Adaptive Multilevel Discretizations for Computational Fluid Dynamics

Ralf Deiterding

Center for Advanced Computing Research
California Institute of Technology
ralf@cacr.caltech.edu

1. Introduction: AMR methods for finite volume schemes
2. Blockstructured AMR for hyperbolic problems
   - The Berger-Colella method
   - Adaptive detonation simulations
3. Blockstructured AMR for elliptic problems
   - Geometric multigrid
   - Case study for the Poisson equation
4. Implementation of AMR algorithms in AMROC
   - Parallelization
5. Outlook: Embedded boundaries methods
   - Blockstructured AMR within the Virtual Test Facility
Conservation Law: \( \partial_t q(x, t) + \nabla \cdot f(q(x, t)) = s(q(x, t)) + \nabla \cdot g(q(x, t), \nabla q(x, t)) \), \( x \in \Omega \subset \mathbb{R}^d, \ t > 0 \)

Integral form (Gauss's theorem):
\[
\int_{\Omega} q(x, t+\Delta t) \, dx - \int_{\Omega} q(x, t) \, dx + \int_{t}^{t+\Delta t} \int_{\partial \Omega} f(q(o, t)) \, do \, dt = \int_{t}^{t+\Delta t} \int_{\Omega} s(q(x, t)) \, dx \, dt + \int_{t}^{t+\Delta t} \int_{\partial \Omega} g(q(o, t), \nabla q(o, t)) \, do \, dt
\]

Finite Volume Discretization in 1D

Integrate over volume \( I_j = [x_j - \frac{1}{2} \Delta x, x_j + \frac{1}{2} \Delta x] = [x_{j-1/2}, x_{j+1/2}] \).

Approximation \( Q_j(t) \approx \frac{1}{|I_j|} \int_{I_j} q(x, t) \, dx \), \( s(Q_j(t)) \approx \frac{1}{|I_j|} \int_{I_j} s(q(x, t)) \, dx \), and numerical fluxes
\[
F(Q_j(t), Q_{j+1}(t)) \approx f(q(x_{j+1/2}, t)), \quad G(Q_j(t), Q_{j+1}(t)) \approx g(q(x_{j+1/2}, t), \nabla q(x_{j+1/2}, t))
\]
yield
\[
Q_j(t_{n+1}) = Q_j(t_n) - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} [F(Q_j(t), Q_{j+1}(t)) - F(Q_{j-1}(t), Q_{j}(t))] \, dt - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} [G(Q_j(t), Q_{j+1}(t)) - G(Q_{j-1}(t), Q_{j}(t))] \, dt + \int_{t_n}^{t_{n+1}} s(Q_j(t)) \, dt
\]
Motivation for Adaptive Mesh Refinement: Detonations Waves

1. Extremely high spatial resolution in reaction zone necessary. Discretization of an exact ZND detonation:

Minimal spatial resolution: $7 - 8 \text{ Pts/lig} \rightarrow \Delta x \approx 0.2 - 0.175 \text{ mm}$
Uniform grids for typical geometries: $> 10^7 \text{ Pts in 2D, } > 10^9 \text{ Pts in 3D} \rightarrow$ Self-adaptive finite volume method (AMR)

2. Problem size even with AMR in 3D enormous $\rightarrow$ parallelization for massively parallel systems with distributed memory

Elements of Parallel AMR Methods

- Base grid
- Solver
- Error indicators
- Grid manipulation
- Interpolation (restriction and prolongation)
- Load-balancing
Unstructured Approach

- Coarse cells replaced by finer ones
- Global time-step
- Cell-based data structures
- Neighborhoods have to stored
  - Geometric flexible
  - No hanging nodes
  - Easy to implement
    - Higher order difficult to achieve
    - Cell aspect ratio must be considered
    - Fragmented data
    - Cache-reuse / vectorization nearly impossible
    - Complex load-balancing
    - Complex synchronization
Quad-tree-based Structured Approach

- Block-based data of equal size
- Time-step refinement
- Block stored in a quad-tree
- Global index coordinate system
- Neighborhoods need not be stored
  + Numerical scheme only for single regular block necessary
  + Easy to implement
  + Simple load-balancing
  + Parent/Child relations according to tree
  +/- Cache-reuse / vectorization only in data block

