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

Outline

Poisson equation

$$
\begin{aligned}
\Delta q(\mathbf{x}) &= \psi(\mathbf{x})\,, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad q \in \mathrm{C}^2(\Omega), \quad \psi \in \mathrm{C}^0(\Omega) \\
q &= \psi^\Gamma(\mathbf{x})\,, \quad \mathbf{x} \in \partial\Omega
\end{aligned}
$$

Poisson equation

$$
\begin{aligned}
\Delta q(\mathbf{x}) &= \psi(\mathbf{x})\,, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad q \in \mathrm{C}^2(\Omega), \quad \psi \in \mathrm{C}^0(\Omega) \\
q &= \psi^\Gamma(\mathbf{x})\,, \quad \mathbf{x} \in \partial\Omega
\end{aligned}
$$

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

$$\begin{aligned} \Delta q(\mathbf{x}) &= \psi(\mathbf{x})\,, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad q \in \mathrm{C}^2(\Omega), \quad \psi \in \mathrm{C}^0(\Omega) \\ q &= \psi^{\Gamma}(\mathbf{x})\,, \quad \mathbf{x} \in \partial\Omega \end{aligned}$$

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

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

Poisson equation

$$\begin{array}{rcl}\Delta q(\mathbf{x}) & = & \psi(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad q \in \mathrm{C}^2(\Omega), \quad \psi \in \mathrm{C}^0(\Omega) \\ q & = & \psi^\Gamma(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega\end{array}$$

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

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

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

with $\sigma = \dfrac{2\Delta x_1^2 + 2\Delta x_2^2}{\Delta x_1^2 \Delta x_2^2}$

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

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

ν iterations with iterative linear solver

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

Smoothing vs. solving

ν iterations with iterative linear solver

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

Defect after m iterations

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

Smoothing vs. solving

ν iterations with iterative linear solver

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

Defect after m iterations

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

Defect after $m+\nu$ iterations

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

with correction

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

Smoothing vs. solving

ν iterations with iterative linear solver

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

Defect after m iterations

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

Defect after $m+\nu$ iterations

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

with correction

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

Neglecting the sub-iterations in the smoother we write

$$Q^{n+1} = Q^n + v = Q^n + \mathcal{S}(d^n)$$

Smoothing vs. solving

ν iterations with iterative linear solver

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

Defect after m iterations

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

Defect after $m+\nu$ iterations

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

with correction

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

Neglecting the sub-iterations in the smoother we write

$$Q^{n+1} = Q^n + v = Q^n + \mathcal{S}(d^n)$$

Observation: Oscillations are damped faster on coarser grid.

Smoothing vs. solving

ν iterations with iterative linear solver

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

Defect after m iterations

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

Defect after $m + \nu$ iterations

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

with correction

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

Neglecting the sub-iterations in the smoother we write

$$Q^{n+1} = Q^n + v = Q^n + \mathcal{S}(d^n)$$

Observation: Oscillations are damped faster on coarser grid.

Coarse grid correction:

$$Q^{n+1} = Q^n + v = Q^n + \mathcal{P}\mathcal{S}\mathcal{R}(d^n)$$

where $\mathcal{R}$ is suitable restriction operator and $\mathcal{P}$ a suitable prolongation operator

Two-grid correction method

Relaxation on current grid:

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

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

Two-grid correction method

Relaxation on current grid:

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

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

Algorithm:

$$\begin{aligned}
\bar{Q} &:= \mathcal{S}(Q^n, \psi, \nu) \\
d &:= \psi - \mathcal{A}(\bar{Q})
\end{aligned}$$

$$\begin{aligned}
d_c &:= \mathcal{R}(d) \\
v_c &:= \mathcal{S}(0, d_c, \mu) \\
v &:= \mathcal{P}(v_c) \\
Q^{n+1} &:= \bar{Q} + v
\end{aligned}$$

Two-grid correction method

Relaxation on current grid:

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

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

Algorithm:

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

with smoothing:

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

Two-grid correction method

Relaxation on current grid:

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

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

Algorithm:

