
Řešení radiální soustavy rovníc

© 1996-2008 Josef Pelikán
KSVI MFF UK Praha

e-mail: Josef.Pelikan@mff.cuni.cz

WWW: <http://cgg.ms.mff.cuni.cz/~pepca/>

Soustava lineárních rovnic

$$\underline{B_i} - \rho_i \cdot \sum_{j=1}^N \underline{B_j} F_{ij} = E_i \quad i = 1..N$$

$$\begin{bmatrix} 1 - \rho_1 F_{1,1} & -\rho_1 F_{1,2} & \dots & -\rho_1 F_{1,N} \\ -\rho_2 F_{2,1} & 1 - \rho_2 F_{2,2} & \dots & -\rho_2 F_{2,N} \\ \dots & \dots & \dots & \dots \\ -\rho_N F_{N,1} & -\rho_N F_{N,2} & \dots & 1 - \rho_N F_{N,N} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \dots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \dots \\ E_N \end{bmatrix}$$

vektor neznámých $[B_i]$

Veličiny

- ➔ B_i .. neznámé **radiosity** jednotlivých plošek
 - při barevném výpočtu je třeba spočítat radiosity pro všechny požadované vlnové délky (barevné složky - např. **R,G,B**)
- ➔ E_i .. vlastní (emitované) **radiosity** (**R,G,B**)
- ➔ ρ_i .. **faktory odrazivosti** materiálu (**R,G,B**)
- ➔ F_{ij} .. **konfigurační faktory**
 - závislé pouze na geometrii scény

Vlastnosti matice soustavy \mathbf{M}

- ➔ matice \mathbf{M} je poměrně **řidká** pro složitější scény
- ➔ \mathbf{M} je **diagonálně dominantní** a dobře podmíněná
 - lze ji úspěšně řešit iteračními metodami (Jacobi, Gauss-Seidel)

$$\sum_{\substack{j=1 \\ j \neq i}}^N \rho_i F_{ij} \leq 1 - \rho_i F_{ii}$$

Gauss-Seidelova metoda

Maticový tvar soustavy:

$$\underline{\mathbf{M} \cdot \mathbf{B} = \mathbf{E}} \quad \mathbf{M} = \left[\mathbf{M}_{ij} \right]_{i,j=1}^N$$

První odhad: $\mathbf{B}_i^{(0)} = \mathbf{E}_i$

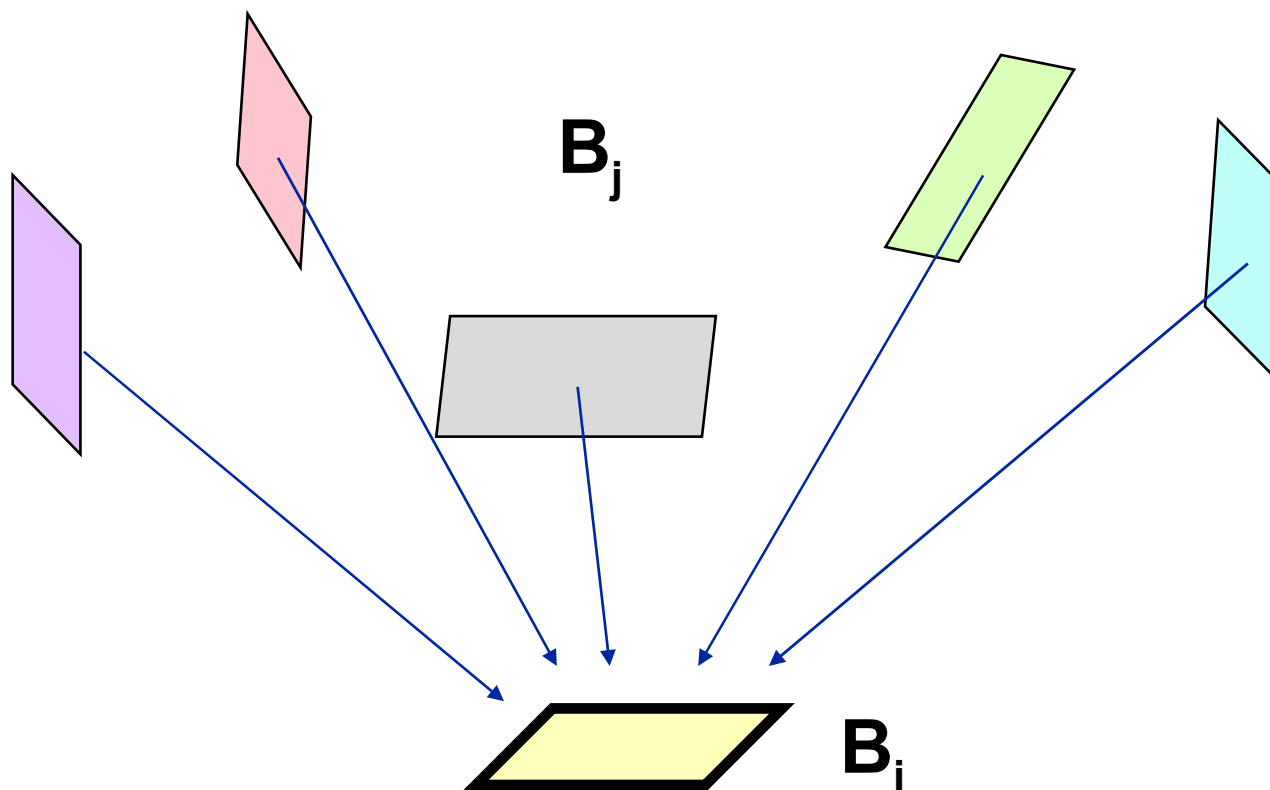
Krok:

