Scientific Computing WS 2018/2019

Lecture 10

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

Homework assessment

General

- Please apologize terse answers - on the bright side of this I found time to reply to all individually who handed things in by yesterday noon
- please stick to the filename scheme, this makes it easier for me to give feedback to all of you
- Good style with zip files is that they unpack into subdir with the same name. E.g. `abc.zip` unpacks into directory `abc`.
- Mac users: try to pack your stuff without the __MACOSX and .DS_Store subdirectories
- No need to include binaries
- Always try to calculate errors if exact data is available (I should have been more specific in assignment text)

Code style

- Try to specify datatypes in constants: 0.1f for float, 0.1l for long double and avoid mixing of datatypes in expressions. In particular write x/2.0 instead of x/2 if you do division of a double number. (There are reasonable automatic conversion rules, but things are clearer if they are explicit).
- Cast ints to double explicitely in floating point expressions. This ensures that you don't accidentally create an integer intermediate result. (`1/i*i` was the reason of many overflow errors in your codes)
- Math headers: use `<cmath>` instead of `<math.h>`. In particular, this gives you long double version of functions if needed, in particular for `abs`.
- When using `printf`, use the right format specifiers for output of floating point numbers: %e for `float` and `double`, and %Le for `long double`. %e,%Le give the exponential notation, and `%f`, `%Lf` give a fixed point notation without exponential which is not very helpful for accuracy assessment.

Representation of real numbers

- Any real number $x \in \mathbb{R}$ can be expressed via representation formula:

$$x = \pm \sum_{i=0}^{\infty} d_i \beta^{-i} \beta^e$$

  - $\beta \in \mathbb{N}, \beta \geq 2$: base
  - $d_i \in \mathbb{N}, 0 \leq d_i < \beta$: mantissa digits
  - $e \in \mathbb{Z}$: exponent
- Scientific notation of floating point numbers: e.g. $x = 6.022 \cdot 10^{23}$
  - $\beta = 10$
  - $d = (6, 0, 2, 2, 0 \ldots)$
  - $e = 23$
- Non-unique: $x = 0.6022 \cdot 10^{24}$
  - $\beta = 10$
  - $d = (0, 6, 0, 2, 2, 0 \ldots)$
  - $e = 24$
- Infinite for periodic decimal numbers, irrational numbers

Floating point numbers

- Computer representation uses $\beta = 2$, therefore $d_i \in \{0, 1\}$
- Truncation to fixed finite size

$$x = \pm \sum_{i=0}^{t-1} d_i \beta^{-i} \beta^e$$

  - t: mantissa length
  - Normalization: assume $d_0 = 1 \Rightarrow$ save one bit for mantissa
  - k: exponent size $-\beta^k + 1 = L \leq e \leq U = \beta^k - 1$
  - Extra bit for sign
  - $\Rightarrow$ storage size: $(t-1) + k + 1$
- IEEE 754 single precision (C++ `float`): $k = 8, t = 24 \Rightarrow 32$ bit
- IEEE 754 double precision (C++ `double`): $k = 11, t = 53 \Rightarrow 64$ bit

Floating point limits

Finite size of reprensentation ⇒ there are minimal and maximal possible numbers which can be represented

- symmetry wrt. 0 because of sign bit
- smallest positive normalized number: $d_0 = 1, d_i = 0, i = 1 \dots t - 1$
  $x_{min} = \beta^L$
  - `float`: 1.175494351e–38
  - `double`: 2.2250738585072014e–308
- smallest positive denormalized number: $d_i = 0, i = 0 \dots t - 2, d_{t-1} = 1$
  $x_{min} = \beta^{1-t}\beta^L$
- largest positive normalized number: $d_i = \beta - 1, 0 \dots t - 1$
  $x_{max} = \beta(1 - \beta^{1-t})\beta^U$
  - `float`: 3.402823466e+38
  - `double`: 1.7976931348623158e+308