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

with smoothing:

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

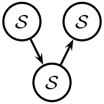
with pre- and post-iteration:

$$\begin{aligned}
d &:= \psi - \mathcal{A}(Q)\\
v &:= \mathcal{S}(0, d, \nu_1)\\
r &:= d - \mathcal{A}(v)\\
d_c &:= \mathcal{R}(r)\\
v_c &:= \mathcal{S}(0, d_c, \mu)\\
v &:= v + \mathcal{P}(v_c)\\
d &:= d - \mathcal{A}(v)\\
r &:= \mathcal{S}(0, d, \nu_2)\\
Q^{n+1} &:= Q + v + r
\end{aligned}$$

[Hackbusch, 1985]

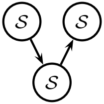
Multi-level methods and cycles

V-cycle
$\gamma = 1$
2-grid

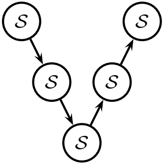


Multi-level methods and cycles

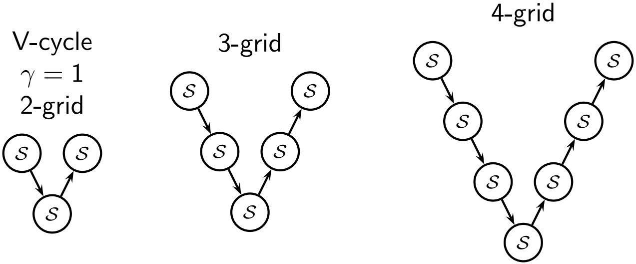
V-cycle
$\gamma = 1$
2-grid



3-grid

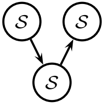


Multi-level methods and cycles

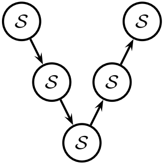


Multi-level methods and cycles

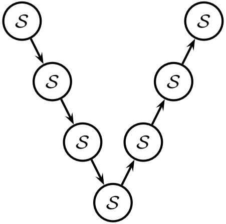
V-cycle
$\gamma = 1$
2-grid



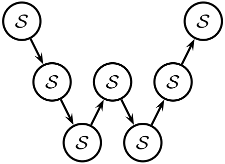
3-grid



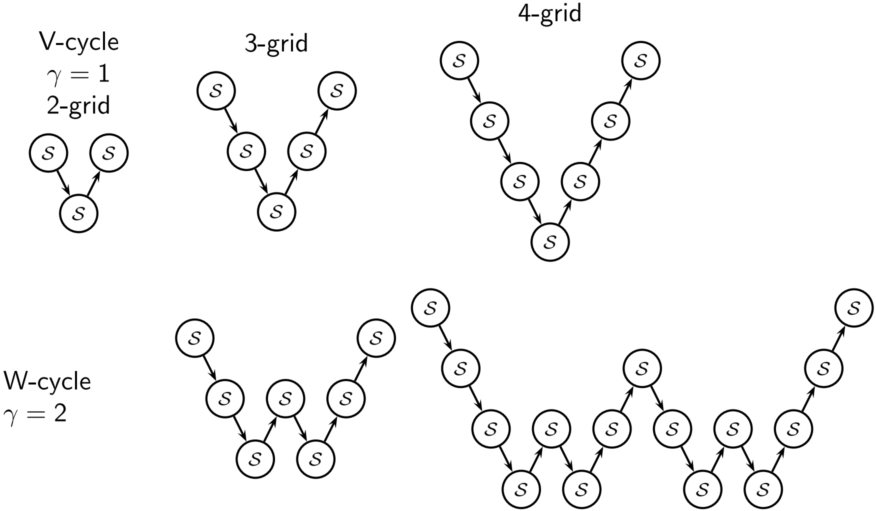
4-grid



W-cycle
$\gamma = 2$



Multi-level methods and cycles



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

Stencil modification at coarse-fine boundaries in 1D

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

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

Stencil modification at coarse-fine boundaries in 1D

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

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

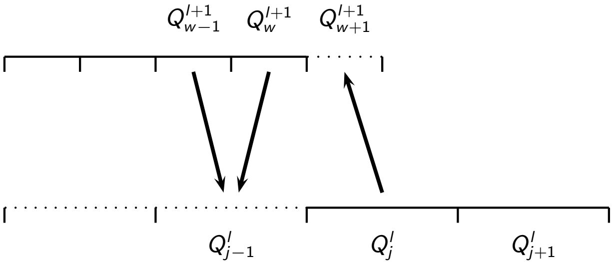
H is approximation to *derivative* of Q^l.