$$\mathbf{B}_i^{(k+1)} = \frac{\mathbf{E}_i}{\mathbf{M}_{ii}} - \sum_{j=1}^{i-1} \frac{\mathbf{M}_{ij}}{\mathbf{M}_{ii}} \mathbf{B}_j^{(k+1)} - \sum_{j=i+1}^N \frac{\mathbf{M}_{ij}}{\mathbf{M}_{ii}} \mathbf{B}_j^{(k)}$$

Výpočet
v praxi:

$$\mathbf{B}_i = \mathbf{E}_i + \rho_i \cdot \sum_{j=1}^N \mathbf{B}_j \mathbf{F}_{ij} \quad i = 1..N$$

Fyzikální interpretace (sbírání)



$$\mathbf{B}_i = \mathbf{E}_i + \underbrace{\rho_i}_{\text{charge density}} \cdot \sum_{j \neq i} \underbrace{\mathbf{B}_j F_{ij}}_{\text{field contribution}}$$

Reziduum

Reziduum (odhad chyby) k-té iterace:

$$\mathbf{r}^{(k)} = \mathbf{E} - \mathbf{M} \cdot \mathbf{B}^{(k)}$$

V jednom kroku výpočtu se aktualizuje jedna složka vektoru řešení \mathbf{B}_i :

$$\mathbf{B}_i^{(k+1)} = \mathbf{B}_i^{(k)} + \frac{\mathbf{r}_i^{(k)}}{\mathbf{M}_{ii}}$$

(Jacobiho metoda .. rezidua se opravují po dokončení iterace, Gauss-Seidel .. oprava po každém kroku)

Southwellova iterační metoda

- ◆ Jacobiho i Gauss-Seidelova metoda v každém kroku výpočtu vynulují jednu složku rezidua (na úkor ostatních!)
 - složky se aktualizují v pořadí **1, 2, ... N**
- Southwellova metoda vybírá k aktualizaci vždy složku s **největší absolutní hodnotou rezidua**
- složky s velkou chybou se opravují **častěji**
 - rychlejší konvergence vektoru řešení

Southwellova iterační metoda

- 1 výběr složky s maximálním reziduem:

$$|r_i| = \max_j \{ |r_j| \}$$

- 2 aktualizace i -té složky řešení \mathbf{B}_i
- 3 aktualizace vektoru reziduí \mathbf{r}
- 4 kroky 1 až 3 se opakují, dokud soustava nesplňuje konvergenční kritérium

Inkrementální výpočet rezidua

Aktualizace vektoru řešení v jednom kroku výpočtu:

$$\mathbf{B}^{(p+1)} = \mathbf{B}^{(p)} + \Delta \mathbf{B}^{(p)}$$

Oprava rezidua:

$$\underline{\mathbf{r}^{(p+1)}} = \mathbf{E} - \mathbf{M} \cdot \left(\mathbf{B}^{(p)} + \Delta \mathbf{B}^{(p)} \right) = \underline{\mathbf{r}^{(p)} - \mathbf{M} \cdot \Delta \mathbf{B}^{(p)}}$$

Protože se změnila pouze i -tá složka vektoru řešení:

$$\mathbf{r}_j^{(p+1)} = \mathbf{r}_j^{(p)} - \mathbf{M}_{ji} \cdot \frac{\mathbf{r}_i^{(p)}}{\mathbf{M}_{ii}} \quad \mathbf{j} = 1..N$$

