Scientific Computing WS 2017/2018

Lecture 9

Jürgen Fuhrmann

juergen.fuhrmann@wias-berlin.de

Numcxx with CodeBlocks

- CodeBlocks support has been added to numcxx-build:
  - `numcxx-build --codeblocks hello.cxx` creates a subdirectory `hello.codeblocks` which contains the codeblocks project file `hello.cbp`
  - Configure and then start codeblocks:
    ```
    $ numcxx-build --codeblocks hello.cxx
    $ codeblocks hello.codeblocks/hello.cbp
    ```
  - Or start codeblocks immediately after configuring
    ```
    $ numcxx-build --codeblocks --execute hello.cxx
    ```
  - In Codeblocks, instead of "all" select target "hello" or "hello/fast", then Build & Run as usual.

Homework assessment

General

- Please apologize terse answers - on the bright side of this I found time to reply to all individually
- 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.
- `Infinity` is a special floating point number which marks the result of an overflow in an operation. In no way it can be used like ∞.
- `NaN` is a special floating point number which marks the result e.g. of a division by zero
- Use type aliases instead of #define:

```
using double as real;
```

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

Floating point representation

- Scientific notation of floating point numbers: e.g. $x = 6.022 \cdot 10^{23}$
- Representation formula:

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

  - $\beta \in \mathbb{N}, \beta \geq 2$: base
  - $d_i \in \mathbb{N}, 0 \leq d_i \leq \beta$: mantissa digits
  - $e \in \mathbb{Z}$: exponent
- Representation on computer:

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

  - $\beta = 2$
  - t: mantissa length, e.g. $t = 53$ for IEEE double
  - $L \leq e \leq U$, e.g. $-1022 \leq e \leq 1023$ (10 bits) for IEEE double
  - $d_0 \neq 0 \Rightarrow$ normalized numbers, unique representation

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	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\pm$	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	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 Addititition $1+\epsilon$ works ?

- 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

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
- `long double` in gcc on x86_64 uses 79 of 128 bits (based on 80 bit internal arithmetic)
- Information in header `<limits>`: `std::numeric_limits`
- Still more to the picture:
  - Optimization not always guaranteed to give the same result
  - Internal precision of calculations in may be larger than memory size $\Rightarrow$ register operations have increased accuracy

Basel sum code

-
- $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$
- Intended answer: 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	1.5497677326202392 5e+00	9.5166447784423828e-02
100	1.6349840164184570e+00	9.9501609802246093 7e-03	1.6349838972091674 8e+00	9.9502801895141601 5e-03
1000	1.6439348459243774e+00	9.9933147430419921 8e-04	1.6439344882965087 8e+00	9.9968910217285156 2e-04
10000	1.6447253227233886e+00	2.0885467529296875 0e-04	1.6448340415954589 8e+00	1.0013580322265625 0e-04
100000	1.6447253227233886e+00	2.0885467529296875 0e-04	1.6449240446090698 2e+00	1.0132789611816406 2e-05
1000000	1.6447253227233886e+00	2.0885467529296875 0e-04	1.6449329853057861 3e+00	1.1920928955078125 0e-06
10000000	1.6447253227233886e+00	2.0885467529296875 0e-04	1.6449339389801025 3e+00	2.3841857910156250 0e-07
100000000	1.6447253227233886e+00	2.0885467529296875 0e-04	1.6449340581893920 8e+00	1.1920928955078125 0e-07

- No gain in accuracy for forward sum for $n > 10000$
- `long double` mostly not a good option

Recap from last time

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

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

Corollary:

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

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

Proof

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

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

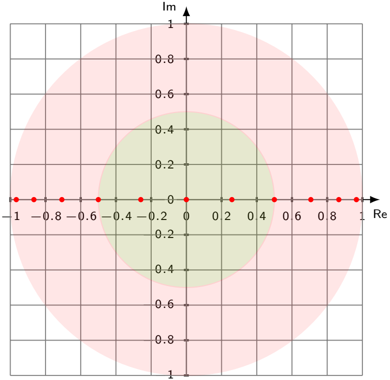
Gershgorin circles: heat example I

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

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

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



n=11, h=0.1

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

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.

Directed matrix graph:

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

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 nonzero matrix entries $a_{ik_1}, a_{k_1 k_2}, \dots, a_{k_r j}$.