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 $r_{l+1} = 2$:

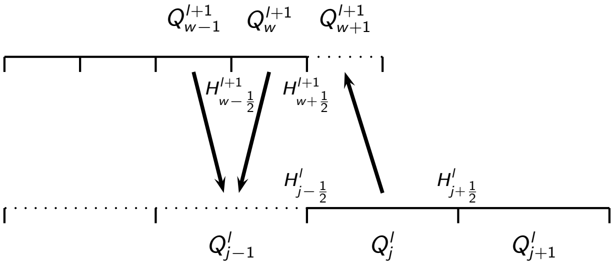


Stencil modification at coarse-fine boundaries in 1D

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

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

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

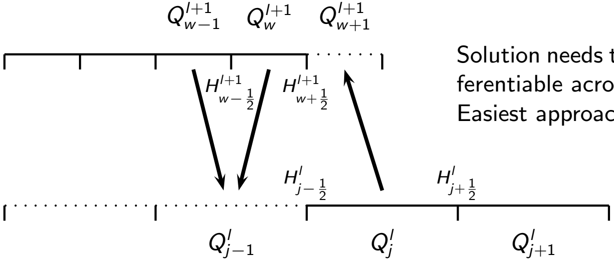


Stencil modification at coarse-fine boundaries in 1D

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

$$d^l_j = \psi_j - \frac{1}{\Delta x_l}\left(\frac{1}{\Delta x_l}(Q^l_{j+1} - Q^l_j) - \frac{1}{\Delta x_l}(Q^l_j - Q^l_{j-1})\right) = \psi_j - \frac{1}{\Delta x_l}\left(H^l_{j+\frac{1}{2}} - H^l_{j-\frac{1}{2}}\right)$$

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



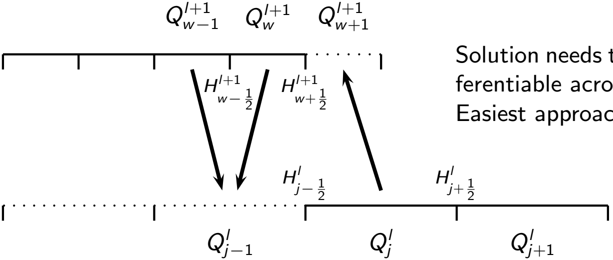
Solution needs to be continuously differentiable across interface.
Easiest approach: $H^{l+1}_{w+\frac{1}{2}} \equiv H^l_{j-\frac{1}{2}}$

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 $r_{l+1} = 2$:



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 $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$.

Stencil modification at coarse-fine boundaries in 1D II

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

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

Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $l+1$.

Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $H^{l}_{j-\frac{1}{2}} \equiv H_{\mathcal{I}}$.

Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $H^{l}_{j-\frac{1}{2}} \equiv H_{\mathcal{I}}$.

Correction pass [Martin, 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 $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $H^{l}_{j-\frac{1}{2}} \equiv H_{\mathcal{I}}$.

Correction pass [Martin, 1998]

1. $\delta H^{l+1}_{j-\frac{1}{2}} := -H^{l}_{j-\frac{1}{2}}$
2. $\delta H^{l+1}_{j-\frac{1}{2}} := \delta H^{l+1}_{j-\frac{1}{2}} + H^{l+1}_{w+\frac{1}{2}} = -H^{l}_{j-\frac{1}{2}} + (Q^{l}_{j} - Q^{l+1}_{w})/\frac{3}{2}\Delta x_{l+1}$

Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $H^{l}_{j-\frac{1}{2}} \equiv H_{\mathcal{I}}$.

