22nd → No lectures in that week!**

- **Lecture Nov. 18th → Nov 13th H4 18:00 (s.t.)**
- **Lecture Nov. 21st → Nov 26th H4 18:00 (s.t.)**