Lecture 8
Structured AMR for elliptic problems

Course *Block-structured Adaptive Finite Volume Methods in C++*

Ralf Deiterding  
University of Southampton  
Engineering and the Environment  
Highfield Campus, Southampton SO17 1BJ, UK  
E-mail: r.deiterding@soton.ac.uk
Outline

Adaptive geometric multigrid methods
  Linear iterative methods for Poisson-type problems
  Multi-level algorithms
  Multigrid algorithms on SAMR data structures
  Example
  Comments on parabolic problems
Outline

Adaptive geometric multigrid methods
  Linear iterative methods for Poisson-type problems
  Multi-level algorithms
  Multigrid algorithms on SAMR data structures
  Example
  Comments on parabolic problems
Poisson equation

\[ \Delta q(x) = \psi(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad q \in C^2(\Omega), \quad \psi \in C^0(\Omega) \]

\[ q = \psi^\Gamma(x), \quad x \in \partial\Omega \]
Poisson equation

\[ \Delta q(x) = \psi(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad q \in C^2(\Omega), \quad \psi \in C^0(\Omega) \]
\[ q = \psi^\Gamma(x), \quad x \in \partial\Omega \]

Discrete Poisson equation in 2D:

\[ \frac{Q_{j+1,k} - 2Q_{jk} + Q_{j-1,k}}{\Delta x_1^2} + \frac{Q_{j,k+1} - 2Q_{jk} + Q_{j,k-1}}{\Delta x_2^2} = \psi_{jk} \]
Poisson equation

\[ \Delta q(x) = \psi(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad q \in C^2(\Omega), \quad \psi \in C^0(\Omega) \]

\[ q = \psi^\Gamma(x), \quad x \in \partial\Omega \]

Discrete Poisson equation in 2D:

\[
\frac{Q_{j+1,k} - 2Q_{jk} + Q_{j-1,k}}{\Delta x_1^2} + \frac{Q_{j,k+1} - 2Q_{jk} + Q_{j,k-1}}{\Delta x_2^2} = \psi_{jk}
\]

Operator

\[
A(Q_{\Delta x_1, \Delta x_2}) = \begin{bmatrix}
\frac{1}{\Delta x_1^2} & -\left(\frac{2}{\Delta x_1^2} + \frac{2}{\Delta x_2^2}\right) & \frac{1}{\Delta x_2^2} \\
\frac{1}{\Delta x_1^2} & \frac{1}{\Delta x_2^2} & \frac{1}{\Delta x_2^2}
\end{bmatrix} Q(x_{1,j}, x_{2,k}) = \psi_{jk}
\]
Poisson equation

\[ \Delta q(x) = \psi(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad q \in C^2(\Omega), \quad \psi \in C^0(\Omega) \]

\[ q = \psi^\Gamma(x), \quad x \in \partial\Omega \]

Discrete Poisson equation in 2D:

\[ \frac{Q_{j+1,k} - 2Q_{jk} + Q_{j-1,k}}{\Delta x_1^2} + \frac{Q_{j,k+1} - 2Q_{jk} + Q_{j,k-1}}{\Delta x_2^2} = \psi_{jk} \]

Operator

\[ A(Q_{\Delta x_1, \Delta x_2}) = \begin{bmatrix} \frac{1}{\Delta x_1^2} & - \left( \frac{2}{\Delta x_1^2} + \frac{2}{\Delta x_2^2} \right) & \frac{1}{\Delta x_2^2} \\ \frac{1}{\Delta x_2^2} & \frac{1}{\Delta x_2^2} \end{bmatrix} Q(x_{1,j}, x_{2,k}) = \psi_{jk} \]

\[ Q_{jk} = \frac{1}{\sigma} \left[ (Q_{j+1,k} + Q_{j-1,k})\Delta x_2^2 + (Q_{j,k+1} + Q_{j,k-1})\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

with \( \sigma = \frac{2\Delta x_1^2 + 2\Delta x_2^2}{\Delta x_1^2 \Delta x_2^2} \)
Iterative methods

Jacobi iteration

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m)\Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m)\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]
Iterative methods

Jacobi iteration

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Lexicographical Gauss-Seidel iteration (use updated values when they become available)

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^{m+1}) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^{m+1}) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]
Iterative methods

Jacobi iteration

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Lexicographical Gauss-Seidel iteration (use updated values when they become available)

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^{m+1}) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^{m+1}) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Efficient parallelization / patch-wise application not possible!
Iterative methods

Jacobi iteration

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Lexicographical Gauss-Seidel iteration (use updated values when they become available)

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^{m+1}) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^{m+1}) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Efficient parallelization / patch-wise application not possible!

Checker-board or Red-Black Gauss Seidel iteration

1. \[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m) \Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \] for \( j + k \mod 2 = 0 \)

2. \[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^{m+1} + Q_{j-1,k}^m) \Delta x_2^2 + (Q_{j,k+1}^{m+1} + Q_{j,k-1}^{m+1}) \Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \] for \( j + k \mod 2 = 1 \)
Iterative methods

Jacobi iteration

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m)\Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m)\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Lexicographical Gauss-Seidel iteration (use updated values when they become available)

\[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^{m+1})\Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^{m+1})\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]

Efficient parallelization / patch-wise application not possible!