$\square$

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

$B = I - D^{-1}A$

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

- A is *diagonally dominant* if

  (i) for $i = 1 \dots n$, $|a_{ii}| \geq \sum_{\substack{j=1\dots n\\ j\neq i}} |a_{ij}|$

- A is *strictly diagonally dominant* (sdd) if

  (i) for $i = 1 \dots n$, $|a_{ii}| > \sum_{\substack{j=1\dots n\\ j\neq i}} |a_{ij}|$

- A is *irreducibly diagonally dominant* (idd) if

  (i) A is irreducible

  (ii) A is diagonally dominant –
  for $i = 1 \dots n$, $|a_{ii}| \geq \sum_{\substack{j=1\dots n\\ j\neq i}} |a_{ij}|$

  (iii) for at least one r, $1 \leq r \leq n$, $|a_{rr}| > \sum_{\substack{j=1\dots n\\ j\neq r}} |a_{rj}|$

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

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

Heat conduction matrix

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

- A is idd $\Rightarrow$ A is nonsingular
- $\mathrm{diag} A$ is positive real $\Rightarrow$ eigenvalues of A have positive real parts
- A is real, symmetric $\Rightarrow$ A is positive definite

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. $\square$

Theorem on Jacobi matrix

Theorem: Let A be sdd or idd, and D its diagonal. Then

$$\rho(|I - D^{-1}A|) < 1$$

Proof: Let $B = (b_{ij}) = I - D^{-1}A$. Then

$$b_{ij} = \begin{cases} 0, & i = j \\ -\frac{a_{ij}}{a_{ii}}, & i \neq j \end{cases}$$

If A is sdd, then for $i = 1 \dots n$,

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

Therefore, $\rho(|B|) < 1$.

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, $\rho(G)$ is an eigvenalue 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: $\tau_2 = \rho(A^{-1}N_2) \geq \rho(A^{-1}N_1) = \tau_1$, $\frac{\tau}{1+\tau}$ is strictly increasing.

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:

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

$$I - B^{k+1} = (I - B)(I + B + B^2 + \cdots + B^k)$$
$$(I - B)^{-1}(I - B^{k+1}) = (I + B + B^2 + \cdots + B^k)$$

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.

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 ?

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.

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--)
{
   doubl 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

Preconditioners

- Leave this topic for a while now
- Hopefully, we well be able to discuss
  - Multigrid: gives $O(n)$ complexity in optimal situations
  - Domain decomposition: Structurally well suited for large scale parallelization

More general iteration schemes

Generalization of iteration schemes

- Simple iterations converge slowly
- For most practical purposes, Krylov subspace methods are used.
- We will introduce one special case and give hints on practically useful more general cases
- Material after J. Shewchuk: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain“

Solution of SPD system as a minimization procedure

Regard $Au = f$,where A is symmetric, positive definite. Then it defines a bilinear form $a : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$

$$a(u, v) = (Au, v) = v^T Au = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} v_i u_j$$

As A is SPD, for all $u \neq 0$ we have $(Au, u) > 0$.

For a given vector b, regard the function

$$f(u) = \frac{1}{2} a(u, u) - b^T u$$

What is the minimizer of f ?

$$f'(u) = Au - b = 0$$

- Solution of SPD system $\equiv$ minimization of f.

Method of steepest descent

- Given some vector u_i, look for a new iterate u_{i+1}.
- The direction of steepest descend is given by $-f'(u_i)$.
- So look for u_{i+1} in the direction of $-f'(u_i) = r_i = b - Au_i$ such that it minimizes f in this direction, i.e. set $u_{i+1} = u_i + \alpha r_i$ with α choosen from

$$
\begin{aligned}
0 &= \frac{d}{d\alpha} f(u_i + \alpha r_i) = f'(u_i + \alpha r_i) \cdot r_i \\
&= (b - A(u_i + \alpha r_i), r_i) \\
&= (b - Au_i, r_i) - \alpha(Ar_i, r_i) \\
&= (r_i, r_i) - \alpha(Ar_i, r_i) \\
\alpha &= \frac{(r_i, r_i)}{(Ar_i, r_i)}
\end{aligned}
$$

Method of steepest descent: iteration scheme

$$
\begin{aligned}
r_i &= b - Au_i \\
\alpha_i &= \frac{(r_i, r_i)}{(Ar_i, r_i)} \\
u_{i+1} &= u_i + \alpha_i r_i
\end{aligned}
$$

Let $\hat{u}$ the exact solution. Define $e_i = u_i - \hat{u}$, then $r_i = -Ae_i$

Let $||u||_A = (Au, u)^{\frac{1}{2}}$ be the *energy norm* wrt. A.

Theorem The convergence rate of the method is

$$||e_i||_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^i ||e_0||_A$$

where $\kappa = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ is the spectral condition number.

Method of steepest descent: advantages

- Simple Richardson iteration $u_{k+1} = u_k - \alpha(Au_k - f)$ needs good eigenvalue estimate to be optimal with $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}}$
- In this case, asymptotic convergence rate is $\rho = \frac{\kappa-1}{\kappa+1}$
- Steepest descent has the same rate without need for spectral estimate