Correction pass [Martin, 1998]

1. $\delta H^{l+1}_{j-\frac{1}{2}} := -H^{l}_{j-\frac{1}{2}}$
2. $\delta H^{l+1}_{j-\frac{1}{2}} := \delta H^{l+1}_{j-\frac{1}{2}} + H^{l+1}_{w+\frac{1}{2}} = -H^{l}_{j-\frac{1}{2}} + (Q^{l}_{j} - Q^{l+1}_{w}) / \frac{3}{2}\Delta x_{l+1}$
3. $\check{d}^{l}_{j} := d^{l}_{j} + \frac{1}{\Delta x_l} \delta H^{l+1}_{j-\frac{1}{2}}$

Stencil modification at coarse-fine boundaries in 1D II

Set $H^{l+1}_{w+\frac{1}{2}} = H_{\mathcal{I}}$. 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 $H^{l}_{j-\frac{1}{2}} \equiv H_{\mathcal{I}}$.

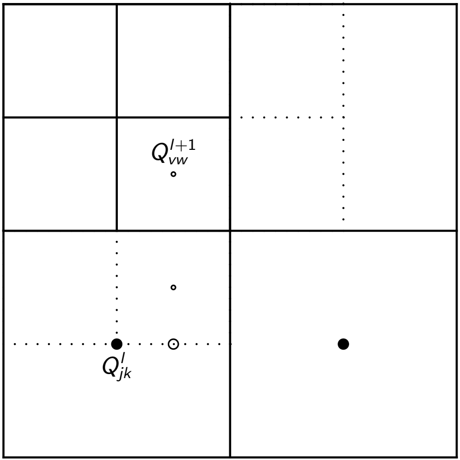
Correction pass [Martin, 1998]

1. $\delta H^{l+1}_{j-\frac{1}{2}} := -H^{l}_{j-\frac{1}{2}}$
2. $\delta H^{l+1}_{j-\frac{1}{2}} := \delta H^{l+1}_{j-\frac{1}{2}} + H^{l+1}_{w+\frac{1}{2}} = -H^{l}_{j-\frac{1}{2}} + (Q^{l}_{j} - Q^{l+1}_{w})/\frac{3}{2}\Delta x_{l+1}$
3. $\check{d}^{l}_{j} := d^{l}_{j} + \frac{1}{\Delta x_l}\delta H^{l+1}_{j-\frac{1}{2}}$

yields

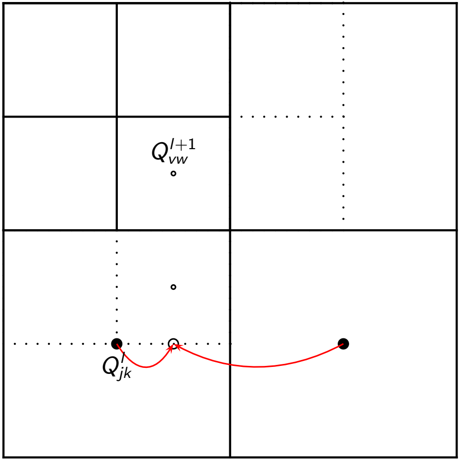
$$\check{d}^{l}_{j} = \psi_j - \frac{1}{\Delta x_l}\left(\frac{1}{\Delta x_l}(Q^{l}_{j+1} - Q^{l}_{j}) - \frac{2}{3\Delta x_{l+1}}(Q^{l}_{j} - Q^{l+1}_{w})\right)$$