Machine precision

- There cannot be more than 2^{t+k} floating point numbers $\Rightarrow$ almost all real numbers have to be approximated
- Let x be an exact value and $\tilde{x}$ be its approximation Then: $|\frac{\tilde{x}-x}{x}| < \epsilon$ is the best accuracy estimate we can get, where
  - $\epsilon = \beta^{1-t}$ (truncation)
  - $\epsilon = \frac{1}{2}\beta^{1-t}$ (rounding)
- Also: ϵ is the smallest representable number such that $1 + \epsilon > 1$.
- Relative errors show up in partiular when
  - subtracting two close numbers
  - adding smaller numbers to larger ones

Machine epsilon

- Smallest floating point number ϵ such that $1 + \epsilon > 1$ in floating point arithmetic
- In exact math it is true that from $1 + \varepsilon = 1$ it follows that $0 + \varepsilon = 0$ and vice versa. In floating point computations this is not true
- Many of you used the right algorithm and used the first value or which $1 + \varepsilon = 1$ as the result. This is half the desired quantity.
- Some did not divide start with 1.0 but by other numbers. E.g. 0.1 is not represented exactly in floating point arithmetic
- Recipe for calculation:

  Set $\epsilon = 1.0$;
  while $1.0 + \epsilon/2.0 > 1.0$ **do**
  | $\epsilon = \epsilon/2.0$
  end
- But ... this may be optimized away...

Normalized floating point number

- IEEE 754 32 bit floating point number – normally the same as C++ `float`

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pm$	e_0	e_1	e_1	e_2	e_4	e_5	e_6	e_7	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}	d_{21}	d_{22}	d_{23}

- Storage layout for a normalized number ($d_0 = 1$)
  - bit 0: sign, $0 \to +, \quad 1 \to -$
  - bit $1 \dots 8$: $r = 8$ exponent bits, value $e + 2^{r-1} - 1 = 127$ is stored $\Rightarrow$ no need for sign bit in exponent
  - bit $9 \dots 31$: $t = 23$ mantissa bits $d_1 \dots d_{23}$
  - $d_0 = 1$ not stored $\equiv$ "hidden bit"
- Examples

1	0_01111111_00000000000000000000000	$e = 0$, stored 127
2	0_10000000_00000000000000000000000	$e = 1$, stored 128
0.5	0_01111110_00000000000000000000000	$e = -1$, stored 126
0.1	0_01111011_10011001100110011001101	infinite periodic
0	0_00000000_00000000000000000000000	

- Numbers which are exactly represented in decimal system may not be exactly represented in binary system.

How Additionworks ?

- General:
  - 1. Adjust exponent of number to be added:
    - Until both exponents are equal, add one to exponent, shift mantissa to right by one bit
  - 2. Add both numbers
  - 3. Normalize result
- For $1+\epsilon$, We have at maximum t bit shifts of normalized mantissa until mantissa becomes 0, so $\epsilon = 2^{-t}$.

Data of IEEE 754 floating point representations

	size	t	r	ϵ
`float`	32	23	8	1.1920928955078125e-07
`double`	64	53	11	2.2204460492503131e-16
`long double`	128	63	15	1.0842021724855044e-19

- Floating point format not standardized by language but by IEEE comitee
- Implementation of `long double` varies, may even be the same as `double`, or may be significantly slower, so it is mostly no good option
- There are high accuracy floating point number packages available, which however perform calculations without support of the CPU floating point arithmetic