Wasted boundary space in a quad-tree.
Blockstructured Adaptive Mesh Refinement (SAMR)

- Refined block overlay coarser ones
- Time-step refinement
- Block-based data structures
- Global index coordinate system
  + Numerical scheme only for single regular block necessary
  + Efficient cache-reuse / vectorization possible
  + Simple load-balancing
  + Minimal synchronization overhead
- Cells without mark are refined
- Cluster-algorithm necessary
- Hanging nodes unavoidable
- Difficult to implement
A Refinement Grid

Parts of A Refinement Grid

Domain of level $l$: $G_l := \bigcup_{m=1}^{M_l} G_{l,m}$ with $G_{l,m} \cap G_{l,n} = \emptyset$ für $m \neq n$

Refinements are properly nested: $G_l^{n} \cap G_{l-1} = G_l^{n} \cap G_0$
The Recursive Berger-Colella Algorithm

\texttt{AdvanceLevel}(l)

Repeat \(r_l\) times

Set ghost cells of \(Q_l(t)\)

If time to regrid?

\texttt{Regrid}(l)

\texttt{UpdateLevel}(l)

If level \(l+1\) exists?

Set ghost cells of \(Q_{l+1}(t+\Delta t_l)\)

\texttt{AdvanceLevel}(l+1)

Average \(Q_{l+1}(t+\Delta t_l)\) onto \(Q_l(t+\Delta t_l)\)

Correct \(Q_l(t+\Delta t_l)\) with \(\delta F_{l+1}\)

\(t := t + \Delta t_l\)

\texttt{Regrid}(l) - Regrid all levels \(i > l\)

For \(i = l_c\) Down to \(l\) Do

Flag \(N^i\) according to \(Q_i(t)\)

If level \(i + 1\) exists?

Flag \(N^i\) below \(\tilde{G}_{i+2}\)

Flag buffer zone on \(N^i\)

Generate \(\tilde{G}_{i+1}\) from \(N^i\)

EnsureNesting \(\tilde{G}_{l+1}, \ldots, \tilde{G}_{l_c+1}\)

\texttt{Recompose}(l)

---

\texttt{Start} - Start integration on level 0

\(l = 0, \ r_0 = 1\)

\texttt{AdvanceLevel}(l)

Refinement factor on level \(l\): \(r_l = \Delta t_{l-1}/\Delta t_l\)

---

Conservative Correction

Example: Cell $j, k$

$$
\tilde{Q}^l_{jk}(t + \Delta t_l) = Q^l_{jk}(t) - \frac{\Delta t_l}{\Delta x_{1,l}} \left( F^{1,l}_{j+\frac{1}{2},k} - \frac{1}{r_{l+1}} \sum_{\kappa=0}^{r_{l+1}-1} \sum_{i=0}^{r_{l+1}-1} F^{1,l+1}_{v+\frac{1}{2},w+\ell}(t + \kappa \Delta t_{l+1}) \right) \\
- \frac{\Delta t_l}{\Delta x_{2,l}} \left( F^{2,l}_{j,k+\frac{1}{2}} - F^{2,l}_{j,k-\frac{1}{2}} \right)
$$

Correction pass:

1. $\delta F^{1,l+1}_{j-\frac{1}{2},k} := -F^{1,l}_{j-\frac{1}{2},k}$

2. $\delta F^{1,l+1}_{j-\frac{1}{2},k} := \delta F^{1,l+1}_{j-\frac{1}{2},k} + \frac{1}{r_{l+1}} \sum_{i=0}^{r_{l+1}-1} F^{1,i+1}_{v+\frac{1}{2},w+\ell}(t + \kappa \Delta t_{l+1})$

3. $\tilde{Q}^l_{jk}(t + \Delta t_l) := Q^l_{jk}(t + \Delta t_l) + \frac{\Delta t_l}{\Delta x_{1,l}} \delta F^{1,l+1}_{j-\frac{1}{2},k}$
Euler Equations

\[
\begin{align*}
\partial_t \rho_i + \nabla \cdot (\rho \mathbf{u}) &= \dot{\omega}_i \text{ für } i = 1, \ldots, K \\
\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p &= 0 \\
\partial_t (\rho E) + \nabla \cdot ((\rho E + p) \mathbf{u}) &= 0
\end{align*}
\]

