Solving PDEs on Overlapping Grids with Overture

Bill Henshaw

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA

University of Illinois at Urbana-Champaign

September 2010.
Acknowledgments.

Supported by
Department of Energy, Office of Science
  ASCR Applied Math Program
LLNL: Laboratory Directed Research and Development (LDRD) program

Current Overture developers
  Kyle Chand
  Bill Henshaw

Major Contributors
  Don Schwendeman (RPI),
  Jeff Banks (LLNL).
What are overlapping grids and why are they useful?

**Overlapping grid**: a set of structured grids that overlap.

- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.
Aerospace applications using overlapping grids.

Space shuttle figures courtesy of William Chan and Reynaldo Gomez. V-22 Osprey figures courtesy of William Chan, Andrew Wissink and Robert Meakin.
Overture: tools for solving PDE’s on overlapping grids

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities (Ogen).
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).
The CG (Composite Grid) suite of PDE solvers

- **cgad**: advection diffusion equations.
- **cgins**: incompressible Navier-Stokes with heat transfer.
- **cgcns**: compressible Navier-Stokes, reactive Euler equations.
- **cgmp**: multi-physics solver (e.g. conjugate heat transfer).
- **cgmx**: time domain Maxwell’s equations solver.
- **cgsm**: solid mechanics (*new*)

Overture and CG are freely available from the web:

www.llnl.gov/CASC/Overture
Components of an Overlapping Grid

Physical space:

\[ \Omega \]

\[ \partial \Omega \]

physical boundary

Parameter space:

\[ i_2 = 0 \]
\[ i_2 = N_2 \]
\[ i_1 = 0 \]
\[ i_1 = N_1 \]

Component grid 1

Component grid 2

Mapping: \( x = G_2(r) \)

\( bc(2,2) \)

\( bc(1,1) \)
\( bc(1,2) \)
\( bc(2,1) \)

- interpolation
- unused
- ghost point

Henshaw (LLNL)

PDEs on Overlapping Grids

UIUC
Ogen can be used to build 2D overlapping grids:

Solutions coupled by interpolation
Ogen can be used to build 3D overlapping grids:
Volkov, circa [1966] developed a *Composite Mesh* method for Laplace’s equation on regions with piece-wise smooth boundaries separated by corners. Polar grids are fitted at corners to handle potential singularities.


Steger, circa [1980] independently conceives the idea of the overlapping grid, subsequently named the *Chimera* approach after the mythical Chimera beast having a human face, a lion’s mane and legs, a goat’s body, and dragon’s tail. NASA groups develop grid generator PEGSUS, hyperbolic grid generation and flow solver Overflow (Steger, Benek, Suhs, Buning, Chan, Meakin, et. al.)

There is extensive numerical analysis theory underpinning this work.

- classic von Neumann stability analysis (periodic domains).
- energy estimates ($L_2$-norm estimates).
- normal mode analysis, GKS theory (initial boundary value problems).

Some references:

- Gustafsson, Kreiss, Oli"{g}er, *Time Dependent Problems and Difference Methods*, (book).
A one-dimensional overlapping grid example

To solve the advection-diffusion equation

\[ u_t + au_x = \nu u_{xx}, \quad x \in (0, 1) \]

\[ u(0, t) = g_0(t), \quad u_x(1, t) = g_1(t), \quad \text{(boundary conditions)} \]

\[ u(x, 0) = u_0(x), \quad \text{(initial conditions)} \]

introduce grid points on the two overlapping component grids,

\[ x^{(1)}_i = x_a + i\Delta x_1, \quad i = -1, 0, 1, \ldots, N_1 + 1, \quad \Delta x_1 = (x_d - x_a)/N_1 \]

\[ x^{(2)}_j = x_c + (j + 1)\Delta x_2, \quad j = -1, 0, 1, \ldots, N_2 + 1, \quad \Delta x_2 = (x_b - x_c)/N_2 \]

and approximations \( U^n_i \approx u(x^{(1)}_i, n\Delta t), \ V^n_i \approx u(x^{(2)}_i, n\Delta t) \).
Discretize with forward-Euler and central differences