Checker-board or Red-Black Gauss Seidel iteration

1. \[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^m + Q_{j-1,k}^m)\Delta x_2^2 + (Q_{j,k+1}^m + Q_{j,k-1}^m)\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]
   for \( j + k \mod 2 = 0 \)

2. \[ Q_{jk}^{m+1} = \frac{1}{\sigma} \left[ (Q_{j+1,k}^{m+1} + Q_{j-1,k}^{m+1})\Delta x_2^2 + (Q_{j,k+1}^{m+1} + Q_{j,k-1}^{m+1})\Delta x_1^2 - \Delta x_1^2 \Delta x_2^2 \psi_{jk} \right] \]
   for \( j + k \mod 2 = 1 \)

Gauss-Seidel methods require \( \sim 1/2 \) of iterations than Jacobi method, however, iteration count still proportional to number of unknowns [Hackbusch, 1994]
Smoothing vs. solving

\[ Q^{m+\nu} = S(Q^m, \psi, \nu) \]
Smoothing vs. solving

$\nu$ iterations with iterative linear solver

$$Q^{m+\nu} = S(Q^m, \psi, \nu)$$

Defect after $m$ iterations

$$d^m = \psi - A(Q^m)$$
Smoothing vs. solving

\( \nu \) iterations with iterative linear solver

\[ Q^{m+\nu} = S(Q^m, \psi, \nu) \]

Defect after \( m \) iterations

\[ d^m = \psi - A(Q^m) \]

Defect after \( m + \nu \) iterations

\[ d^{m+\nu} = \psi - A(Q^{m+\nu}) = \psi - A(Q^m + v^m_\nu) = d^m - A(v^m_\nu) \]

with correction

\[ v^m_\nu = S(\bar{0}, d^m, \nu) \]
Smoothing vs. solving

$\nu$ iterations with iterative linear solver

$$Q^{m+\nu} = S(Q^m, \psi, \nu)$$

Defect after $m$ iterations

$$d^m = \psi - A(Q^m)$$

Defect after $m + \nu$ iterations

$$d^{m+\nu} = \psi - A(Q^{m+\nu}) = \psi - A(Q^m + \nu^m) = d^m - A(\nu^m)$$

with correction

$$\nu^m = S(\vec{0}, d^m, \nu)$$

Neglecting the sub-iterations in the smoother we write

$$Q^{n+1} = Q^n + \nu = Q^n + S(d^n)$$
Smoothing vs. solving

\( \nu \) iterations with iterative linear solver

\[ Q^{m+\nu} = S(Q^m, \psi, \nu) \]

Defect after \( m \) iterations

\[ d^m = \psi - A(Q^m) \]

Defect after \( m + \nu \) iterations

\[ d^{m+\nu} = \psi - A(Q^{m+\nu}) = \psi - A(Q^m + v^m_\nu) = d^m - A(v^m_\nu) \]

with correction

\[ v^m_\nu = S(\tilde{0}, d^m, \nu) \]

Neglecting the sub-iterations in the smoother we write

\[ Q^{n+1} = Q^n + \nu = Q^n + S(d^n) \]

Observation: Oscillations are damped faster on coarser grid.
Smoothing vs. solving

ν iterations with iterative linear solver

\[ Q^{m+\nu} = S(Q^m, \psi, \nu) \]

Defect after m iterations

\[ d^m = \psi - A(Q^m) \]

Defect after m + ν iterations

\[ d^{m+\nu} = \psi - A(Q^{m+\nu}) = \psi - A(Q^m + v^m) = d^m - A(v^m) \]

with correction

\[ v^m = S(\vec{0}, d^m, \nu) \]

Neglecting the sub-iterations in the smoother we write

\[ Q^{n+1} = Q^n + \nu = Q^n + S(d^n) \]

Observation: Oscillations are damped faster on coarser grid.

Coarse grid correction:

\[ Q^{n+1} = Q^n + \nu = Q^n + P\mathcal{R}(d^n) \]

where \( \mathcal{R} \) is suitable restriction operator and \( P \) a suitable prolongation operator.
Two-grid correction method

Relaxation on current grid:

$$\bar{Q} = S(Q^n, \psi, \nu)$$

$$Q^{n+1} = \bar{Q} + PS(\vec{0}, \cdot, \mu)R(\psi - A(\bar{Q}))$$
Two-grid correction method

Relaxation on current grid:

\[ \tilde{Q} = S(Q^n, \psi, \nu) \]
\[ Q^{n+1} = \tilde{Q} + PS(\vec{0}, \cdot, \mu)R(\psi - A(\tilde{Q})) \]