Ideal gas law and Dalton’s law for gas-mixtures:

\[
p(\rho_1, \ldots, \rho_K, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal{R}}{W_i} T = \rho \frac{\mathcal{R}}{W} T \quad \text{with} \quad \sum_{i=1}^{K} \rho_i = \rho \quad \text{and} \quad Y_i = \frac{\rho_i}{\rho}.
\]

Caloric equation:

\[
h(Y_1, \ldots, Y_K, T) = \sum_{i=1}^{K} Y_i h_i(T) \quad \text{with} \quad h_i(T) = h_0^i + \int_0^T c_{pi}(s) ds
\]

Computation of \( T = T(\rho_1, \ldots, \rho_K, e) \) from implicit equation

\[
\sum_{i=1}^{K} \rho_i h_i(T) - \mathcal{R} T \sum_{i=1}^{K} \frac{\rho_i}{W_i} - \rho e = 0
\]

for thermally perfect gases with \( \gamma_i(T) = c_{pi}(T)/c_{vi}(T) \).

Arrhenius-Kinetics:

\[
\dot{\omega}_i = \sum_{j=1}^{M} (\nu_{j}^r - \nu_{j}^f) \left[ k_j^f \prod_{n=1}^{K} \left( \frac{\rho_n}{W_n} \right)^{\nu_{j}^f_n} - k_j^r \prod_{n=1}^{K} \left( \frac{\rho_n}{W_n} \right)^{\nu_{j}^r_n} \right] 
\]

\( i = 1, \ldots, K \)

- Parsing of mechanisms with Chemkin-II
- Evaluation of \( \dot{\omega}_i \) with automatically generated optimized Fortran-77 functions in the line of Chemkin-II
Fractional Step Methods

Solve homogeneous PDE and ODE successively!

\[ \mathcal{H}^{(\Delta t)}: \quad \partial_t \mathbf{q} + \nabla \cdot \mathbf{f}(\mathbf{q}) = 0, \quad \text{IC: } Q(t_m) \xrightarrow{\Delta t} \tilde{Q} \]

\[ \mathcal{S}^{(\Delta t)}: \quad \partial_t \mathbf{q} = \mathbf{s}(\mathbf{q}), \quad \text{IC: } \tilde{Q} \xrightarrow{\Delta t} Q(t_m + \Delta t) \]

1st-order: \( Q(t_m + \Delta t) = \mathcal{S}^{(\Delta t)} \mathcal{H}^{(\Delta t)}(Q(t_m)) \), 2nd-order: \( Q(t_m + \Delta t) = \mathcal{S}^{(\frac{1}{2} \Delta t)} \mathcal{H}^{(\Delta t)} \mathcal{S}^{(\frac{1}{2} \Delta t)}(Q(t_m)) \)

ODE integration in \( \mathcal{S}(\cdot) \) for Euler equations with chemical reaction

- Standard implicit or semi-implicit ODE-solver subcycles within each cell
- \( \rho, e, \mathbf{u} \) remain unchanged!

\[ \partial_t \rho_i = W_i \dot{\omega}_i(\rho_1, \ldots, \rho_K, T) \quad i = 1, \ldots, K \]

Use Newton or bisection method to compute \( T \) iteratively.

Dimensional splitting for \( \mathcal{H}(\cdot) \):

\[ \mathcal{X}_1^{(\Delta t)}: \quad \partial_t \mathbf{q} + \partial_{x_1} \mathbf{f}_1(\mathbf{q}) = 0, \quad \text{IC: } Q(t_m) \xrightarrow{\Delta t} \tilde{Q}^{1/2} \]

\[ \mathcal{X}_2^{(\Delta t)}: \quad \partial_t \mathbf{q} + \partial_{x_2} \mathbf{f}_2(\mathbf{q}) = 0, \quad \text{IC: } \tilde{Q}^{1/2} \xrightarrow{\Delta t} \tilde{Q} \]