Stencil modification at coarse-fine boundaries: 2D



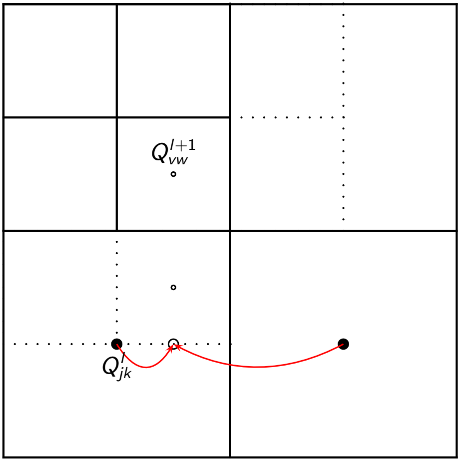
$$Q^{l+1}_{v,w-1} = \qquad +$$

Stencil modification at coarse-fine boundaries: 2D



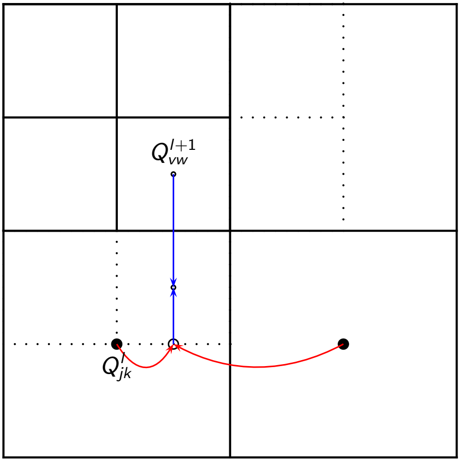
$$Q_{v,w-1}^{l+1} = \qquad +$$

Stencil modification at coarse-fine boundaries: 2D



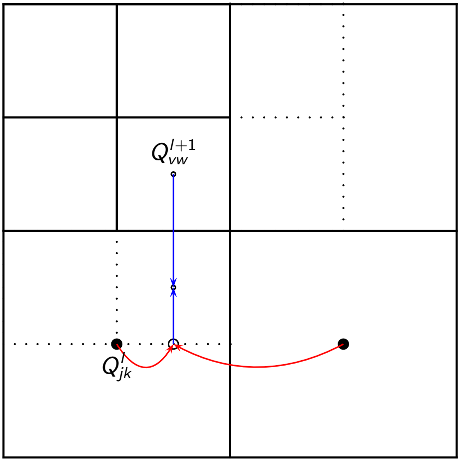
$$Q^{l+1}_{v,w-1} = \qquad + \left(\frac{3}{4}Q^l_{jk} + \frac{1}{4}Q^l_{j+1,k}\right)$$

Stencil modification at coarse-fine boundaries: 2D



$$
\begin{aligned}
Q^{l+1}_{v,w-1} = & \frac{1}{3} Q^{l+1}_{vw} + \\
& \frac{2}{3}\left(\frac{3}{4} Q^{l}_{jk} + \frac{1}{4} Q^{l}_{j+1,k}\right)
\end{aligned}
$$

Stencil modification at coarse-fine boundaries: 2D



$$Q^{l+1}_{v,w-1} = \frac{1}{3} Q^{l+1}_{vw} + \frac{2}{3}\left(\frac{3}{4} Q^{l}_{jk} + \frac{1}{4} Q^{l}_{j+1,k}\right)$$

In general:

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

with

$$f = \frac{x^{v}_{1,l+1} - x^{j}_{1,l}}{\Delta x_{1,l}}$$

Components of an SAMR multigrid method

- Stencil operators

Components of an SAMR multigrid method

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

Components of an SAMR multigrid method

- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ 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

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

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

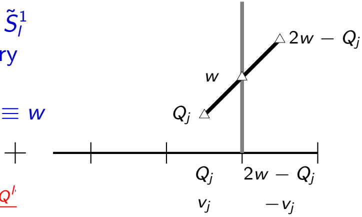
- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level l
  - Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level l
- Boundary (ghost cell) operators
  - Synchronization of Q^l and v^l on $\tilde{S}_l^1$
  - Specification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

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 $\tilde{S}_l^1$
  - 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 $\tilde{I}_l^1$

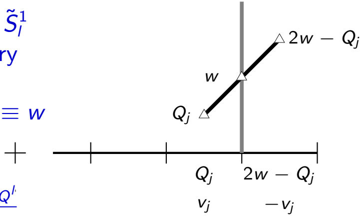
Components of an SAMR multigrid method

- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ 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 $\tilde{S}_l^1$
  - 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 $\tilde{I}_l^1$