Algorithm:

\[ \tilde{Q} := S(Q^n, \psi, \nu) \]
\[ d := \psi - A(\tilde{Q}) \]
\[ d_c := R(d) \]
\[ v_c := S(0, d_c, \mu) \]
\[ v := P(v_c) \]
\[ Q^{n+1} := \tilde{Q} + v \]
Two-grid correction method

Relaxation on current grid:

\[ \tilde{Q} = S(Q^n, \psi, \nu) \]
\[ Q^{n+1} = \tilde{Q} + \mathcal{P}S(\tilde{0}, \cdot, \mu)R(\psi - A(\tilde{Q})) \]

Algorithm:

\[
\begin{align*}
\tilde{Q} &:= S(Q^n, \psi, \nu) \\
d &:= \psi - A(Q) \\
d_c &:= R(d) \\
v_c &:= S(0, d_c, \mu) \\
v &:= \mathcal{P}(v_c) \\
Q^{n+1} &:= \tilde{Q} + v
\end{align*}
\]

with smoothing:

\[
\begin{align*}
v_c &:= S(0, d_c, \mu) \\
v &:= v + \mathcal{P}(v_c) \\
Q^{n+1} &:= Q + v
\end{align*}
\]
Two-grid correction method

Relaxation on current grid:

$$\bar{Q} = S(Q^n, \psi, \nu)$$

$$Q^{n+1} = \bar{Q} + \mathcal{P}S(\tilde{0}, \cdot, \mu)\mathcal{R}(\psi - A(\bar{Q}))$$

Algorithm: with smoothing: with pre- and post-iteration:

$$\bar{Q} := S(Q^n, \psi, \nu)$$ $$d := \psi - A(Q)$$ $$d := \psi - A(Q)$$
$$d := \psi - A(\bar{Q})$$ $$v := S(0, d, \nu)$$ $$v := S(0, d, \nu_1)$$
$$d_c := \mathcal{R}(d)$$ $$r := d - A(v)$$ $$r := d - A(v)$$
$$v_c := S(0, d_c, \mu)$$ $$d_c := \mathcal{R}(r)$$ $$d_c := \mathcal{R}(r)$$
$$v := \mathcal{P}(v_c)$$ $$v_c := S(0, d_c, \mu)$$ $$v_c := S(0, d_c, \mu)$$
$$Q^{n+1} := \bar{Q} + v$$ $$v := v + \mathcal{P}(v_c)$$ $$v := v + \mathcal{P}(v_c)$$

[Hackbusch, 1985]
Multi-level methods and cycles

V-cycle
\[ \gamma = 1 \]
2-grid

\[ S \rightarrow S \rightarrow S \]
Multi-level methods and cycles

V-cycle
\[ \gamma = 1 \]
2-grid

3-grid
Multi-level methods and cycles

V-cycle
\( \gamma = 1 \)
2-grid

3-grid

4-grid

[Hackbusch, 1985] [Wesseling, 1992]...

Structured AMR for elliptic problems
Multi-level methods and cycles

V-cycle
\[ \gamma = 1 \]
2-grid

3-grid

4-grid

W-cycle
\[ \gamma = 2 \]
Multi-level methods and cycles

V-cycle
\( \gamma = 1 \)
2-grid

3-grid

4-grid

W-cycle
\( \gamma = 2 \)

[Hackbusch, 1985] [Wesseling, 1992] ...
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell $j$, $\psi - \nabla \cdot \nabla q = 0$

$$d_j' = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1} - Q_j') - \frac{1}{\Delta x_l} (Q_j' - Q_{j-1}') \right)$$

$H_l$ is an approximation to the derivative of $Q_l$.

Consider a 2-level situation with $r_{l+1} = 2$:

No specific modification necessary for 1D vertex-based stencils, cf. [Bastian, 1996]
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell $j$, $\psi - \nabla \cdot \nabla q = 0$

$$d_j^l = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_l} (Q_j^l - Q_{j-1}^l) \right) = \psi_j - \frac{1}{\Delta x_l} \left( H_{j+\frac{1}{2}}^l - H_{j-\frac{1}{2}}^l \right)$$

$H$ is approximation to derivative of $Q^l$. 

Consider 2-level situation with $r_{l+1} = 2$: 

$$Q_{l}^j - Q_{l}^{j-1} = Q_{l+1}^j - Q_{l+1}^{j-1}$$

Solution needs to be continuously differentiable across interface. Easiest approach:

$$H_{l+\frac{1}{2}}^l = H_{l-\frac{1}{2}}^l$$

No specific modification necessary for 1D vertex-based stencils, cf. [Bastian, 1996]
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell \( j \), \( \psi - \nabla \cdot \nabla q = 0 \)

\[
d^l_j = \psi_j - \frac{1}{\Delta x^l} \left( \frac{1}{\Delta x^l} (Q^l_{j+1} - Q^l_j) - \frac{1}{\Delta x^l} (Q^l_j - Q^l_{j-1}) \right) = \psi_j - \frac{1}{\Delta x^l} \left( H^l_{j+\frac{1}{2}} - H^l_{j-\frac{1}{2}} \right)
\]

\( H \) is approximation to derivative of \( Q^l \).