Given the solution at time $t^n$, compute the solution at time $t^{n+1}$:

\[
(U_{i}^{n+1} - U_{i}^{n})/\Delta t = -aD_0 U_{i}^{n} + \nu D_+ D_- U_{i}^{n}, \quad i = 1, 2, \ldots, N_1
\]

\[
(V_{j}^{n+1} - V_{j}^{n})/\Delta t = -aD_0 V_{j}^{n} + \nu D_+ D_- V_{j}^{n}, \quad j = 0, 2, \ldots, N_2
\]

\[
U_{0}^{n+1} = g(t^n), \quad D_0 V_{N_2}^{n+1} = g_1(t^{n+1}), \quad \text{(boundary conditions)}
\]

\[
U_{N_1+1}^{n+1} = (1 - \alpha)(1 - \frac{\alpha}{2}) V_{-1}^{n+1} + \alpha(2 - \alpha) V_{0}^{n+1} + \frac{\alpha}{2}(\alpha - 1) V_{1}^{n+1}, \quad \text{(interpolation)}
\]

\[
V_{-1}^{n+1} = (1 - \beta)(1 - \frac{\beta}{2}) U_{N_1-1}^{n+1} + \beta(2 - \beta) U_{N_1}^{n+1} + \frac{\beta}{2}(\beta - 1) U_{N_1+1}^{n+1}, \quad \text{(interpolation)}
\]

\[
D_0 U_{i}^{n} = \frac{U_{i+1}^{n} - U_{i-1}^{n}}{2\Delta x}, \quad D_+ U_{i}^{n} = \frac{U_{i+1}^{n} - U_{i}^{n}}{\Delta x}, \quad D_- U_{i}^{n} = \frac{U_{i}^{n} - U_{i-1}^{n}}{\Delta x}.
\]
Overture supports a high-level C++ interface
But is built upon mainly Fortran kernels.

\begin{equation}
\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} = \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)
\end{equation}

```
CompositeGrid cg;  // create a composite grid
getFromADatabaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg);  // create a grid function
u=1.;
CompositeGridOperators op(cg);  // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
    u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) );  // forward Euler
    t+=dt;
    u.interpolate();
    u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
    u.finishBoundaryConditions();
}
```
Overture is used by research groups worldwide

- Blood flow in veins with blood clot filters. (Mike Singer, LLNL).
- Pitching airfoils and micro-air vehicles (Yongsheng Lian, U. of Louisville).
- Compressible flow/ice-formation (Graeme Leese, U. Cambridge).
- Tear films and droplets (Rich Braun U. Delaware, Kara Maki UMN).
- High-order accurate subsonic/transonic aero-acoustics (Phillipe Lafon, CNRS, EDF, France).
- Low Reynolds flow for pitching airfoils (D. Chandar, R. Yapalparvi, M. Damodaran, NTU, Singapore).
- Incompressible flow in pumps (J.P. Potanza, Shell Oil, Houston).
- High-order accurate, compact Hermite-Taylor schemes (Tom Hagstrom, SMU, Dallas).
Cgins: incompressible Navier-Stokes solver.

- 2nd-order and 4th-order accurate (DNS).
- support for moving rigid-bodies (not parallel yet).
- heat transfer (Boussinesq approximation).
- semi-implicit (time accurate), pseudo steady-state (efficient line solver), full implicit.

Incompressible Navier-Stokes.

Split-step, velocity-pressure formulation:

\[
\begin{align*}
    u_t + (u \cdot \nabla)u + \nabla p - \nu \Delta u - f &= 0, & t > 0, & \mathbf{x} \in \Omega, \\
    \nabla \cdot u &= 0 & \Delta p - \nabla u : \nabla u - \alpha \nabla \cdot u - \nabla \cdot f &= 0, & t > 0, & \mathbf{x} \in \Omega,
\end{align*}
\]

Divergence damping term: \( \alpha \nabla \cdot u \) is important.

