An AMR-algorithm for distributed memory computers

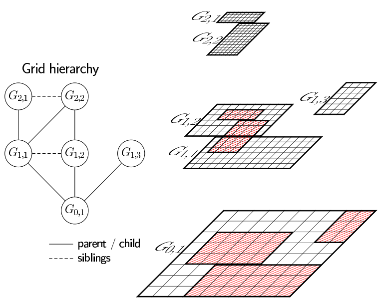
Ralf Deiterding

Chair of Numerical Mathematics & Scientific Computing
Institut of Mathematics, Technische University Cottbus
Germany
{deiterding}@math.tu-cottbus.de
http://www.math.tu-cottbus.de/~deiter

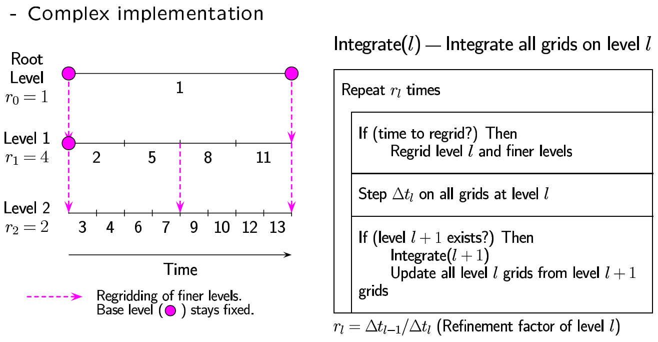
Blockstructured AMR

The AMR algorithm of Berger and Oliger is the most efficient adaptive method for hyperbolic conservation laws on blockstructured grids. Instead of refining single cells a multi-level hierarchy of recursively embedded subgrids is constructed. The underlying regular data structures allow much higher resolved computations than usual cell-based approaches.

+ Discretization necessary only for a single logically rectangular grid
+ Spatial *and* temporal refinement, no global time step restriction
+ No neighboring cell information has to be stored
+ Efficient cache reuse and vectorization possible
+ Simple load balancing
- Appropriate only for simple geometries
- Cluster algorithm necessary for grid generation
- Hanging nodes unavoidable and require special treatment
- Complex implementation



The blockstructured refinement strategy creates a hierarchy of properly nested subgrids.

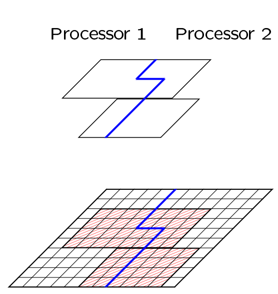


The AMR-algorithm uses a recursive integration procedure that allows the construction of boundary conditions for refined subgrids by time-space interpolation.

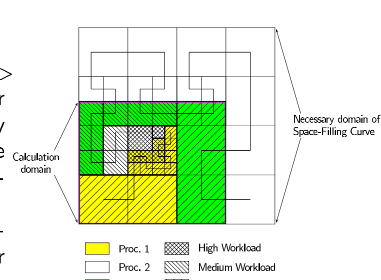
A generic framework for AMR

Three abstraction levels can be identified:

1. Specific application. The demo application is an extended version of Clawpack. Features:
   - Single grid Fortran-functions implement the discretization, initial conditions, boundary conditions, etc. (Usual Clawpack-interfaces, no knowledge of AMR required)
   - Standard discretizations for Euler equations on cartesian grids already implemented, e.g. Van Leer-FVS, Steger-Warming-FVS, Roe's approximative Riemann-solver, exact Riemann-solver
   - Multidimensional wave propagation method with 2nd order correction and wave limiting
   - Dimensional-splitting with MUSCL-extrapolation and slope limiting
2. AMROC (Adaptive Mesh Refinement in Object-oriented C++):
   - AMR-solver and its specific components formulated nearly like in the serial case and independent of the spatial dimension
   - Parallel flux correction algorithm
   - Various exchangeable adaption criteria, e.g. error estimation by Richardson extrapolation, scaled gradients
3. Hierarchical data structures:
   - Distributed GridFunctions<Dim,DataType> automatically follow the "floor plan" of a single Grid Hierarchy
   - Data of all levels resides on the same node → Most AMR operations are strictly local
   - Neighboring grids are synchronized transparently even over processor borders when boundary conditions are applied
   - Distribution algorithm: Generalization of Hilbert's space-filling curve



All higher level data follow the distribution of the base level.