Consider 2-level situation with \( r_{l+1} = 2 \):

\[
\begin{align*}
Q^l_{w-1} & \quad Q^l_w & \quad Q^l_{w+1} \\
\vdots & \quad \vdots & \quad \vdots \\
Q^l_{j-1} & \quad Q^l_j & \quad Q^l_{j+1}
\end{align*}
\]
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell $j$, $\psi - \nabla \cdot \nabla q = 0$

$$d_j^l = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_l} (Q_j^l - Q_{j-1}^l) \right) = \psi_j - \frac{1}{\Delta x_l} \left( H_{j+\frac{1}{2}}^l - H_{j-\frac{1}{2}}^l \right)$$

$H$ is approximation to derivative of $Q^l$.

Consider 2-level situation with $r_{l+1} = 2$:
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell \( j \), \( \psi - \nabla \cdot \nabla q = 0 \)

\[
d_j = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_l} (Q_j^l - Q_{j-1}^l) \right) = \psi_j - \frac{1}{\Delta x_l} \left( H_{j+\frac{1}{2}}^l - H_{j-\frac{1}{2}}^l \right)
\]

\( H \) is approximation to derivative of \( Q^l \).
Consider 2-level situation with \( r_{l+1} = 2 \):

Solution needs to be continuously differentiable across interface.
Easiest approach: \( H_{w+\frac{1}{2}}^{l+1} \equiv H_{j-\frac{1}{2}}^l \)
Stencil modification at coarse-fine boundaries in 1D

1D Example: Cell $j$, $\psi - \nabla \cdot \nabla q = 0$

$$d_j = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_l} (Q_j^l - Q_{j-1}^l) \right) = \psi_j - \frac{1}{\Delta x_l} \left( H_{j+\frac{1}{2}}^l - H_{j-\frac{1}{2}}^l \right)$$

$H$ is approximation to derivative of $Q^l$.

Consider 2-level situation with $r_{l+1} = 2$:

Solution needs to be continuously differentiable across interface.

Easiest approach: $H_{w+\frac{1}{2}}^{l+1} \equiv H_{j-\frac{1}{2}}^l$

No specific modification necessary for 1D vertex-based stencils, cf. [Bastian, 1996]
Stencil modification at coarse-fine boundaries in 1D II

Set $H_l^{i+1} = H_I$.
Stencils modification at coarse-fine boundaries in 1D II

Set $H_{l+1}^{w+\frac{1}{2}} = H_I$. Inserting $Q$ gives

$$\frac{Q_{w+1}^{l+1} - Q_w^{l+1}}{\Delta x_{l+1}} = \frac{Q_j^l - Q_w^{l+1}}{\frac{3}{2} \Delta x_{l+1}}$$
Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_I$. Inserting $Q$ gives

$$
\frac{Q^{l+1}_{w+1} - Q^{l+1}_w}{\Delta x_{l+1}} = \frac{Q'_j - Q^{l+1}_w}{\frac{3}{2} \Delta x_{l+1}}
$$

from which we readily derive

$$
Q^{l+1}_{w+1} = \frac{2}{3} Q'_j + \frac{1}{3} Q^{l+1}_w
$$

for the boundary cell on $l + 1$. 
Stencil modification at coarse-fine boundaries in 1D II

Set $H_{w+\frac{1}{2}}^{l+1} = H_l$. Inserting $Q$ gives

$$\frac{Q_{w+1}^{l+1} - Q_w^{l+1}}{\Delta x_{l+1}} = \frac{Q_j^l - Q_w^{l+1}}{\frac{3}{2} \Delta x_{l+1}}$$

from which we readily derive

$$Q_{w+1}^{l+1} = \frac{2}{3} Q_j^l + \frac{1}{3} Q_w^{l+1}$$

for the boundary cell on $l+1$. We use the flux correction procedure to enforce $H_{w+\frac{1}{2}}^{l+1} \equiv H_l^{l+1}$ and thereby $H_{j-\frac{1}{2}}^l \equiv H_l$.
Stencil modification at coarse-fine boundaries in 1D II

Set \( H^{l+1}_{w+\frac{1}{2}} = H_I \). Inserting \( Q \) gives

\[
\frac{Q^{l+1}_{w+1} - Q^{l+1}_w}{\Delta x_{l+1}} = \frac{Q^l_j - Q^{l+1}_w}{\frac{3}{2} \Delta x_{l+1}}
\]

from which we readily derive

\[
Q^{l+1}_{w+1} = \frac{2}{3} Q^l_j + \frac{1}{3} Q^{l+1}_w
\]

for the boundary cell on \( l + 1 \). We use the flux correction procedure to enforce

\( H^{l+1}_{w+\frac{1}{2}} \equiv H^l_{j-\frac{1}{2}} \) and thereby \( H^l_{j-\frac{1}{2}} \equiv H_I \).