Southwellův algoritmus

```
double B[N], E[N], r[N], M[N][N];

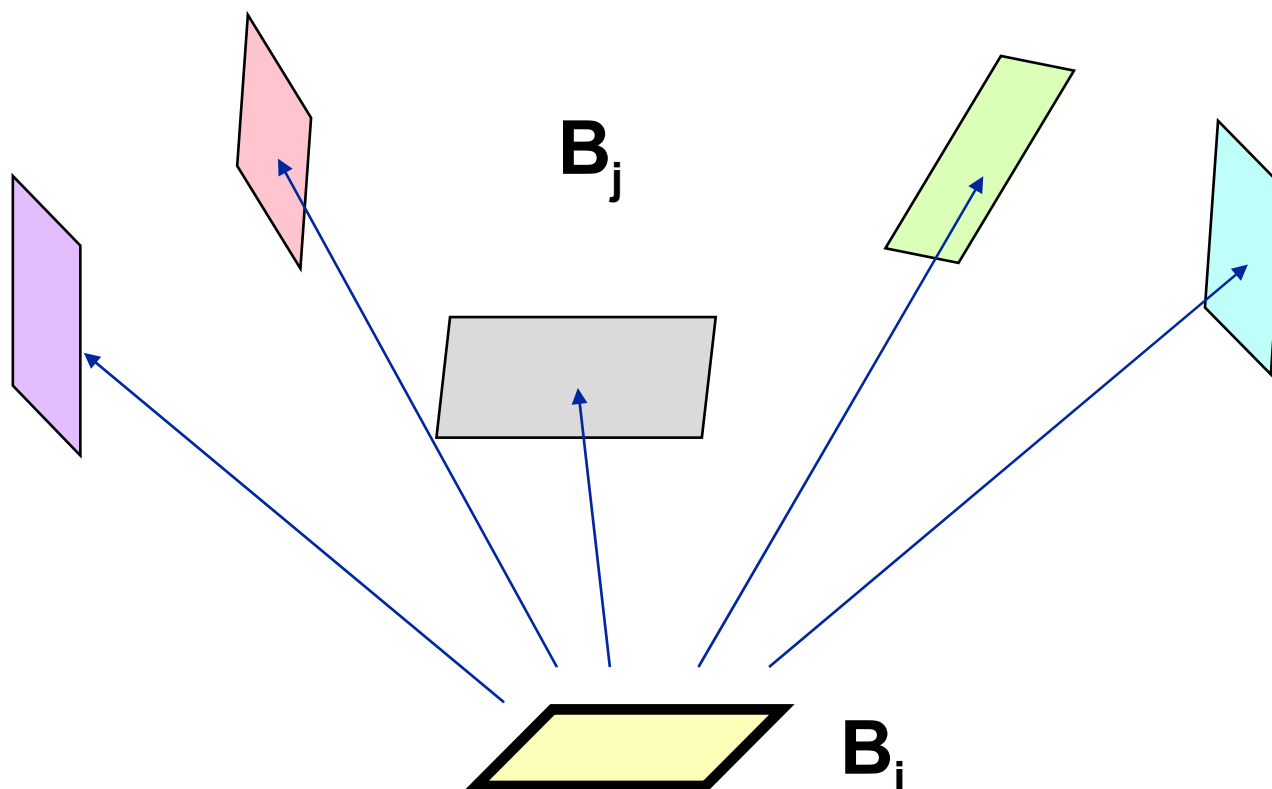
    // inicializace řešení a rezidua
for ( int i=0; i<N; i++ ) {
    B[i] := 0.0;
    r[i] := E[i];
}
while ( "nezkonvergovalo" ) {
    // jeden krok výpočtu:
    "výběr i tak, aby fabs(r[i])== max(fabs(r[i]))"
    double delta = r[i]/M[i][i];
    B[i] += delta;
    for ( int j=0; j<N; j++ )
        r[j] -= M[j][i]*delta;
}
```

Fyzikální interpretace (střílení)

- ➔ B_i .. radiosita i -té plošky (přímá i nepřímá)
- ➔ **jeden krok výpočtu** .. rozdělení (výstřel) radiosity z i -té plošky do okolí
- ➔ r_i .. dosud **nevystřelená radiosita** i -té plošky
- ➔ **konvergence metody** .. celková nevystřelená energie ve scéně se zmenšuje

Fyzikální interpretace (střílení)

$$\mathbf{B}_j^{(p+1)} = \mathbf{B}_j^{(p)} + \underline{\mathbf{r}_i^{(p)} \cdot \rho_j \cdot \mathbf{F}_{ji}}$$



Celková nevystřelená energie

Podle recipročního pravidla pro konfigurační faktory:

$$\underline{r_j^{(p+1)}} = r_j^{(p)} + \rho_j \cdot F_{ji} \cdot r_i^{(p)} = r_j^{(p)} + \rho_j \cdot F_{ij} \frac{A_i}{A_j} \cdot r_i^{(p)}$$