Summation

- Basel sum: $\sum_{n=1}^{K} \frac{1}{n^2} = \frac{\pi^2}{6}$
- Intended answer for accuracy: sum in reverse order. Start with adding up many small values which would be cancelled out if added to an already large sum value.
- Results for float:

n	forward sum	forward sum error	reverse sum	reverse sum error
10	1.5497677326202392e+00	9.5166447784423828e-02	1.54976773262023925e+00	9.5166447784423828e-02
100	1.6349840164184570e+00	9.95016098022460937e-03	1.63498389720916748e+00	9.95028018951416015e-03
1000	1.6439348459243774e+00	9.99331474304199218e-04	1.64393448829650878e+00	9.99689102172851562e-04
10000	1.6447253227233886e+00	2.08854675292968750e-04	1.64483404159545898e+00	1.00135803222656250e-04
100000	1.6447253227233886e+00	2.08854675292968750e-04	1.64492404460906982e+00	1.01327896118164062e-05
1000000	1.6447253227233886e+00	2.08854675292968750e-04	1.64493298530578613e+00	1.19209289550781250e-06
10000000	1.6447253227233886e+00	2.08854675292968750e-04	1.64493393898010253e+00	2.38418579101562500e-07
100000000	1.6447253227233886e+00	2.08854675292968750e-04	1.64493405818939208e+00	1.19209289550781250e-07

- No gain in accuracy for forward sum for $n > 10000$

Kahan summation

- Some of you hinted at the Kahan compensated summation algorithm (thanks!):

```
T sum_kah=0.0;
T error_compensation=0.0;
for (int i=1; i<=n;i++)
{
  T x=i;
  T increment=1.0/(x*x);
  T corrected_increment=increment-error_compensation;
  T good_sum=sum_kah+corrected_increment;
  error_compensation= (good_sum-sum_kah)-corrected_increment;
  sum_kah =good_sum;
}
```

- When implementing, be careful that expressions are not optimized away ...
- William Kahan (1933-) is the principle architect of the IEEE 754 floating point standard ...

Recap on nonnegative matrices

The Gershgorin Circle Theorem (Semyon Gershgorin,1931)

(everywhere, we assume $n \geq 2$)

Theorem (Varga, Th. 1.11) Let A be an $n \times n$ (real or complex) matrix. Let

$$\Lambda_i = \sum_{\substack{j=1\dots n\\ j\neq i}} |a_{ij}|$$

If λ is an eigenvalue of A then there exists r, $1 \leq r \leq n$ such that

$$|\lambda - a_{rr}| \leq \Lambda_r$$

Proof Assume λ is eigenvalue, $\mathbf{x}$ a corresponding eigenvector, normalized such that $\max_{i=1\dots n} |x_i| = |x_r| = 1$. From $A\mathbf{x} = \lambda\mathbf{x}$ it follows that

$$(\lambda - a_{ii})x_i = \sum_{\substack{j=1\dots n\\ j\neq i}} a_{ij}x_j$$

$$|\lambda - a_{rr}| = |\sum_{\substack{j=1\dots n\\ j\neq r}} a_{rj}x_j| \leq \sum_{\substack{j=1\dots n\\ j\neq r}} |a_{rj}||x_j| \leq \sum_{\substack{j=1\dots n\\ j\neq r}} |a_{rj}| = \Lambda_r$$

$\square$

Gershgorin Circle Corollaries

Corollary: Any eigenvalue of A lies in the union of the disks defined by the Gershgorin circles

$$\lambda \in \bigcup_{i=1\dots n} \{\mu \in \mathbb{V} : |\mu - a_{ii}| \leq \Lambda_i\}$$

Corollary:

$$\rho(A) \leq \max_{i=1\dots n} \sum_{j=1}^{n} |a_{ij}| = ||A||_\infty$$

$$\rho(A) \leq \max_{j=1\dots n} \sum_{i=1}^{n} |a_{ij}| = ||A||_1$$

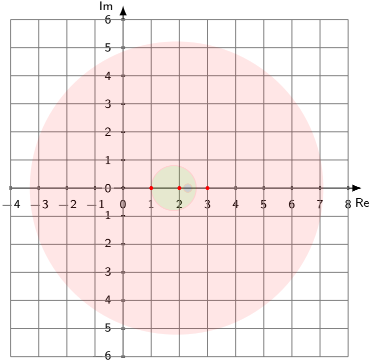
Proof

$$|\mu - a_{ii}| \leq \Lambda_i \quad \Rightarrow \quad |\mu| \leq \Lambda_i + |a_{ii}| = \sum_{j=1}^{n} |a_{ij}|$$