Conservative quasi-1D finite volume methods:

\[ \mathcal{X}_1^{(\Delta t)} : \quad \tilde{Q}_{jk}^{m+\frac{1}{2}} = Q_{jk}^{m} - \frac{\Delta t}{\Delta x_1} \left[ F_1(Q_{j-s+1,k}, \ldots, Q_{j+s,k}) - F_1(Q_{j-s,k}, \ldots, Q_{j+s-1,k}) \right], \]

\[ \mathcal{X}_2^{(\Delta t)} : \quad \tilde{Q}_{jk}^{m+1} = \tilde{Q}_{jk}^{m+\frac{1}{2}} - \frac{\Delta t}{\Delta x_2} \left[ F_2(\tilde{Q}_{j,k-s+1}^{m+\frac{1}{2}}, \ldots, \tilde{Q}_{j,k+s}^{m+\frac{1}{2}}) - F_2(\tilde{Q}_{j,k-s}^{m+\frac{1}{2}}, \ldots, \tilde{Q}_{j,k+s-1}^{m+\frac{1}{2}}) \right] \]

1st-order: \( \mathcal{H}(\Delta t) = \mathcal{X}_2^{(\Delta t)} \mathcal{X}_1^{(\Delta t)}(Q(t_m)) \), 2nd-order: \( \mathcal{H}(\Delta t) = \mathcal{X}_1^{(\frac{1}{2} \Delta t)} \mathcal{X}_2^{(\Delta t)} \mathcal{X}_1^{(\frac{1}{2} \Delta t)}(Q(t_m)) \)
Roe's Approximate Riemann Solver

Appropriate matrix\[ A(\hat{Q}) = R(\hat{Q})\Lambda(\hat{Q})R^{-1}(\hat{Q}) \]

Wave decomposition: \[ \Delta Q = Q_r - Q_l = \sum_m a_m \hat{r}_m \]

\[ F(Q_l, Q_r) = f(Q_l) - \sum_{\lambda_m > 0} \hat{\lambda}_m a_m \hat{r}_m = f(Q_r) + \sum_{\lambda_m < 0} \hat{\lambda}_m a_m \hat{r}_m \]

\[ = \frac{1}{2} \left( f(Q_l) + f(Q_r) - \sum_m |\hat{\lambda}_m| a_m \hat{r}_m \right) \]

Insert appropriate average state based on \[ \hat{x} = \frac{x_l\sqrt{\rho_l} + x_r\sqrt{\rho_r}}{\sqrt{\rho_l} + \sqrt{\rho_r}} \] into \( A \) as \( A(\hat{Q}) \).

Harten-Lax-Van Leer (HLL) Approxmate Riemann Solver

\[ F_{HLL}(Q_l, Q_r) = \begin{cases} 
  f(Q_l), & 0 < s_1, \\
  \frac{s_3 f(Q_l) - s_1 f(Q_r) + s_1 s_3 (Q_r - Q_l)}{s_3 - s_1}, & 0 \leq s_1 \leq s_3, \\
  f(Q_r), & 0 > s_3,
\end{cases} \]

\[ s_1 = \min(u_{1,l} - c_l, u_{1,r} - c_r), \quad s_3 = \max(u_{1,l} + c_l, u_{1,r} + c_r) \]

Switch from Roe to HLL scheme near vacuum state to avoid unphysical values.

\[ \bar{Q}(x,t) = \begin{cases} 
  Q_l, & x < s_1 t \\
  Q^*, & s_1 t \leq x \leq s_3 t \\
  Q_r, & x > s_3 t
\end{cases} \]
A Robust and Reliable Roe-type Scheme

(S1) Calculate standard Roe-averages \( \hat{\rho}, \hat{u}_n, \hat{H}, \hat{Y}_i, \hat{T} \).

(S2) Compute \( \hat{\gamma} := \frac{\hat{c}_p}{\hat{c}_v} \) with \( \hat{c}_{(p/v)i} = \frac{1}{T_r - T_l} \int_{T_l}^{T_r} c_{(p/v)i}(\tau) d\tau \).

(S3) Calculate \( \hat{\phi}_i := (\hat{\gamma} - 1) \left( \frac{\hat{u}^2}{2} - \hat{h}_i \right) + \hat{\gamma} R_i \hat{T} \) with standard Roe-averages \( \hat{e}_i \) or \( \hat{h}_i \).