Correction pass [Martín, 1998]

1. \( \delta H^{l+1}_{j-\frac{1}{2}} := -H^l_{j-\frac{1}{2}} \)
Stencil modification at coarse-fine boundaries in 1D II

Set $H_{w+\frac{1}{2}}^{l+1} = H_I$. Inserting $Q$ gives

$$\frac{Q_{w+1}^{l+1} - Q_w^{l+1}}{\Delta x_{l+1}} = \frac{Q_j^l - Q_w^{l+1}}{\frac{3}{2} \Delta x_{l+1}}$$

from which we readily derive

$$Q_{w+1}^{l+1} = 2 \frac{Q_j^l}{3} + \frac{1}{3} Q_w^{l+1}$$

for the boundary cell on $l + 1$. We use the flux correction procedure to enforce $H_{w+\frac{1}{2}}^{l+1} \equiv H_j^{l-\frac{1}{2}}$ and thereby $H_{j-\frac{1}{2}}^l \equiv H_I$.

Correction pass [Martin, 1998]

1. $\delta H_{j-\frac{1}{2}}^{l+1} := -H_{j-\frac{1}{2}}^l$

2. $\delta H_{j-\frac{1}{2}}^{l+1} := \delta H_{j-\frac{1}{2}}^{l+1} + H_{w+\frac{1}{2}}^{l+1} = -H_{j-\frac{1}{2}}^l + (Q_j^l - Q_w^{l+1})/\frac{3}{2} \Delta x_{l+1}$
Stencil modification at coarse-fine boundaries in 1D II

Set \( H_{w+\frac{1}{2}}^{l+1} = H_{\frac{l}{2}} \). Inserting \( Q \) gives

\[
\frac{Q_{w+1}^{l+1} - Q_{w}^{l+1}}{\Delta x_{l+1}} = \frac{Q_{j}^{l} - Q_{w}^{l+1}}{\frac{3}{2} \Delta x_{l+1}}
\]

from which we readily derive

\[
Q_{w+1}^{l+1} = \frac{2}{3} Q_{j}^{l} + \frac{1}{3} Q_{w}^{l+1}
\]

for the boundary cell on \( l + 1 \). We use the flux correction procedure to enforce \( H_{w+\frac{1}{2}}^{l+1} \equiv H_{j-\frac{1}{2}}^{l} \) and thereby \( H_{j-\frac{1}{2}}^{l} \equiv H_{\frac{l}{2}} \).

Correction pass [Martin, 1998]

1. \( \delta H_{j-\frac{1}{2}}^{l+1} := -H_{j-\frac{1}{2}}^{l} \)

2. \( \delta H_{j-\frac{1}{2}}^{l+1} := \delta H_{j-\frac{1}{2}}^{l+1} + H_{w+\frac{1}{2}}^{l+1} = -H_{j-\frac{1}{2}}^{l} + (Q_{j}^{l} - Q_{w}^{l+1})/\frac{3}{2} \Delta x_{l+1} \)

3. \( \tilde{d}_{j}^{l} := d_{j}^{l} + \frac{1}{\Delta x_{l}} \delta H_{j-\frac{1}{2}}^{l+1} \)
Stencil modification at coarse-fine boundaries in 1D II

Set $H_{w+\frac{1}{2}}^{l+1} = H_I$. Inserting $Q$ gives

$$\frac{Q_{w+1}^{l+1} - Q_w^{l+1}}{\Delta x_{l+1}} = \frac{Q_j^l - Q_w^{l+1}}{\frac{3}{2} \Delta x_{l+1}}$$

from which we readily derive

$$Q_{w+1}^{l+1} = \frac{2}{3} Q_j^l + \frac{1}{3} Q_w^{l+1}$$

for the boundary cell on $l+1$. We use the flux correction procedure to enforce

$H_{w+\frac{1}{2}}^{l+1} \equiv H_{j-\frac{1}{2}}^l$ and thereby $H_{j-\frac{1}{2}}^l \equiv H_I$.

Correction pass [Martin, 1998]

1. $\delta H_{j-\frac{1}{2}}^{l+1} := -H_{j-\frac{1}{2}}^l$

2. $\delta H_{j-\frac{1}{2}}^{l+1} := \delta H_{j-\frac{1}{2}}^{l+1} + H_{w+\frac{1}{2}}^{l+1} = -H_{j-\frac{1}{2}}^l + (Q_j^l - Q_w^{l+1})/\frac{3}{2} \Delta x_{l+1}$

