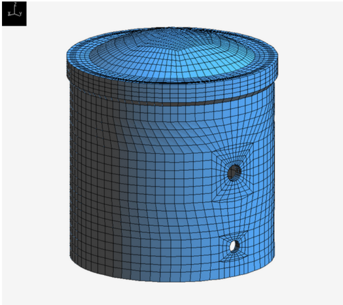
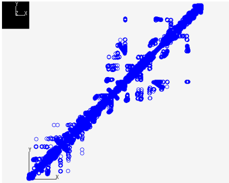
Úvod do metod řešení řídkých soustav lineárních rovnic

- Hledáme řešení
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
- kde počet rovnic je velký (10^6 či větší) a matice $\boldsymbol{A}$ je *řídká*



Ochranná obálka JE

- Počet uzlů: 14970
- Počet prvků: 11764
- Počet rovnic: 43875
- Profil: 67 600 240, po přečíslování 5 866 165



Přímé metody

- Idea: faktorizace (rozklad) matice na součin matic, které jsou snadněji invertovatelné (trojúhelníkové) s možnou permutací pro dosažení stability
- Příklad: LU Dekompozice $\boldsymbol{A} = \boldsymbol{LU}$, kde $\boldsymbol{L}$ a $\boldsymbol{U}$ jsou dolní, resp. horní trojúhelníkové matice. Pokud je rozklad k dispozici, řešení je pak

$$\boldsymbol{Ax} = (\boldsymbol{LU})\boldsymbol{x} = \boldsymbol{L}(\boldsymbol{Ux}) = b,$$

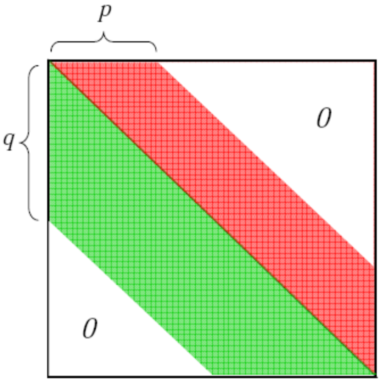
$$\boldsymbol{Ly} = \boldsymbol{b}, \quad \boldsymbol{Ux} = \boldsymbol{y}$$

- Výhoda rozkladu spočívá ve snadném řešení obou podproblémů (dopředná a zpětná substituce).
- Výpočetní komplexita pro LU dekompozici $\mathcal{O}(n^3)$ a $\mathcal{O}(np^2)$ pro řídkou pásovou matici o šířce pásu p.
- Výhody
  - Garantovaný počet oparací
  - Schopnost řešit velké 2D a 3D problémy
  - Rychlost, robustní
- Nevýhody
  - Potřeba sestavit matici soustavy - může znamenat značné

Metody ukládání řídkých matic

- Pásová matice

General, Banded Coefficient Matrix



$$\left.\begin{array}{l} j > i + p \\ \\ i > j + q \end{array}\right\} a_{ij} = 0$$

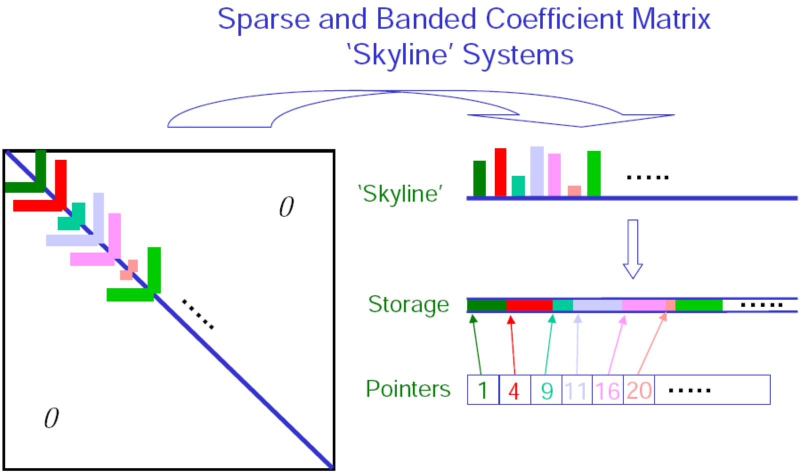
p super-diagonals
q sub-diagonals
$w = p+q+1$ bandwidth

Banded Symmetric Matrix

$$\begin{aligned} a_{ij} &= a_{ji}\,, \quad |i-j| \le b \\ a_{ij} &= a_{ji} = 0\,, \quad |i-j| > b \end{aligned}$$

b is half-bandwidth

- Skyline



- Souřadnicové ukládání (vhodné jen pro iterační řešiče)

Iterační řešiče

- Dva hlavní typy iteračních algoritmů: relaxační (Jacobi, Gauss-Saidel,..) a projekční (Krylovovy metody: CG, GMRES,...)
- Idea: generovat posloupnost aproximací řešení $x_0, x_1, \cdot, x_n$ tak, aby lim $x_n \to x^*$, kde x^* je přesné řešení.
- Na rozdíl od přímých řešičů můžeme řešení ukončit "předčasně" na základě vhodného kritéria, např. $\|\boldsymbol{Ax}_n - \boldsymbol{b}\| \leq \varepsilon$.
- Výhody
  - Nemusí vyžadovat explicitní sestavení matice soustavy
  - (Velmi) Nízké pamětové nároky
  - Efektivní pro velmi řídké systémy, zejména ve 3D
- Nevýhody
  - Mohou vyžadovat velký počet iterací
  - Často nutné efektivní předpodmínění
  - Snadná paralelizace