Construction of generalized Hilbert space-filling curve.

Important features

- Multiblock domains
- Periodic boundary conditions
- Restart facility for arbitrary number of nodes
- Output in HDF-format
- Supported visualization tools: Matlab, Visual3, IBM Data Explorer, Gnuplot

Benchmark: Circular expanding shock-wave in a box

- 2D Euler equations for an ideal gas
- Roe's approximative Riemann-solver, wave propagation scheme with Minmod wave-limiter and transverse wave propagation
- 199 time steps with 3 refinement levels. Finest level corresponds to 1200x1200 grid.

Task	P=1		P=2		P=4	
	s	%	s	%	s	%
Integration	9083	80.1	4546	75.8	2246	67.2
Flux correction	460	4.1	251	4.2	180	5.4
Boundary setting	399	3.5	345	5.8	315	9.4
Recomposition	982	8.7	641	10.7	496	14.9
Clustering	221	1.9	114	1.9	37	1.1
Misc.	190	1.8	94	1.7	56	2.0
Total / Parallel Efficiency	**11336**	**100.0**	**5991**	**94.6**	**3329**	**85.1**
AMRClaw / Speed Up	**9893**		**1.65**		**2.97**	

The benchmark is run on a typical PC-Cluster of Pentium III-PC's connected with Fast Ethernet.

Generalized Euler equations

The computation of inviscid flows with detailed chemical reaction requires the usage of generalized Euler equations. In cartesian coordinates the following equations have to be applied:

K continuity equations for K different gaseous species:

$$\partial_t \rho_i + \sum_{n=1}^{N} \partial_{x_n}(\rho_i v_n) = W_i\,\dot\omega_i \qquad \text{for } i = 1,\dots,K$$

N momentum equations:

$$\partial_t(\rho v_m) + \sum_{n=1}^{N} \partial_{x_n}(\rho v_n v_m + \delta_{n,m}\, p) = 0 \qquad \text{for } m = 1,\dots,N$$

Energy equation:

$$\partial_t(\rho E) + \sum_{n=1}^{N} \partial_{x_n}[v_n(\rho E + p)] = 0$$

Equation of state

The species are assumed to be ideal gases in thermal equilibrium. The ideal gas law and Dalton's law can be applied:

$$p(\boldsymbol\rho, T) = \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} \rho_i \frac{\mathcal R}{W_i} T = \rho \frac{\mathcal R}{W} T \qquad \text{with} \quad \rho = \sum_{i=1}^{K} \rho_i$$

Ideal gases are *thermally perfect* and the specific heats are functions of the temperature:

$$c_{pi} = c_{pi}(T) \quad , \quad c_{vi} = c_{vi}(T) \quad , \quad \gamma_i(T) = c_{pi}(T)\,/\,c_{vi}(T)$$

Caloric equation:

$$h(\boldsymbol\rho, T) = \sum_{i=1}^{K} Y_i h_i(T) \quad \text{mit} \quad h_i(T) = h_i^0 + \int_0^T c_{pi}(s)ds$$

Evaluation of $p(\boldsymbol\rho, T)$ requires the computation of $T = T(\boldsymbol U)$ from the implicit equation:

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

Detailed chemistry

The chemical production rates $\dot\omega_i(\rho_1,\dots,\rho_K,T)$ are derived from a *reaction mechanism* that consists of M chemical reactions:

$$\sum_{i=1}^{K} \nu_{ji}^f S_i \rightleftharpoons \sum_{i=1}^{K} \nu_{ji}^r S_i \qquad j = 1,\dots,M$$