Wall boundary conditions:

\[
\begin{align*}
    u &= 0, & \nabla \cdot u &= 0, & (\text{pressure BC}) & \mathbf{x} \in \partial \Omega,
\end{align*}
\]

with numerical boundary condition:

\[
    p_n = -\mathbf{n} \cdot ( \nu \nabla \times \nabla \times u ).
\]

Use \( \nabla \times \nabla \times u \) instead of \( \Delta u \) for implicit time-stepping.

Flow past a blood-clot filter using cgins

Overlapping grid for the filter

Spherical clot trapped in the filter

Trap-ease wire filter

Cone shaped clot

Spherical clot trapped near the front

Cgcns: compressible N-S and reactive-Euler.

- reactive and non-reactive Euler equations, Don Schwendeman (RPI).
- compressible Navier-Stokes.
- multi-fluid formulation, Jeff Banks (LLNL).
- adaptive mesh refinement and moving grids.

Moving overlapping grids and AMR

A shock hitting a collection of cylinders (density).
Notes: cgcn, reactive-Euler: one refinement level, factor 4, 4930 time steps, 48 processors, from 5 to 682 grids, 100M pts (max) (eff. resolution 400 M).
Estimating Convergence Rates

Define the volume-weighted discrete $L_p$-norm of a grid function $U_i$ as

$$
\|U_i\|_p = \left( \frac{\sum_i |U_i|^p d\nu_i}{\sum_i d\nu_i} \right)^{1/p},
$$

where $d\nu_i = \left| \frac{\partial x}{\partial r} \right|_i dr_1 dr_2 dr_3$.

Assume the discrete solution $U_i^m$ at grid spacing $h_m$ satisfies

$$U_i^m - u(x_i^m, t) \approx C_i^m h_i^\mu,$$

The difference between resolution $h_n$ and $h_m$ is

$$\|U_i^m - R_n^m U_i^n\|_p \approx C |h_i^\mu - h_n^\mu|,$$

where $R_n^m$ is a fine to coarse restriction operator.

Result: Given three solutions we can estimate the convergence rate $\mu$ and the error.
Detonation in a T-Pipe

<table>
<thead>
<tr>
<th>$h_m$</th>
<th>$\mathcal{E}_1^m$</th>
<th>$\mathcal{E}_2^m$</th>
<th>$\mathcal{E}_1^m$</th>
<th>$\mathcal{E}_2^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/120</td>
<td>4.0e−3</td>
<td>3.0e−2</td>
<td>3.8e−2</td>
<td>2.6e−1</td>
</tr>
<tr>
<td>1/160</td>
<td>2.2e−3</td>
<td>1.6e−2</td>
<td>2.4e−2</td>
<td>1.9e−1</td>
</tr>
<tr>
<td>1/240</td>
<td>9.8e−4</td>
<td>7.1e−3</td>
<td>1.2e−2</td>
<td>1.2e−1</td>
</tr>
</tbody>
</table>

Estimated $L_1$ and $L_2$ errors in the density, $\mathcal{E}_1^m$ and $\mathcal{E}_2^m$, respectively, and convergence rates $\mu$ at $t = 2.0$ and $t = 2.8$. 
Cgmx: electromagnetics solver.

- a time-domain finite difference scheme.
- fourth-order accurate, 2D, 3D.
- Efficient time-stepping with the modified-equation approach
- High-order accurate symmetric difference approximations.
- High-order-accurate centered boundary and interface conditions.