3. $\tilde{d}_j^l := d_j^l + \frac{1}{\Delta x_l} \delta H_{j-\frac{1}{2}}^{l+1}$

yields

$$\tilde{d}_j^l = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{2}{3 \Delta x_{l+1}} (Q_j^l - Q_w^{l+1}) \right)$$
Stencil modification at coarse-fine boundaries: 2D

\[
Q_{v,w}^{l+1} = +
\]

\[
Q_{v,w}^{l+1} = (1 - \frac{2}{r_{l+1} + 1})Q_{v,w}^{l+1} + \frac{2}{r_{l+1} + 1}(1 - f)Q_{j,k}^{l+1} + fQ_{j,k}^{l+1}
\]

with

\[
f = \frac{x_{1,v}^{l+1} - x_{1,j}^{l}}{\Delta x_{1,v}^{l}}
\]
Stencil modification at coarse-fine boundaries: 2D

\[ Q^{l+1}_{v,w-1} = + \]

\[ Q^{l+1}_{vw} \]

\[ Q^{l}_{jk} \]
Stencil modification at coarse-fine boundaries: 2D

\[ Q_{v,w-1}^{l+1} = + \left( \frac{3}{4} Q_{j,k}^l + \frac{1}{4} Q_{j+1,k}^l \right) \]
Stencil modification at coarse-fine boundaries: 2D

\[ Q_{v,w-1}^{l+1} = \frac{1}{3} Q_{v,w}^{l+1} + \frac{2}{3} \left( \frac{3}{4} Q_{j,k}^l + \frac{1}{4} Q_{j+1,k}^l \right) \]
Stochastic modification at coarse-fine boundaries: 2D

\[ Q_{v,w}^{l+1} = \frac{1}{3} Q_{vw}^{l+1} + \]
\[ \frac{2}{3} \left( \frac{3}{4} Q_{jk}^l + \frac{1}{4} Q_{j+1,k}^l \right) \]

In general:

\[ Q_{v,w}^{l+1} = \left( 1 - \frac{2}{r_{l+1} + 1} \right) Q_{vw}^{l+1} + \]
\[ \frac{2}{r_{l+1} + 1} \left( (1 - f) Q_{jk}^l + f Q_{j+1,k}^l \right) \]

with

\[ f = \frac{x_{1,v,l+1}^y - x_{1,l}^j}{\Delta x_{1,l}} \]
Components of an SAMR multigrid method

- Stencil operators
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect $d^l = \psi^l - A(Q^l)$ on each grid $G_{l,m}$ of level $l$
Components of an SAMR multigrid method

▶ Stencil operators

▶ Application of defect $d^l = \psi^l - A(Q^l)$ on each grid $G_{l,m}$ of level $l$
▶ Computation of correction $v^l = S(0, d^l, \nu)$ on each grid of level $l$
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect $d^l = \psi^l - A(Q^l)$ on each grid $G_{l,m}$ of level $l$
  - Computation of correction $v^l = S(0, d^l, \nu)$ on each grid of level $l$
- Boundary (ghost cell) operators
Components of an SAMR multigrid method

- **Stencil operators**
  - Application of defect \( d^l = \psi^l - \mathcal{A}(Q^l) \) on each grid \( G_{l,m} \) of level \( l \)
  - Computation of correction \( v^l = \mathcal{S}(0, d^l, \nu) \) on each grid of level \( l \)

- **Boundary (ghost cell) operators**
  - Synchronization of \( Q^l \) and \( v^l \) on \( \tilde{S}_l^1 \)
Components of an SAMR multigrid method

▶ Stencil operators
  ▶ Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level $l$
  ▶ Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level $l$
▶ Boundary (ghost cell) operators
  ▶ Synchronization of $Q^l$ and $v^l$ on $\tilde{S}^1_l$
  ▶ Specification of Dirichlet boundary conditions for a finite volume discretization for $Q^l \equiv w$ and $v^l \equiv w$ on $\tilde{P}^1_l$
Components of an SAMR multigrid method

- **Stencil operators**
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level $l$
  - Computation of correction $v^l = S(0, d^l, \nu)$ on each grid of level $l$

- **Boundary (ghost cell) operators**
  - Synchronization of $Q^l$ and $v^l$ on $\tilde{S}_l^1$
  - Specification of Dirichlet boundary conditions for a finite volume discretization for $Q^l \equiv w$ and $v^l \equiv w$
    - on $\tilde{P}_l^1$
  - Specification of $v^l \equiv 0$ on $\tilde{I}_l^1$
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect \( d^l = \psi^l - A(Q^l) \) on each grid \( G_{l,m} \) of level \( l \)
  - Computation of correction \( v^l = S(0, d^l, \nu) \) on each grid of level \( l \)

- Boundary (ghost cell) operators
  - Synchronization of \( Q^l \) and \( v^l \) on \( \tilde{S}^1_l \)
  - Specification of Dirichlet boundary conditions for a finite volume discretization for \( Q^l = w \) and \( v^l = w \) on \( \tilde{P}^1_l \)
  - Specification of \( v^l \equiv 0 \) on \( \tilde{I}^1_l \)
  - Specification of \( Q^l = \frac{(r_l - 1)Q^{l+1} + 2Q^l}{r_l + 1} \) on \( \tilde{I}^1_l \)
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect $d^l = \psi^l - A(Q^l)$ on each grid $G_{l,m}$ of level $l$
  - Computation of correction $v^l = S(0, d^l, \nu)$ on each grid of level $l$