Components of an SAMR multigrid method

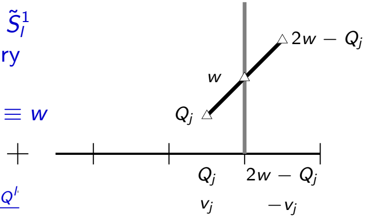
- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level l
  - Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level l
- Boundary (ghost cell) operators
  - Synchronization of Q^l and v^l on $\tilde{S}_l^1$
  - Specification 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 δH^{l+1} on defect d^l

Components of an SAMR multigrid method

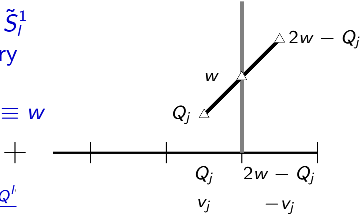
- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level l
  - Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level l
- Boundary (ghost cell) operators
  - Synchronization of Q^l and v^l on $\tilde{S}_l^1$
  - Specification 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 δH^{l+1} on defect d^l
- Standard prolongation and restriction on grids between adjacent levels

Components of an SAMR multigrid method

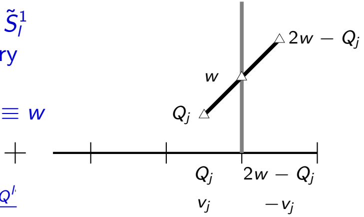
- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level l
  - Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level l
- Boundary (ghost cell) operators
  - Synchronization of Q^l and v^l on $\tilde{S}_l^1$
  - Specification 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 δH^{l+1} on defect d^l
- Standard prolongation and restriction on grids between adjacent levels
- Adaptation criteria

Components of an SAMR multigrid method

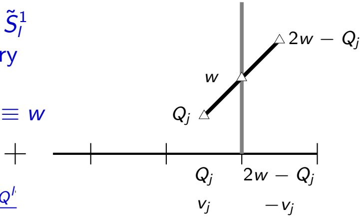
- Stencil operators
  - Application of defect $d^l = \psi^l - \mathcal{A}(Q^l)$ on each grid $G_{l,m}$ of level l
  - Computation of correction $v^l = \mathcal{S}(0, d^l, \nu)$ on each grid of level l
- Boundary (ghost cell) operators
  - Synchronization of Q^l and v^l on $\tilde{S}_l^1$
  - Specification 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 δH^{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 coarsended grid, then use local error estimation