Distribuce energie v jednom kroku výpočtu:

$$r_i^{(p+1)} \cdot A_i = 0$$
$$r_j^{(p+1)} \cdot A_j = r_j^{(p)} \cdot A_j + \underbrace{\rho_j}_{<1} \cdot \underbrace{F_{ij}} \cdot r_i^{(p)} \cdot A_i \quad j = 1..N$$

Progresivní radiační metoda

- ◆ M. Cohen a spol., SIGGRAPH '88
- ◆ interaktivní výpočet osvětlení
 - po každém kroku se nakreslí průběžný výsledek
 - snaha dobře odhadnout řešení již v několika prvních krocích
- ➔ modifikace Southwellovy metody
 - výběr plošky s největší dosud nevystřelenou energií
 - použití okolní složky osvětlení

Progresivní radiační metoda

```
double B[N], E[N], dB[N], F[N][N], A[N], ro[N];

for ( int i=0; i<N; i++ ) { // inicializace B, dB
    B[i] := E[i];
    dB[i] := E[i];
}
while ( "nezkonvergovalo" ) { // jeden krok výpočtu
    "výběr i tak, aby dB[i]*A[i]== max(dB[i]*A[i])"
    for ( int j=0; j<N; j++ ) {
        double dRad = dB[i]*ro[j]*F[j][i];
        B[j] += dRad;
        dB[j] += dRad;
    }
    dB[i] = 0.0;
    "zobrazení mezivýsledku pomocí radiosit B[i]"
}
```


Okolní složka (“ambient term”)

- ◆ vylepšení vzhledu průběžně kreslených mezivýsledků
- ➔ aproximace dosud nespočítaných odrazů světla

Celková dosud nevystřelená radiosita:

$$\overline{\Delta \mathbf{B}} = \frac{\sum r_i \cdot \mathbf{A}_i}{\sum \mathbf{A}_i}$$

Okolní složka

Průměrný koeficient odrazu: $\bar{\rho} = \frac{\sum \rho_i \cdot A_i}{\sum A_i}$

Odhad zbytkové (okolní) radiosity:

$$\underline{\mathbf{B}_{\text{amb}}} = \overline{\Delta \mathbf{B}} \cdot \left(\mathbf{1} + \bar{\rho} + \bar{\rho}^2 + \dots \right) = \underline{\frac{\overline{\Delta \mathbf{B}}}{\mathbf{1} - \bar{\rho}}}$$

Pro zobrazení se radiosity každé plošky upraví:

$$\mathbf{B}_i^{\text{disp}} = \mathbf{B}_i + \rho_i \cdot \mathbf{B}_{\text{amb}}$$

Hyper-relaxace

- ◆ **urychlení konvergence** iterační metody (Jacobi, Gauss-Seidel, progresivní radiální metoda)
- ➔ při aktualizaci rozdělím/seberu o trochu větší množství energie
 - předpovídám budoucí vývoj konvergence
 - pozor na příliš velký koeficient hyper-relaxace (metoda pak už nemusí konvergovat)!
 - je nutné počítat i se záporným reziduem!

Hyper-relaxace

Krok výpočtu s hyper-relaxací:

$$\mathbf{B}_i^{(k+1)} = \mathbf{B}_i^{(k)} + \omega \cdot \frac{\mathbf{r}_i^{(k)}}{\mathbf{M}_{ii}}$$

Hyper-relaxační koeficient: $\omega \geq 1$ (např. 1.2)

Příslušná složka rezidua se již nenuluje, ale bude mít hodnotu:

$$\mathbf{r}_i^{(k+1)} = (1 - \omega) \cdot \mathbf{r}_i^{(k)}$$

“Přestřelování” (“overshooting”)

- ◆ M. Feda, W. Purgathofer, 1992
- ➔ při hyper-relaxaci beru v úvahu množství dosud nevystřelené energie
 - v prvních fázích výpočtu přestřelují více, později již méně
 - jistější konvergence

Literatura

- **M. Cohen, J. Wallace:** *Radiosity and Realistic Image Synthesis*, Academic Press, 1993, 109-130 (chyby!)
- **M. Cohen, S. E. Chen, J. R. Wallace, D. P. Greenberg:** *A progressive refinement approach to fast radiosity image generation*, SIGGRAPH '88, 75-84

Konec

Další informace:

- **A. Glassner: *Principles of Digital Image Synthesis*, Morgan Kaufmann, 1995, 900-916**
- **J. Foley, A. van Dam, S. Feiner, J. Hughes: *Computer Graphics, Principles and Practice*, 800-803**
- **M. Feda, W. Purgathofer: *Accelerating radiosity by overshooting*, The Third EG Workshop on Rendering, Bristol, 1992, 21-32**