- Boundary (ghost cell) operators
  - Synchronization of $Q^l$ and $v^l$ on $\tilde{S}_l^1$
  - Specification of Dirichlet boundary conditions for a finite volume discretization for $Q^l \equiv w$ and $v^l \equiv w$ on $\tilde{P}_l^1$
  - Specification of $v^l \equiv 0$ on $\tilde{T}_l^1$
  - Specification of $Q^l = \frac{(r_l-1)Q^{l+1} + 2Q^l}{r_l+1}$ on $\tilde{T}_l^1$
  - Coarse-fine boundary flux accumulation and application of $\delta H^{l+1}$ on defect $d^l$
Components of an SAMR multigrid method

- **Stencil operators**
  - Application of defect $d^l = \psi^l - A(Q^l)$ on each grid $G_{l,m}$ of level $l$
  - Computation of correction $v^l = S(0, d^l, \nu)$ on each grid of level $l$

- **Boundary (ghost cell) operators**
  - Synchronization of $Q^l$ and $v^l$ on $\tilde{S}_l^1$
  - Specification of Dirichlet boundary conditions for a finite volume discretization for $Q^l \equiv w$ and $v^l \equiv w$ on $\tilde{P}_l^1$
  - Specification of $v^l \equiv 0$ on $\tilde{I}_l^1$
  - Specification of $Q^l = \frac{(r_l-1)Q^{l+1}+2Q^l}{r_l+1}$ on $\tilde{I}_l^1$
  - Coarse-fine boundary flux accumulation and application of $\delta H^{l+1}$ on defect $d^l$
  - Standard prolongation and restriction on grids between adjacent levels

![Diagram of multigrid method components](image-url)
Components of an SAMR multigrid method

- **Stencil operators**
  - Application of defect \( d^l = \psi^l - A(Q^l) \) on each grid \( G_{l,m} \) of level \( l \)
  - Computation of correction \( v^l = S(0, d^l, \nu) \) on each grid of level \( l \)

- **Boundary (ghost cell) operators**
  - Synchronization of \( Q^l \) and \( v^l \) on \( \tilde{S}^1_l \)
  - Specification of Dirichlet boundary conditions for a finite volume discretization for \( Q^l \equiv w \) and \( v^l \equiv w \) on \( \tilde{P}^1_l \)
  - Specification of \( v^l \equiv 0 \) on \( \tilde{I}^1_l \)
  - Specification of \( Q^l = \frac{(r_l-1)Q^{l+1} + 2Q^l}{r_l+1} \) on \( \tilde{I}^1_l \)

- Coarse-fine boundary flux accumulation and application of \( \delta H^{l+1} \) on defect \( d^l \)

- Standard prolongation and restriction on grids between adjacent levels

- **Adaptation criteria**
Components of an SAMR multigrid method

- **Stencil operators**
  - Application of defect \( d^l = \psi^l - \mathcal{A}(Q^l) \) on each grid \( G_{l,m} \) of level \( l \)
  - Computation of correction \( v^l = S(0, d^l, \nu) \) on each grid of level \( l \)

- **Boundary (ghost cell) operators**
  - Synchronization of \( Q^l \) and \( v^l \) on \( \tilde{S}^l \)
  - Specification of Dirichlet boundary conditions for a finite volume discretization for \( Q^l \equiv w \) and \( v^l \equiv w \) on \( \tilde{P}^l \)
  - Specification of \( v^l \equiv 0 \) on \( \tilde{T}^l \)
  - Specification of \( Q_i = \frac{(r_i-1)Q_i^{l+1} + 2Q_i}{r_i+1} \) on \( \tilde{T}^l \)

- **Coarse-fine boundary flux accumulation and application of \( \delta H^{l+1} \) on defect \( d^l \)**

- **Standard prolongation and restriction on grids between adjacent levels**

- **Adaptation criteria**
  - E.g., standard restriction to project solution on 2x coarsened grid, then use local error estimation
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect \( d^l = \psi^l - A(Q^l) \) on each grid \( G_{l,m} \) of level \( l \)
  - Computation of correction \( v^l = S(0, d^l, \nu) \) on each grid of level \( l \)

- Boundary (ghost cell) operators
  - Synchronization of \( Q^l \) and \( v^l \) on \( \tilde{S}^1 \)
  - Specification of Dirichlet boundary conditions for a finite volume discretization for \( Q^l \equiv w \) and \( v^l \equiv w \) on \( \tilde{P}^1 \)
  - Specification of \( v^l \equiv 0 \) on \( \tilde{I}^1 \)
  - Specification of \( Q^l = \frac{(r_l-1)Q^{l+1}+2Q^l}{r_l+1} \) on \( \tilde{I}^1 \)

- Coarse-fine boundary flux accumulation and application of \( \delta H^{l+1} \) on defect \( d^l \)

- Standard prolongation and restriction on grids between adjacent levels

- Adaptation criteria
  - E.g., standard restriction to project solution on 2x coarsened grid, then use local error estimation
  - Looping instead of time steps and check of convergence
Additive geometric multigrid algorithm