(S4) Calculate \( \hat{c} := \left( \sum_{i=1}^{K+d} \hat{Y}_i \hat{\phi}_i - (\hat{\gamma} - 1) \hat{u}^2 + (\hat{\gamma} - 1) \hat{H} \right)^{1/2} \).

(S5) Use \( \Delta Q = Q_r - Q_l \) and \( \Delta p \) to compute the wave strengths \( a_m \).

(S6) Calculate \( W_1 = a_1 \hat{r}_1, W_2 = \sum_{i=2}^{K+d} a_i \hat{r}_i, W_3 = a_{K+d+1} \hat{r}_{K+d+1} \).

(S7) Evaluate \( s_1 = \hat{u}_1 - \hat{c}, s_2 = \hat{u}_1, s_3 = \hat{u}_1 + \hat{c} \).

(S8) Evaluate \( \rho^*_{l/r}, u^*_{l/r}, e^*_{l/r}, c^*_{l/r} \) from \( Q^*_l = Q_l + W_1 \) and \( Q^*_r = Q_r - W_3 \).

(S9) If \( \rho^*_{l/r} \leq 0 \) or \( e^*_{l/r} \leq 0 \) use \( F_{HLL}(Q_l, Q_r) \) and go to (S12).

(S10) Entropy correction: Evaluate \( |\tilde{s}_i| \).

\[ F_{Roe}(Q_l, Q_r) = \frac{1}{2} \left( f(Q_l) + f(Q_r) - \sum_{i=1}^{3} |\tilde{s}_i| W_i \right) \]

(S11) Positivity correction: Replace \( F_i \) by

\[ F^*_i = F^*_\rho \cdot \begin{cases} Y_l^i, & F^*_\rho \geq 0, \\ Y_r^i, & F^*_\rho < 0. \end{cases} \]

(S12) Evaluate maximal signal speed by \( S = \max(|s_1|, |s_3|) \).

Possible Entropy corrections

1. Replace \( |s_i| \) by \( |\tilde{s}_i| \) only if \( s_l(Q_l) < 0 < s_r(Q_r) \).
2. \( |\tilde{s}_i| = \begin{cases} |s_i| & \text{if } |s_i| \geq 2\eta \\ \frac{|s_i^2|}{4\eta} + \eta & \text{otherwise} \end{cases} \)

\( \eta = \frac{1}{2} \max_i \{ |s_l(Q_r) - s_r(Q_l)| \} \)

2D modification of entropy correction:

To avoid the carbuncle phenomenon at strong shocks.
Transverse Detonation Structure - Regular Instability

Cellular Structure Simulation - 2D

Domain 62 cm × 3 cm. CJ-detonation placed at $x = 4.6$ cm. Unreacted pocket 0.3 cm behind detonation front.


- Adaption criteria:
  1. Scaled gradients of $\rho$ and $p$
  2. Error estimation in $Y_i$ by Richardson extrapolation

$$Q^n \Delta t, \Delta x \Rightarrow \hat{Q} \quad \text{and} \quad Q^{n-1} 2\Delta t, 2\Delta x \Rightarrow \hat{Q} \quad \tau = \frac{|\hat{Q} - Q|}{2^{d+1} - 2}$$

- Quasi-stationary. Unburned gas flows in with CJ velocity.
- Coarse grid shifted through domain with CJ velocity to obtain triple point tracks.

$$\max_t |\omega| \quad \text{with} \quad \omega = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}$$

- 11.2 to 44.8 Pts/$l_{ig}$. 2 refinement levels (2,4) or (4,4).
- 36k instead 128k (11.2 Pts/$l_{ig}$) to 450k cells instead of 2.0 M (44.8 Pts/$l_{ig}$).
Grid Comparison

PC-Cluster of 7 Pentium III-850MHz-PCs connected with 1 GHz Myrinet.
Flow Around a Triple-point

AB: Reflected shock
ACF: Slip-line
FH: Diff. Extension of ACF
B: Triple-point
BC: Reflected shock of B
BDE: Slip-line of B
C: Triple-point
CD: Slip-line of C
DL: Diff. extension of CD
G: Diff. extension of fire
EJ, EK: Diff. extension of fire and of BDE

Primary triple-point moving downwards

Mach-stem shock

Induction length

Incident shock

Temperature [K]

Mass Fraction OH [\text{-}]
Collision of Triple Points and Development of the Double Mach Structure