Předpodnínění

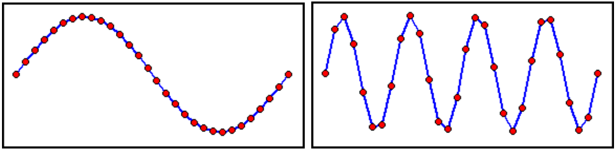
- Konveargence iteračního řešení závisí na čísle podmíněnosti matice soustavy
- Myšlenka předpodmínění je vylepšit číslo podmíněnosti. Předpokládejme, že $\boldsymbol{M}$ je matice, která aproximuje matici soustavy $\boldsymbol{A}$ jež je snadno invertovatelná. Potom můžeme řešit problém $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ nepřímo pomocí

$$\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}$$

- Ideální předpodmínění je $M = A$, ale to je jen teoretická možnost
- Dostupné metody: Diagonální předpomínění, nekompletní faktorizace, ...
- Efektivita závisí na problému, nároky na paměť se mohou výrazně lišit
- Efektivní paralelizace může být problém

Hybridní metody

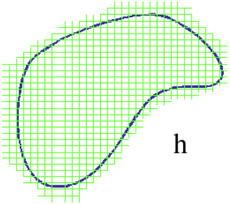
- Multigridní metody
  - Idea: řešením “hrubého” problému dostat lepší počáteční odhad pro řešení problému.
  - Iterační metody mají rychlou konvergenci pro vysokofrekvenční chybu - rychle redukují lokální chyby v řešení, ale velmi pomalu redukují globální (nízkofrekvenční) chybu.
  - Příklad: Rešení Laplaceovy rovnice v 1D, nulové okrajové podmínky, dva různé počáteční stavy (nízko a vysokofrekvenční).

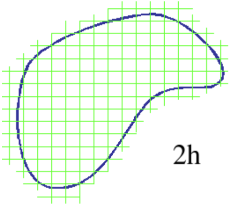
    

    Řešení: (kritérium konvergence: residuum $\varepsilon \leq 10^{-3}$). Vysokofrkvenční problém: 30 iterací, nízkofrekvenční: 433 iterací - tj. 14-ti násobný rozdíl!

Multigrid: Algoritmus

- Problém:$\boldsymbol{Ax} = \boldsymbol{b}$, chyba řešení $\boldsymbol{e} = \boldsymbol{x} - \tilde{\boldsymbol{x}}$. Definujme reziduál $\boldsymbol{r}$: $\boldsymbol{Ae} = \boldsymbol{b} - \boldsymbol{a}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{r}}$





$$
\begin{array}{ccc}
\boldsymbol{A}^h\boldsymbol{e}^h & = & \boldsymbol{b} - \boldsymbol{A}^h\tilde{\boldsymbol{x}}^h \\
 & = & \boldsymbol{r}^h \\
\uparrow & & \downarrow \\
\boldsymbol{A}^{2h}\boldsymbol{e}^{2h} & = & \boldsymbol{r}^{2h}
\end{array}
$$

- Základní kameny: interpolační operátory (restrikce a prolongace)

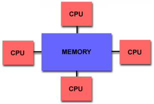
Paralelní řešení soustav rovnic

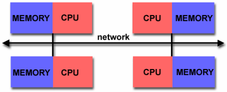
- Velikost řešeného problému na jednom počítači (procesoru) je a bude vždy omezena (doba řešení, dostupná paměť)
- Cestou jsou paralelní, distribuované výpočty využívající nejen potenciál moderních paralelních počítačů, ale i např. počítačových svazků (clusters).
- Flynnova taxomomie:

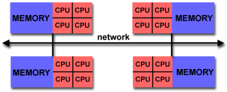
SISD	**SIMD**
MISD	**MIMD**

SI/MI - Single/Multiple Instruction,
SD/MD - Single/Multiple Data

- Nejčastější architektury:
  - Sdílená paměť (Shared memory)
  - Distribuovaná paměť (homogenní a nehomogenní)
  - hybridní systémy



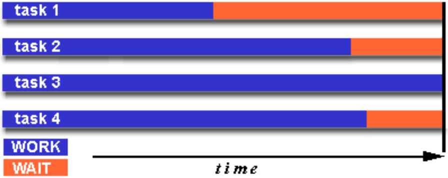




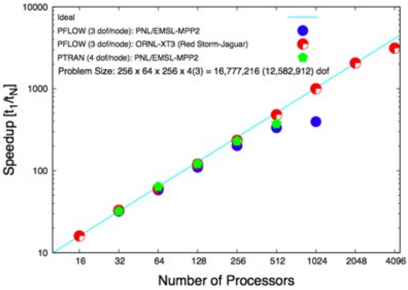
- Programovací modely
  - Vlákna (threads) - sdílená paměť (POSIX, OpenMP)
  - **Message passing** - distribuované i sdílené systémy (MPI)
  - Paralelní datový model - sdílená paměť (F90, HPF)

- Princip: rozdělení problému na podproblémy, které mohou být řešeny na individuálních uzlech, vzájemná závislost vynucuje vzájemnou komunikaci.
- v MKP se používá doménová dekompozice: rozdělení řešené oblasti na subdomény. Pro efektivní zpracování nutný paralelní, distribuovaný solver.
- Požadavky na dekompozici: rovnoměrná distribuce práce (počet prvků), minimální rozhranní mezi subdoménami (komunikace)

- Load Ballancing - distribuce práce mezi uzly (statická, dynamická). Důležitá pro dosažení efektivity.



- Efektivita - často vyjádřená pomocí škálovatelnosti algoritmu



Doporučená literatura:

1. Z. Bittnar, J. Šejnoha:*Numerické metody mechaniky I*, ES ČVUT, 1992.
2. J. Šejnoha, J. Bittnarová:*Pružnost a pevnost 10*, ES ČVUT, 1997.
3. http://www.mgnet.org/mgnet-tuts.html