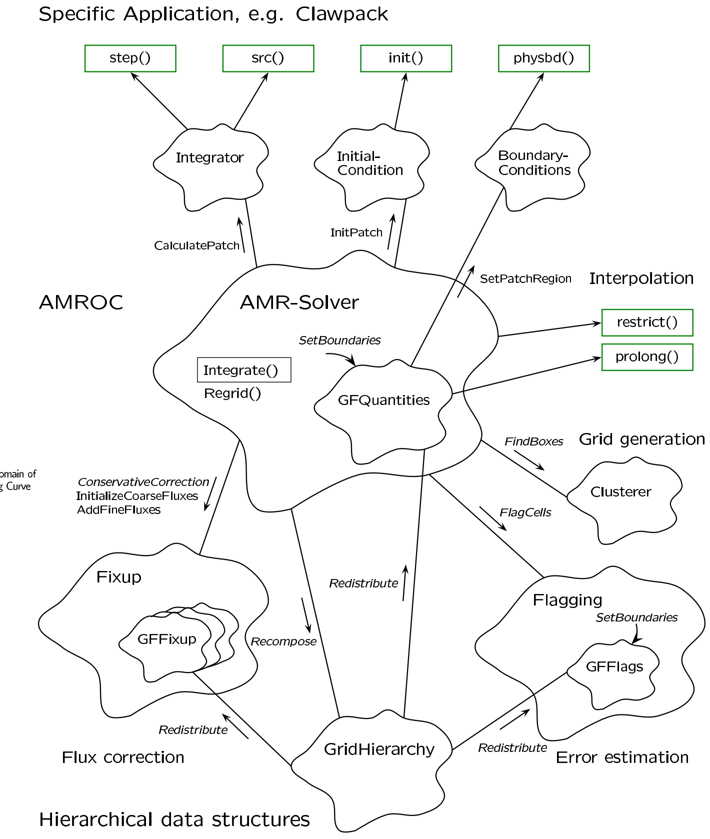
The forward reaction rate $k_j^f(T)$ is calculated with an empirical Arrhenius law:

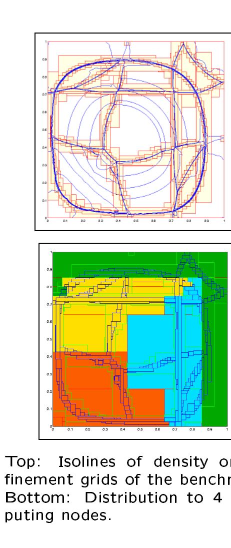
$$k_j^f(T) = A_j T^{\beta_j} \exp(-E_j/\mathcal R T)$$

Evaluation of the equilibrium constant $K_j^c(T)$ allows the calculation of the corresponding backward reaction rate $k_j^r(T) = k_j^f(T)/K_j^c(T)$.

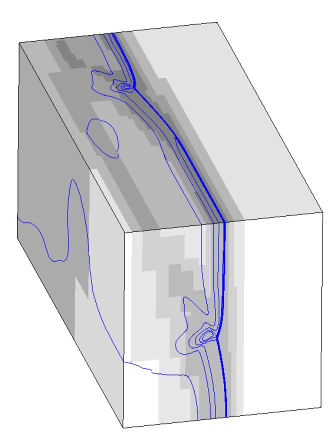
Mass production rate of specie S_i:

$$W_i\,\dot\omega_i = W_i \sum_{j=1}^{M} (\nu_{ji}^r - \nu_{ji}^f)\left[k_j^f \prod_{n=1}^{K}\left(\frac{\rho_n}{W_n}\right)^{\nu_{jn}^f} - k_j^r \prod_{n=1}^{K}\left(\frac{\rho_n}{W_n}\right)^{\nu_{jn}^r}\right] \qquad i = 1,\dots,K$$





Top: Isolines of density on refinement grids of the benchmark. Bottom: Distribution to 4 computing nodes.



Isolines of density on refinement grids of the three-dimensional detonation wave.

Numerical Method

The reactive source term is incorporated in *Integrator* with a fractional-step method:

Successive solution of the homogeneous transport equations and the system of ordinary differential equations

$$\partial_t \rho_i = W_i\,\dot\omega_i\,(\rho_1,\dots,\rho_K,T) \qquad i = 1,\dots,K$$

High Resolution method in *step()*:

- Approximative Riemann solver of Roe-typ
- MUSCL extrapolation of primitive variables Y_i, v_n, p with Van Albada-limiter
- Dimensional splitting in 2D and 3D

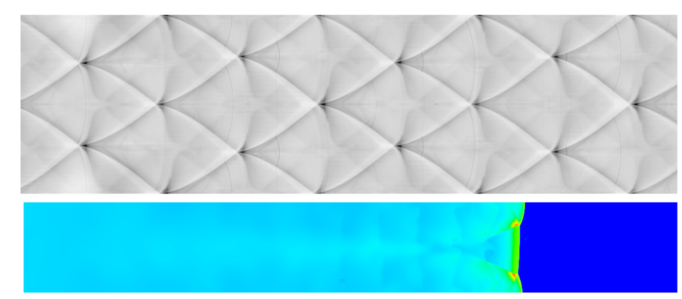
Decoupled source term integration in *src()*:

- Semi-implicit Rosenbrock-Wanner method

The integration of stiff source terms requires automatic stepsize adjustment in a single transport step.