Furthermore, $\sigma(A) = \sigma(A^T)$. $\square$

Gershgorin circles: example

$$A = \begin{pmatrix} 1.9 & 1.8 & 3.4 \\ 0.4 & 1.8 & 0.4 \\ 0.05 & 0.1 & 2.3 \end{pmatrix}, \lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, \Lambda_1 = 5.2, \Lambda_2 = 0.8, \lambda_3 = 0.15$$



Gershgorin circles: heat example I

$$A = \begin{pmatrix} \frac{2}{h} & -\frac{1}{h} & & & & & \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & & \\ & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & \\ & & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} \\ & & & & -\frac{1}{h} & \frac{2}{h} \end{pmatrix}$$

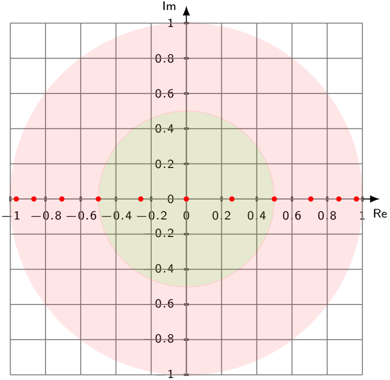
$$B = (I - D^{-1}A) = \begin{pmatrix} 0 & \frac{1}{2} & & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & & \\ & \frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \frac{1}{2} & 0 & \frac{1}{2} & \\ & & & \frac{1}{2} & 0 & \frac{1}{2} \\ & & & & \frac{1}{2} & 0 \end{pmatrix}$$

We have $b_{ii} = 0$, $\Lambda_i = \begin{cases} \frac{1}{2}, & i = 1, n \\ 1 & i = 2 \dots n-1 \end{cases} \Rightarrow$ estimate $|\lambda_i| \leq 1$

Gershgorin circles: heat example II

Let n=11, h=0.1:

$$\lambda_i = \cos\left(\frac{ih\pi}{1+2h}\right) \quad (i = 1 \ldots n)$$



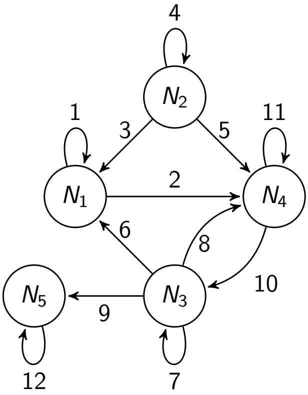
$\Rightarrow$ the Gershgorin circle theorem is too pessimistic...

Weighted directed graph representation of matrices

Define a directed graph from the nonzero entries of a matrix $A = (a_{ik})$:

- Nodes: $\mathcal{N} = \{N_i\}_{i=1\dots n}$
- Directed edges: $\mathcal{E} = \{\overrightarrow{N_k N_l} | a_{kl} \neq 0\}$
- Matrix entries $\equiv$ weights of directed edges

$$A = \begin{pmatrix} 1. & 0. & 0. & 2. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 6. & 0. & 7. & 8. & 9. \\ 0. & 0. & 10. & 11. & 0. \\ 0. & 0. & 0. & 0. & 12. \end{pmatrix}$$



- 1:1 equivalence between matrices and weighted directed graphs
- Convenient e.g. for sparse matrices

Reducible and irreducible matrices

Definition A is *reducible* if there exists a permutation matrix P such that

$$PAP^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

A is *irreducible* if it is not reducible.

Theorem (Varga, Th. 1.17): A is irreducible $\Leftrightarrow$ the matrix graph is connected, i.e. for each *ordered* pair (N_i, N_j) there is a path consisting of directed edges, connecting them.

Equivalently, for each i, j there is a sequence of consecutive nonzero matrix entries $a_{ik_1}, a_{k_1 k_2}, a_{k_2 k_3} \dots, a_{k_{r-1} k_r} a_{k_r j}$.