$t = 620.4 \mu s$

$t = 621.2 \mu s$

$t = 622.0 \mu s$

$t = 622.8 \mu s$

$t = 623.6 \mu s$

$t = 624.4 \mu s$

$t = 625.2 \mu s$

$t = 626.0 \mu s$
Development of the Triple Mach Structure Out of the Double Mach Structure Before the Next Collision

$t = 626.0 \mu s$

$t = 628.4 \mu s$

$t = 629.2 \mu s$

$t = 630.0 \mu s$

$t = 630.8 \mu s$

$t = 632.4 \mu s$

$t = 634.0 \mu s$
Cellular Structure Simulation - 3D

- Adaption criteria:
  1. Scaled gradients of $\rho$ and $p$
  2. Error estimation in $Y_i$ by Richardson extrapolation
- Quasi-stationary. Unburned gas flows in with CJ velocity.
- Coarse grid shifted through domain with CJ velocity to obtain triple point tracks.
- Symmetry boundary conditions in $x_2$-direction.
- 16.8 Pts/$l_{ig}$. 2 refinement levels (2,3).
- Adaptive computation uses 2.0M-2.5M cells instead of 8.7M cells (uniform grid).
- 74h real time on 48 nodes Athlon 1.4GHz.
Periodicity of the Solution

Front view of the periodic solution

$t = 680 \mu s + 600 \mu s$ (Computation 1)

$t = 660 \mu s + 620 \mu s$ (Computation 2)
Schlieren Plot of Density

$t = 676 \mu s + 600 \mu s$
$t = 680 \mu s + 600 \mu s$
$t = 684 \mu s + 600 \mu s$
$t = 688 \mu s + 600 \mu s$

$t = 692 \mu s + 600 \mu s$
$t = 696 \mu s + 600 \mu s$
$t = 700 \mu s + 600 \mu s$
$t = 704 \mu s + 600 \mu s$
Schlieren Plot of Mass Fraction OH

Animation
Temporal Development of Detonation Velocity

Point-wise reinitiation along L1 (left) and L1’ (right)

\[ Y_{OH} \] during reinitiation

\[ t = 689.6 \, \mu s + 600 \, \mu s \]

\[ t = 690.4 \, \mu s + 600 \, \mu s \]

\[ \frac{d^*_v}{d_{CJ}} \]

Simulation time [\mu s]

Ralf Deiterding
Adaptive Multilevel Discretizations for Computational Fluid Dynamics
Comparison with 2D Simulation

The flow field is different in 2D and 3D, but the oscillation period is identical!
Simulation - Detonation Diffraction

- CJ detonation for \( \text{H}_2 : \text{O}_2 : \text{Ar} / 2 : 1 : 7 \) at \( T_0 = 298 \text{ K} \) and \( p_0 = 10 \text{ kPa} \). \( \lambda_c = 1.6 \text{ cm} \).
- Adaption criteria:
  1. Scaled gradients of \( \rho \) and \( p \)
  2. Error estimation in \( Y_i \) by Richardson extrapolation
- \( 25 \text{ Pts/l}_{ig} \). 5 refinement levels (2,2,2,4).
- Adaptive computations use up to 2.2M (\( w = 6.4 \text{ cm} \equiv 4\lambda_c \)) and 2.4M cells (\( w = 9.6 \text{ cm} \equiv 6\lambda_c \)) instead of 145.6M-150.8M cells (uniform grid).
- 80h (\( w = 4\lambda_c \)) to 84h (\( w = 6\lambda_c \)) real time on 48 nodes Athlon 1.4GHz.

\( t = 240 \). Refinement on 5 levels. \( w = 5\lambda_c \)
Diffraction of a $\text{H}_2 : \text{O}_2 : \text{Ar}$-Detonation

Upper row: $t = 160 \mu s$, lower row: $t = 240 \mu s$

$w = 4\lambda_c$  $w = 5\lambda_c$  $w = 6\lambda_c$

Blockstructured AMR for the Poisson Equation *

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

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


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

2. Synchronization

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

In a finite volume discretization the boundary conditions can only be prescribed iteratively.