AdvanceLevelMG(\(l\)) - Correction Scheme

Set ghost cells of \(Q^l\)
Calculate defect \(d^l\) from \(Q^l, \psi^l\)
\[ d^l := \psi^l - A(Q^l) \]
If \((l < l_{\text{max}})\)
  Calculate updated defect \(r^{l+1}\) from \(v^{l+1}, d^{l+1}\)
  Restrict \(d^{l+1}\) onto \(d^l\)
Do \(\nu_1\) smoothing steps to get correction \(v^l\)
If \((l > l_{\text{min}})\)
  Do \(\gamma > 1\) times
    AdvanceLevelMG(\(l - 1\))
  Set ghost cells of \(v^{l-1}\)
  Prolongate and add \(v^{l-1}\) to \(v^l\)
  If \((\nu_2 > 0)\)
    Set ghost cells of \(v^l\)
    Update defect \(d^l\) according to \(v^l\)
    Do \(\nu_2\) post-smoothing steps to get \(r^l\)
    Add additional correction \(r^l\) to \(v^l\)
  Add correction \(v^l\) to \(Q^l\)
\[ Q^l := Q^l + v^l \]
Additive Geometric Multiplicative Multigrid Algorithm

Start - Start iteration on level $l_{\text{max}}$

For $l = l_{\text{max}}$ Down to $l_{\text{min}} + 1$ Do

Restrict $Q'$ onto $Q'^{-1}$

Regrid(0)

AdvanceLevelMG($l_{\text{max}}$)

See also: [Trottenberg et al., 2001], [Canu and Ritzdorf, 1994]
Vertex-based: [Brandt, 1977], [Briggs et al., 2001]
Example

On $\Omega = [0, 10] \times [0, 10]$ use hat function

$$
\psi = \begin{cases} 
  -A_n \cos\left(\frac{\pi r}{2 R_n}\right), & r < R_n \\
  0 & \text{elsewhere}
\end{cases}
$$

with $r = \sqrt{(x_1 - X_n)^2 + (x_2 - Y_n)^2}$

to define three sources with

<table>
<thead>
<tr>
<th>n</th>
<th>$A_n$</th>
<th>$R_n$</th>
<th>$X_n$</th>
<th>$Y_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3</td>
<td>0.3</td>
<td>6.5</td>
<td>8.0</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.3</td>
<td>2.0</td>
<td>7.0</td>
</tr>
<tr>
<td>3</td>
<td>-0.1</td>
<td>0.4</td>
<td>7.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>
Example

On $\Omega = [0, 10] \times [0, 10]$ use hat function

$$
\psi = \begin{cases} 
-A_n \cos \left( \frac{\pi r}{2R_n} \right), & r < R_n \\
0 & \text{elsewhere}
\end{cases}
$$

with $r = \sqrt{(x_1 - X_n)^2 + (x_2 - Y_n)^2}$

to define three sources with

<table>
<thead>
<tr>
<th>$n$</th>
<th>$A_n$</th>
<th>$R_n$</th>
<th>$X_n$</th>
<th>$Y_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3</td>
<td>0.3</td>
<td>6.5</td>
<td>8.0</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.3</td>
<td>2.0</td>
<td>7.0</td>
</tr>
<tr>
<td>3</td>
<td>-0.1</td>
<td>0.4</td>
<td>7.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>
Example

On $\Omega = [0, 10] \times [0, 10]$ use hat function

$$\psi = \begin{cases} -A_n \cos \left( \frac{\pi r}{2R_n} \right), & r < R_n \\ 0, & \text{elsewhere} \end{cases}$$

with $r = \sqrt{(x_1 - X_n)^2 + (x_2 - Y_n)^2}$

to define three sources with

<table>
<thead>
<tr>
<th>$n$</th>
<th>$A_n$</th>
<th>$R_n$</th>
<th>$X_n$</th>
<th>$Y_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3</td>
<td>0.3</td>
<td>6.5</td>
<td>8.0</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.3</td>
<td>2.0</td>
<td>7.0</td>
</tr>
<tr>
<td>3</td>
<td>-0.1</td>
<td>0.4</td>
<td>7.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$l_{\text{max}}$</th>
<th>$l_{\text{min}}$</th>
<th>$\nu_1$</th>
<th>$\nu_2$</th>
<th>$V$-Cycles</th>
<th>Time [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>-4</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>9.4</td>
</tr>
<tr>
<td>0</td>
<td>-7</td>
<td>5</td>
<td>5</td>
<td>16</td>
<td>27.7</td>
</tr>
<tr>
<td>0</td>
<td>-4</td>
<td>5</td>
<td>5</td>
<td>341</td>
<td>563</td>
</tr>
</tbody>
</table>

Stop at $\|d^l\|_{\text{max}} < 10^{-7}$ for $l \geq 0$, $\gamma = 1$, $r_l = 2$
Some comments on parabolic problems

- Consequences of time step refinement
- Level-wise elliptic solves vs. global solve
- If time step refinement is used an elliptic flux correction is unavoidable.
References I