$\square$

Taussky theorem (Olga Taussky, 1948)

Theorem (Varga, Th. 1.18) Let A be irreducible. Assume that the eigenvalue λ is a boundary point of the union of all the disks

$$\lambda \in \partial \bigcup_{i=1\dots n} \{\mu \in \mathbb{C} : |\mu - a_{ii}| \leq \Lambda_i\}$$

Then, all n Gershgorin circles pass through λ, i.e. for $i = 1 \dots n$,

$$|\lambda - a_{ii}| = \Lambda_i$$

Consequences for heat example from Taussky theorem

- $B = I - D^{-1}A$
- We had $b_{ii} = 0$, $\Lambda_i = \begin{cases} \frac{1}{2}, & i = 1, n \\ 1 & i = 2 \dots n-1 \end{cases} \Rightarrow$ estimate $|\lambda_i| \leq 1$
- Assume $|\lambda_i| = 1$. Then λ_i lies on the boundary of the union of the Gershgorin circles. But then it must lie on the boundary of both circles with radius $\frac{1}{2}$ and 1 around 0.
- Contradiction $\Rightarrow |\lambda_i| < 1$, $\rho(B) < 1$!

Diagonally dominant matrices

Definition Let $A = (a_{ij})$ be an $n \times n$ matrix.

- A is *diagonally dominant* if
  (i) for $i = 1 \ldots n$, $|a_{ii}| \geq \sum_{\substack{j=1\ldots n \\ j\neq i}} |a_{ij}|$
- A is *strictly diagonally dominant* (sdd) if
  (i) for $i = 1 \ldots n$, $|a_{ii}| > \sum_{\substack{j=1\ldots n \\ j\neq i}} |a_{ij}|$
- A is *irreducibly diagonally dominant* (idd) if
  (i) A is irreducible
  (ii) A is diagonally dominant –
  for $i = 1 \ldots n$, $|a_{ii}| \geq \sum_{\substack{j=1\ldots n \\ j\neq i}} |a_{ij}|$
  (iii) for at least one r, $1 \leq r \leq n$, $|a_{rr}| > \sum_{\substack{j=1\ldots n \\ j\neq r}} |a_{rj}|$

A very practical nonsingularity criterion

Theorem (Varga, Th. 1.21): Let A be strictly diagonally dominant or irreducibly diagonally dominant. Then A is nonsingular.

If in addition, $a_{ii} > 0$ is real for $i = 1 \dots n$, then all real parts of the eigenvalues of A are positive:

$$\mathrm{Re}\lambda_i > 0, \quad i = 1 \dots n$$

Corollary

Theorem: If A is complex hermitian or real symmetric, sdd or idd, with positive diagonal entries, it is positive definite.

Proof: All eigenvalues of A are real, and due to the nonsingularity criterion, they must be positive, so A is positive definite.

$\square$

Perron-Frobenius Theorem (1912/1907)

Definition: A real n-vector $\mathbf{x}$ is

- positive ($\mathbf{x} > 0$) if all entries of $\mathbf{x}$ are positive
- nonnegative ($\mathbf{x} \geq 0$) if all entries of $\mathbf{x}$ are nonnegative

Definition: A real $n \times n$ matrix A is

- positive ($A > 0$) if all entries of A are positive
- nonnegative ($A \geq 0$) if all entries of A are nonnegative

Theorem(Varga, Th. 2.7) Let $A \geq 0$ be an irreducible $n \times n$ matrix. Then

(i) A has a positive real eigenvalue equal to its spectral radius $\rho(A)$.
(ii) To $\rho(A)$ there corresponds a positive eigenvector $\mathbf{x} > 0$.
(iii) $\rho(A)$ increases when any entry of A increases.
(iv) $\rho(A)$ is a simple eigenvalue of A.