Direct method: \( Q^{m+1} := S(Q^m, \psi) \) by iterating on \( AQ = \psi \)

Smoothing (requires patch-wise defect calculation and patch-wise smoother):
\[ d^m := \psi - A(Q^m), \ v^m := S(0, d^m) \) by iterating on \( A(v) = d^m \), \( Q^{m+1} := Q^m + v^m = Q^m + C(d^m) \)

Idea of geometric multigrid: \( Q_{l}^{m+1} := Q_{l}^m + \mathcal{P}_{l-1} S_{l-1} R_{l-1}^{-1}(d_{l}^m) \)

Smoothing of fine grid problem on coarser mesh to achieve faster damping (convergence!) of low frequency oscillations.

Additive Geometric Multiplicative Multigrid Algorithm

AdvanceLevelMG($l$) - Correction Scheme

1. Set ghost cells of $Q_l$
2. Calculate defect $d_l$ from $Q_l, \psi_l$
   
   If ($l < l_{\text{max}}$)
   1. Restrict $d_{l+1}$ onto $d_l$
   2. Do $\nu_1$ smoothing steps to get correction $v_l$
   
   If ($l > l_{\text{min}}$)
   1. Do $\gamma > 1$ times
      1. AdvanceLevelMG($l - 1$)
   2. Set ghost cells of $v_{l-1}$
   3. Prolongate and add $v_{l-1}$ to $v_l$
   4. If ($\nu_2 > 0$)
      1. Set ghost cells of $v_l$
      2. Update defect $d_l$ according to $v_l$
      3. Do $\nu_2$ post-smoothing steps to get $r_l$
      4. Add additional correction $r_l$ to $v_l$

   Add correction $v_l$ to $Q_l$

3. $d_l := \psi_l - A(Q_l)$
4. $d_l := R_{l+1}^l(d_{l+1})$
5. $v_l := S(0, d_l, \nu_1)$
6. $v_l := v_l + P_{l-1}^l(v_{l-1})$
7. $d_l := d_l - A(v_l)$
8. $r_l := S(v_l, d_l, \nu_2)$
9. $v_l := v_l + r_l$
10. $Q_l := Q_l + v_l$

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

For $l = l_{\text{max}}$ Down to $l_{\text{min}} + 1$
1. Restrict $Q_l$ onto $Q_{l-1}$
2. Rgrid(0)
3. AdvanceLevelMG($l_{\text{max}}$)
4. $Q_{l-1} := R_{l}^{-1}(Q_l)$
Modification of Stencil at Coarse-Fine Boundaries

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

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

Correction pass:

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

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

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

\[
\tilde{d}_j^l = \psi_j - \frac{1}{\Delta x_l} \left( \frac{1}{\Delta x_l} (Q_{j+1}^l - Q_j^l) - \frac{1}{\Delta x_{l+1}} (Q_j^l - Q_{w+1}^l) \right)
\]
Example

Stop at $\|d_l\|_{max} < 10^{-7}$ for $l \geq 0$, $\gamma = 1$, $r_l = 2$
Design of AMROC

AMROC

AMROC’s DAGH

The concept of GridFunctions

AMROC with Clawpack:
http://amroc.sourceforge.net
• Rigorous domain decomposition approach
• Data of all levels resides on same node
• Grid hierarchy defines unique "floor-plan"
• Redistribution of data blocks during reorganization of hierarchical data
• Synchronization when setting ghost cells

Parallelization Strategy

• Strictly local calculation of flux correction terms
• Only application requires communication
Partitioning

Calculation domain

Necessary domain of Space-Filling Curve

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray</td>
<td>Proc. 1</td>
</tr>
<tr>
<td>Patterned</td>
<td>High Workload</td>
</tr>
<tr>
<td>White</td>
<td>Proc. 2</td>
</tr>
<tr>
<td>Patterned</td>
<td>Medium Workload</td>
</tr>
<tr>
<td>Gray</td>
<td>Proc. 3</td>
</tr>
<tr>
<td>Patterned</td>
<td>Low Workload</td>
</tr>
</tbody>
</table>
Embedded Boundary Methods

Representation of a moving boundary in a Cartesian method.