Planar detonation with transverse waves

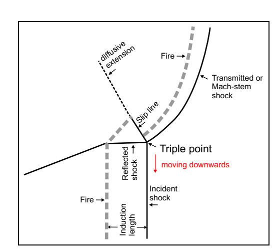
Experiments have shown that self-sustaining detonation waves are locally multidimensional and nonsteady. Tripel-points may form, which enhance the local chemical reaction significantly. Equilibrium-configurations with regular detonation cells are possible in particular cases. The accurate numerical simulation of transverse wave phenomena in detonation waves requires extraordinarily high resolution.



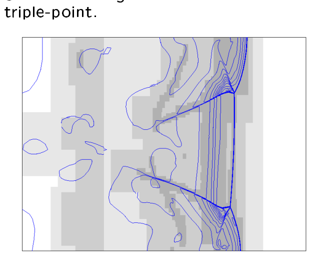
Top: The time history of the released chemical energy shows the detonation cells. Bottom: Interacting transverse waves behind the detonation front.

Reaction mechanism: 34 elementary reactions for the 9 *thermally perfect* species H, O, OH, H_2, O_2, H_2O, HO_2, H_2O_2, Ar.
Configuration: Stoichiometric H_2-O_2-system with 70% Ar, at 6.7 kPa and 298 K.

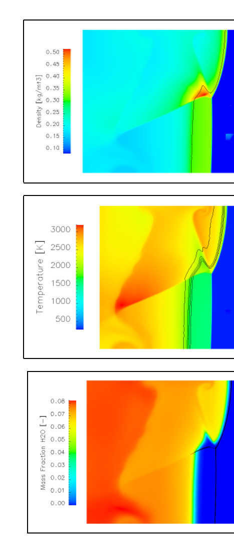
- 1044 time steps with 3 refinement levels (factors: 2,4,4). Finest level corresponds to 19840x640 grid (12.7 M cells).
- ≈ 32 cells within induction length.
- Adaptive computation uses 150k-200k cells.
- 121h real time on 7 nodes Pentium III-750 MHz.



Schematic diagram of the flow around a triple-point.



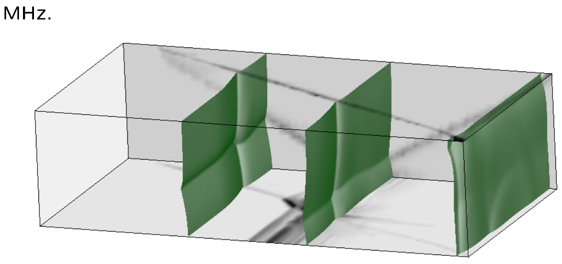
Isolines of density on refinement grids show the dynamic adaption of the detonation wave.



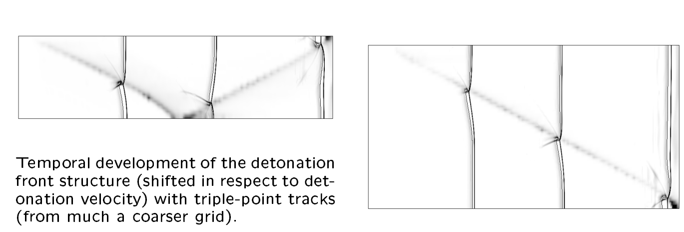
Detailed study of the flow around a triple-point. Additional isolines show the induction length.

Detonation with two orthogonal transverse waves

- Detonation front remains quasi-stationary, because unburned gas flows in with CJ-velocity.
- 264 time steps with 3 refinement levels (factors: 2,2,2). Finest level corresponds to 224x96x192 grid (4.1 M cells).
- ≈ 8 cells within induction length.
- Adaptive computation uses 800k-1.2M cells.
- 66h real time on 15 nodes Pentium III-750 and Pentium III-450 MHz.



Temporal development of the detonation front with two orthogonal triple-point lines (shifted in respect to detonation velocity for visualization).



Temporal development of the detonation front structure (shifted in respect to detonation velocity) with triple-point tracks (from much a coarser grid).