Proof: See Varga. $\square$

Perron-Frobenius for general nonnegative matrices

Each $n \times n$ matrix can be brought to the normal form

$$PAP^T = \begin{pmatrix} R_{11} & R_{12} & \dots & R_{1m} \\ 0 & R_{22} & \dots & R_{2m} \\ \vdots & & \ddots & \\ 0 & 0 & \dots & R_{mm} \end{pmatrix}$$

where for $j = 1 \dots m$, either R_{jj} irreducible or $R_{jj} = (0)$.

Theorem(Varga, Th. 2.20) Let $A \geq 0$ be an $n \times n$ matrix. Then

(i) A has a nonnegative eigenvalue equal to its spectral radius $\rho(A)$. This eigenvalue is positive unless A is reducible and its normal form is strictly upper triangular

(ii) To $\rho(A)$ there corresponds a nonzero eigenvector $\mathbf{x} \geq 0$.

(iii) $\rho(A)$ does not decrease when any entry of A increases.

Proof: See Varga; $\sigma(A) = \bigcup_{j=1}^{m} \sigma(R_{jj})$, apply irreducible Perron-Frobenius to R_{jj}. $\square$

Jacobi method convergence

Corollary: Let A be sdd or idd, and D its diagonal. Assume that $a_{ii} > 0$ and $a_{ij} \leq 0$ for $i \neq j$. Then $\rho(I - D^{-1}A) < 1$, i.e. the Jacobi method converges.

Proof In this case, $|B| = B$ $\square$.

Regular splittings

- $A = M - N$ is a regular splitting if
  - M is nonsingular
  - M^{-1}, N are nonnegative, i.e. have nonnegative entries
- Regard the iteration $u_{k+1} = M^{-1}Nu_k + M^{-1}b$.
- We have $I - M^{-1}A = M^{-1}N$.

Convergence theorem for regular splitting

Theorem: Assume A is nonsingular, $A^{-1} \geq 0$, and $A = M - N$ is a regular splitting. Then $\rho(M^{-1}N) < 1$.

Proof: Let $G = M^{-1}N$. Then $A = M(I - G)$, therefore $I - G$ is nonsingular.

In addition

$$A^{-1}N = (M(I - M^{-1}N))^{-1}N = (I - M^{-1}N)^{-1}M^{-1}N = (I - G)^{-1}G$$

By Perron-Frobenius (for general matrices), $\rho(G)$ is an eigenvalue with a nonnegative eigenvector $\mathbf{x}$. Thus,

$$0 \leq A^{-1}N\mathbf{x} = \frac{\rho(G)}{1 - \rho(G)}\mathbf{x}$$

Therefore $0 \leq \rho(G) \leq 1$.
As $I - G$ is nonsingular, $\rho(G) < 1$. $\square$

Convergence rate comparison

Corollary: $\rho(M^{-1}N) = \frac{\tau}{1+\tau}$ where $\tau = \rho(A^{-1}N)$.

Proof: Rearrange $\tau = \frac{\rho(G)}{1-\rho(G)}$ $\square$

Corollary: Let $A \geq 0$, $A = M_1 - N_1$ and $A = M_2 - N_2$ be regular splittings. If $N_2 \geq N_1 \geq 0$, then $1 > \rho(M_2^{-1}N_2) \geq \rho(M_1^{-1}N_1)$.

Proof: $\tau_2 = \rho(A^{-1}N_2) \geq \rho(A^{-1}N_1) = \tau_1$

But $\frac{\tau}{1+\tau}$ is strictly increasing. $\square$

M-Matrix definition

Definition Let A be an $n \times n$ real matrix. A is called M-Matrix if

(i) $a_{ij} \leq 0$ for $i \neq j$

(ii) A is nonsingular

(iii) $A^{-1} \geq 0$

Corollary: If A is an M-Matrix, then $A^{-1} > 0 \Leftrightarrow A$ is irreducible.

Proof: See Varga. □

Main practical M-Matrix criterion

Corollary: Let A be sdd or idd. Assume that $a_{ii} > 0$ and $a_{ij} \leq 0$ for $i \neq j$. Then A is an M-Matrix.