1. Methods that diffuse the boundary in one cell:
   - E.g. internal ghost cell values
   - Not conservative by construction
   - Usually combined with implicit geometry representation

2. Methods that represent the boundary sharply:
   - Cut-cell approach constructs appropriate finite volumes.
   - Conservative by construction. Correct boundary flux.
   - Key question: How to avoid small-cell time step restriction?
   - Usually explicit geometry representation used

K. J. Richards et al., On the use of the immersed boundary method for engine modeling

Ralf Deiterding
Adaptive Multilevel Discretizations for Computational Fluid Dynamics
SAMR within the Virtual Test Facility *

- Construct AMR framework for Cartesian finite volume methods that supports the combination of embedded boundary and ghost-fluid methods.
- Implicit geometry representation based on level sets.
- Transformation of moving surface meshes into level set with exact and optimal Closest-point-transform algorithm by Sean Mauch.

![b-rep](image)

**b-rep**

Surface mesh  $\phi(x_1, x_2)$  Distance  Normal $n = \nabla \phi / |\nabla \phi|$ to closest point

![Distance](image)

- Implement diffused boundary method first, but consider accurate embedded boundary method based on

$$V_{j}^{n+1}Q_{j}^{n+1} = V_{j}^{n}Q_{j}^{n} - \Delta t \left( A_{j+1/2}^{n+1/2} F(Q,j) - A_{j-1/2}^{n-1/2} F(Q,j-1) \right)$$

via subsequent correction step after Cartesian finite-volume update step later.

Construction of Boundary Values for Cartesian Fluid Scheme

Moving boundary/interface is treated as a moving contact discontinuity.

Solid-fluid coupling for FEM and FV scheme with mirroring:

*Linear interpolation on fluid nodes*

- One-sided construction of mirrored ghost cell and new FEM nodal point values
- FEM ansatz-function interpolation to obtain intermediate surface values

\[
\begin{align*}
u^F_n &= u^S_n, \\
-p^F &= \sigma^S_n, \\
0 &= \sigma^S_i
\end{align*}
\]

Find values \( \rho_M, p_M, u^F_M \) by interpolation at \( \tilde{x} = x + 2\phi n \)

Velocity in ghost cells

\[
u^F_{Gh} = (2u^S - u^F_M) \cdot n + u^F_M - (u^F_M \cdot n)n
\]

Parallelization Strategy for Coupled Simulations

Coupling of an Eulerian Finite Volume Fluid Solver and a Lagrangian Finite Shell Element Solver:

- Fluid solver operates on volume data, while solid solver uses surface meshes.
- For compressible gas dynamics the size of the fluid problems exceeds the solid problem by magnitudes.
- Effective parallelization approach in distributed memory environment:
  - Distribute both meshes separately and copy necessary nodal values and geometry data to fluid nodes.
  - Setting of ghost cell values becomes strictly local operation.
  - Construct new nodal values strictly local on fluid nodes and transfer them back to solid nodes.
  - Only surface data is transferred.
  - Asynchronous communication ensures scalability
  - Generic encapsulated implementation guarantees reusability
Shock Focusing and Fluid Instability in Converging Geometry

- Experiment by Paul Dimotakis with Amy Lang, Garret Katzenstein, Daniel Lang.
- Focusing of stable shock wave (without Mach reflection).
- Richtmyer-Meshkov instability between two gases in converging geometry.
- Shock interaction with complex geometry and fluid interface.
- Simulation of compressible turbulence requires very accurate higher-order shock capturing method —→ Hybrid Tuned Centered-Difference with Weighted Essentially Non-Oscillatory (WENO) stencil (David Hill).
- Subgrid-scale turbulence modeling for Large-Eddy Simulation (LES) for Euler equations (Dale Pullin, David Hill).
- Validation of simulation (Ravi Samtaney, David Hill) via
  - Shock evolution
  - Mixing layer thickness

Preliminary simulation (Julian Cummings, Michael Aivazis)
Detonation Driven Fracture

- Experiment by Joe Shepherd, T.-W. Chao, J. Austin
- Interaction of detonation, ductile deformation, fracture → solid-fluid coupling of Eulerian AMR to Lagrangian FEM code (Michael Ortiz and his group, Fehmi Cirak) using shells elements with fracture capability
- Validation of simulation (Fehmi Cirak, Patrick Hung) via
  - Stress history of cylinder
  - Crack propagation history
  - Species concentration and detonation fine structure