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 $\tilde{S}_l^1$
  - 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 δH^{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 coarsended 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, ψ^l                  d^l := ψ^l − A(Q^l)
 If (l < l_max)
     Calculate updated defect r^{l+1} from v^{l+1}, d^{l+1}     r^{l+1} := d^{l+1} − A(v^{l+1})
     Restrict d^{l+1} onto d^l                           d^l := R_l^{l+1}(r^{l+1})
 Do ν_1 smoothing steps to get correction v^l        v^l := S(0, d^l, ν_1)
 If (l > l_min)
     Do γ > 1 times
         AdvanceLevelMG(l − 1)
     Set ghost cells of v^{l−1}
     Prolongate and add v^{l−1} to v^l                  v^l := v^l + P_l^{l−1}(v^{l−1})
     If (ν_2 > 0)
         Set ghost cells of v^l
         Update defect d^l according to v^l              d^l := d^l − A(v^l)
         Do ν_2 post-smoothing steps to get r^l          r^l := S(v^l, d^l, ν_2)
         Add addional correction r^l to v^l              v^l := v^l + r^l
 Add correction v^l to Q^l                           Q^l := Q^l + v^l
```

Equations used in the scheme:

$d^l := \psi^l - \mathcal{A}(Q^l)$

$r^{l+1} := d^{l+1} - \mathcal{A}(v^{l+1})$

$d^l := \mathcal{R}_l^{l+1}(r^{l+1})$

$v^l := \mathcal{S}(0, d^l, \nu_1)$

$v^l := v^l + \mathcal{P}_l^{l-1}(v^{l-1})$

$d^l := d^l - \mathcal{A}(v^l)$

$r^l := \mathcal{S}(v^l, d^l, \nu_2)$

$v^l := v^l + r^l$

$Q^l := Q^l + v^l$

Additive Geometric Multiplicative Multigrid Algorithm

`Start - Start iteration on level` l_{max}

`For` $l = l_{max}$ `Downto` $l_{min} + 1$ `Do`
`Restrict` Q^l `onto` Q^{l-1}
`Regrid(0)`
`AdvanceLevelMG(`l_{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

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

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

n	A_n	R_n	X_n	Y_n
1	0.3	0.3	6.5	8.0
2	0.2	0.3	2.0	7.0
3	-0.1	0.4	7.0	3.0

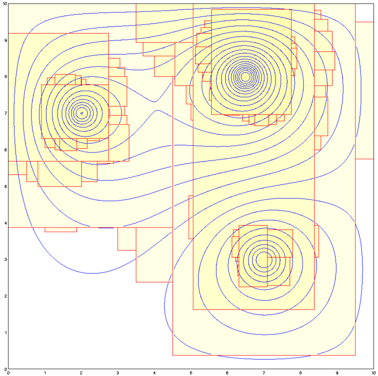
Example

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

n	A_n	R_n	X_n	Y_n
1	0.3	0.3	6.5	8.0
2	0.2	0.3	2.0	7.0
3	-0.1	0.4	7.0	3.0



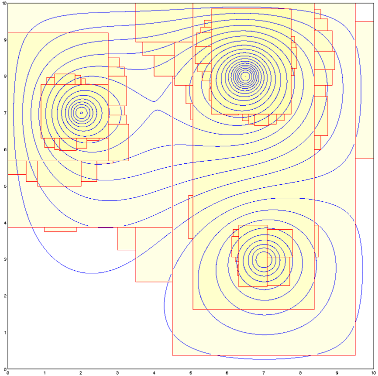
Example

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

n	A_n	R_n	X_n	Y_n
1	0.3	0.3	6.5	8.0
2	0.2	0.3	2.0	7.0
3	-0.1	0.4	7.0	3.0



	128×128	1024×1024	1024×1024
l_{max}	3	0	0
l_{min}	-4	-7	-4
ν_1	5	5	5
ν_2	5	5	5
V-Cycles	15	16	341
Time [sec]	9.4	27.7	563

Stop at $\|d^l\|_{max} < 10^{-7}$ 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.
- The correction is explained in Bell, J. (2004). Block-structured adaptive mesh refinement. Lecture 2. Available at https://ccse.lbl.gov/people/jbb/shortcourse/lecture2.pdf.

References I

[Bastian, 1996] Bastian, P. (1996). *Parallele adaptive Mehrgitterverfahren*. Teubner Skripten zur Numerik. B. G. Teubner, Stuttgart.

[Brandt, 1977] Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computations*, 31(183):333–390.

[Briggs et al., 2001] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2001). *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 2nd edition.

[Canu and Ritzdorf, 1994] Canu, J. and Ritzdorf, H. (1994). Adaptive, block-structured multigrid on local memory machines. In Hackbuch, W. and Wittum, G., editors, *Adaptive Methods-Algorithms, Theory and Applications*, pages 84–98, Braunschweig/Wiesbaden. Proceedings of the Ninth GAMM-Seminar, Vieweg & Sohn.

[Hackbusch, 1985] Hackbusch, W. (1985). *Multi-Grid Methods and Applications*. Springer Verlag, Berlin, Heidelberg.

[Hackbusch, 1994] Hackbusch, W. (1994). *Iterative solution of large sparse systems of equations*. Springer Verlag, New York.

References II

[Martin, 1998] Martin, D. F. (1998). *A cell-centered adaptive projection method for the incompressible Euler equations*. PhD thesis, University of California at Berkeley.

[Trottenberg et al., 2001] Trottenberg, U., Oosterlee, C., and Schüller, A. (2001). *Multigrid*. Academic Press, San Antonio.

[Wesseling, 1992] Wesseling, P. (1992). *An introduction to multigrid methods*. Wiley, Chichester.