Proof: We know that A is nonsingular, but we have to show $A^{-1} \geq 0$.

- Let $B = I - D^{-1}A$. Then $\rho(B) < 1$, therefore $I - B$ is nonsingular.
- We have for $k > 0$:

$$
\begin{aligned}
I - B^{k+1} &= (I - B)(I + B + B^2 + \dots + B^k) \\
(I - B)^{-1}(I - B^{k+1}) &= (I + B + B^2 + \dots + B^k)
\end{aligned}
$$

The left hand side for $k \to \infty$ converges to $(I - B)^{-1}$, therefore

$$
(I - B)^{-1} = \sum_{k=0}^{\infty} B^k
$$

As $B \geq 0$, we have $(I - B)^{-1} = A^{-1}D \geq 0$. As $D > 0$ we must have $A^{-1} \geq 0$. $\square$

Application

Let A be an M-Matrix. Assume $A = D - E - F$.

- Jacobi method: $M = D$ is nonsingular, $M^{-1} \geq 0$. $N = E + F$ nonnegative $\Rightarrow$ convergence
- Gauss-Seidel: $M = D - E$ is an M-Matrix as $A \leq M$ and M has non-positive off-digonal entries. $N = F \geq 0$. $\Rightarrow$ convergence
- Comparison: $N_J \geq N_{GS} \Rightarrow$ Gauss-Seidel converges faster.
- More general: Block Jacobi, Block Gauss Seidel etc.

Intermediate Summary

- Given some matrix, we now have some nice recipies to establish nonsingularity and iterative method convergence:
- **Check if the matrix is irreducible.**
  This is mostly the case for elliptic and parabolic PDEs.
- **Check if the matrix is strictly or irreducibly diagonally dominant**.
  If yes, it is in addition nonsingular.
- **Check if main diagonal entries are positive and off-diagonal entries are nonpositive.**
  If yes, in addition, the matrix is an M-Matrix, its inverse is nonnegative, and elementary iterative methods converge.
- These critera do not depend on the symmetry of the matrix!

Incomplete LU factorizations (ILU)

Idea (Varga, Buleev, 1960):

- fix a predefined zero pattern
- apply the standard LU factorization method, but calculate only those elements, which do not correspond to the given zero pattern
- Result: incomplete LU factors L, U, remainder R:

$$A = LU - R$$

- Problem: with complete LU factorization procedure, for any nonsingular matrix, the method is stable, i.e. zero pivots never occur. Is this true for the incomplete LU Factorization as well ?

Comparison of M-Matrices

Theorem(Saad, Th. 1.33): Let A, B $n \times n$ matrices such that

(i) $A \leq B$

(ii) $b_{ij} \leq 0$ for $i \neq j$.

Then, if A is an M-Matrix, so is B.

Proof: For the diagonal parts, one has $D_B \geq D_A > 0$, $D_A - A \geq D_B - B \geq 0$ Therefore

$$I - D_A^{-1}A \geq D_A^{-1}(D_B - B) \geq D_B^{-1}(D_B - B) = I - D_B^{-1}B =: G \geq 0.$$

Perron-Frobenius $\Rightarrow \rho(G) = \rho(I - D_B^{-1}B) \leq \rho(I - D_A^{-1}A) < 1$
$\Rightarrow I - G$ is nonsingular. From the proof of the M-matrix criterion, $D_B^{-1}B = (I-G)^{-1} = \sum_{k=0}^{\infty} G^k \geq 0$. As $D_B > 0$, we get $B \geq 0$.

$\square$

M-Property propagation in Gaussian Elimination

Theorem:(Ky Fan; Saad Th 1.10) Let A be an M-matrix. Then the matrix A_1 obtained from the first step of Gaussian elimination is an M-matrix.

Proof: One has $a^1_{ij} = a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}}$,
$a_{ij}, a_{i1}, a_{1j} \leq 0$, $a_{11} > 0$
$\Rightarrow a^1_{ij} \leq 0$ for $i \neq j$

$$A = L_1 A_1 \text{ with } L_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \frac{-a_{12}}{a_{11}} & 1 & \dots & 0 \\ \vdots & & \ddots & 0 \\ \frac{-a_{1n}}{a_{11}} & 0 & \dots & 1 \end{pmatrix} \text{ nonsingular, nonnegative}$$

$\Rightarrow A_1$ nonsingular

Let $e_1 \dots e_n$ be the unit vectors. Then $A_1^{-1} e_1 = \frac{1}{a_11} e_1 \geq 0$. For $j > 1$,
$A_1^{-1} e_j = A^{-1} L^{-1} e_j = A^{-1} e_j \geq 0$.
$\Rightarrow A_1^{-1} \geq 0$

$\square$

Stability of ILU

Theorem (Saad, Th. 10.2): If A is an M-Matrix, then the algorithm to compute the incomplete LU factorization with a given nonzero pattern

$$A = LU - R$$

is stable. Moreover, $A = LU - R$ is a regular splitting.

Stability of ILU decomposition II

Proof

Let $\tilde{A}_1 = A_1 + R_1 = L_1 A + R_1$ where R_1 is a nonnegative matrix which occurs from dropping some off diagonal entries from A_1. Thus, $\tilde{A}_1 \geq A_1$ and $\tilde{A}_1$ is an M-matrix. We can repeat this recursively

$$\begin{aligned}\tilde{A}_k = A_k + R_k &= L_k A_{k-1} + R_k \\ &= L_k L_{k-1} A_{k-2} + L_k R_{k-1} + R_k \\ &= L_k L_{k-1} \cdot \ldots \cdot L_1 A + L_k L_{k-1} \cdot \ldots \cdot L_2 R_1 + \cdots + R_k\end{aligned}$$

Let $L = (L_{n-1} \cdot \ldots \cdot L_1)^{-1}$, $U = \tilde{A}_{n-1}$. Then $U = L^{-1}A + S$ with

$$S = L_{n-1}L_{n-2} \cdot \ldots \cdot L_2 R_1 + \cdots + R_{n-1} = L_{n-1}L_{n-2} \cdot \ldots \cdot L_2 (R_1 + R_2 + \ldots R_{n-1})$$

Let $R = R_1 + R_2 + \ldots R_{n-1}$, then $A = LU - R$ where $U^{-1}L^{-1}$, R are nonnegative.

$\square$

ILU(0)

- Special case of ILU: ignore any fill-in.
- Representation:

$$M = (\tilde{D} - E)\tilde{D}^{-1}(\tilde{D} - F)$$

- $\tilde{D}$ is a diagonal matrix (wich can be stored in one vector) which is calculated by the incomplete factorization algorithm.

- Setup:

```
for(int i=0;i<n;i++)
d(i)=a(i,i)

for(int i=0;i<n;i++)
{
  d(i)=1.0/d(i)
  for (int j=i+1;j<n;j++)
  d(j)=d(j)-a(i,j)*d(i)*a(j,i)
}
```

ILU(0)

Solve $Mu = v$

```
for(int i=0;i<n;i++)
{
  double x=0.0;
  for (int j=0;j<i;i++)
  x=x+a(i,j)*u(j)
  u(i)=d(i)*(v(i)-x)

}

for(int i=n-1;i>=0;i--)
{
  double x=0.0
  for(int j=i+1;j<n;j++)
  x=x+a(i,j)*u(j)
  u(i)=u(i)-d(i)*x
}
```

ILU(0)

- Generally better convergence properties than Jacobi, Gauss-Seidel
- One can develop block variants
- Alternatives:
  - ILUM: ("modified"): add ignored off-diagonal entries to $\tilde{D}$
  - ILUT: zero pattern calculated dynamically based on drop tolerance
- Dependence on ordering
- Can be parallelized using graph coloring
- Not much theory: experiment for particular systems
- I recommend it as the default initial guess for a sensible preconditioner
- Incomplete Cholesky: symmetric variant of ILU