Maxwell’s equations are solved in second-order form

Maxwell’s equations:

\[
\begin{align*}
\epsilon \mu \frac{\partial^2}{\partial t^2} \mathbf{E} &= \Delta \mathbf{E} + \nabla \left( \nabla \ln \epsilon \cdot \mathbf{E} \right) + \nabla \ln \mu \times \left( \nabla \times \mathbf{E} \right) - \mu \frac{\partial}{\partial t} \mathbf{j} \\
\epsilon \mu \frac{\partial^2}{\partial t^2} \mathbf{H} &= \Delta \mathbf{H} + \nabla \left( \nabla \ln \mu \cdot \mathbf{H} \right) + \nabla \ln \epsilon \times \left( \nabla \times \mathbf{H} \right) + \epsilon \nabla \times \left( \frac{1}{\epsilon} \mathbf{j} \right)
\end{align*}
\]

Advantages of the second-order form:

- No need for a staggered grid since the operator \( \Delta \) is elliptic.
- One can solve for \( \mathbf{E} \) alone.
Modified Equation time stepping

Taylor series in time:

\[
\frac{u(t + \Delta t) - 2u(t) + u(t - \Delta t)}{\Delta t^2} = u_{tt} + \frac{\Delta t^2}{12} u_{tttt} + O(\Delta t^4)
\]

For the wave equation

\[u_{tt} = \Delta u\]

a fourth-order scheme in space and time is

\[
\frac{U_{i}^{n+1} - 2U_{i}^{n} + U_{i}^{n-1}}{\Delta t^2} = \Delta_4 h U_{i}^{n} + \frac{\Delta t^2}{12} (\Delta^2)_{2h} U_{i}^{n}
\]

This scheme is very efficient (especially on Cartesian grids) and allows a large (cfl=1) time step.
Centered numerical boundary conditions for high-order approximations

Vector wave equation on a square

\[ \mathbf{E}_{tt} = \mathbf{E}_{xx} + \mathbf{E}_{yy} \quad \mathbf{x} \in \Omega = [0, 1]^2 \]

PEC (perfect electrical conductor) boundary at \( x = 0 \):

\[ E^y(0, y, t) = 0 \quad \text{(from} n \times \mathbf{E} = 0), \]
\[ \partial_x E^x(0, y, t) = 0 \quad \text{(from} \nabla \cdot \mathbf{E} = 0). \]

Taking time derivatives of the above and using the equations:

\[ \partial_x^{2m} E^y(0, y, t) = 0 \quad m = 0, 1, 2, 3, \ldots \]
\[ \partial_x^{2m+1} E^x(0, y, t) = 0 \quad m = 0, 1, 2, 3, \ldots \]

These centered conditions are used on the boundary instead of one-sided approximations.
Scattering of a plane wave by a sphere

Maximum errors at $t = 3$.
The finest grid has 6.5 million grid points.

<table>
<thead>
<tr>
<th>grid</th>
<th>N</th>
<th>$|e^E_x|_\infty$</th>
<th>$|e^E_y|_\infty$</th>
<th>$|e^E_z|_\infty$</th>
<th>$|\nabla \cdot E|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sib1</td>
<td>40</td>
<td>$1.1e^{-2}$</td>
<td>$7.9e^{-3}$</td>
<td>$5.6e^{-3}$</td>
<td>$4.0e^{-3}$</td>
</tr>
<tr>
<td>sib2</td>
<td>80</td>
<td>$8.1e^{-4}$</td>
<td>$5.6e^{-4}$</td>
<td>$4.0e^{-4}$</td>
<td>$4.2e^{-4}$</td>
</tr>
<tr>
<td>sib4</td>
<td>160</td>
<td>$5.4e^{-5}$</td>
<td>$3.7e^{-5}$</td>
<td>$2.7e^{-5}$</td>
<td>$5.4e^{-5}$</td>
</tr>
<tr>
<td>rate</td>
<td>3.84</td>
<td>3.87</td>
<td>3.86</td>
<td>3.10</td>
<td></td>
</tr>
</tbody>
</table>
Scattering of a plane wave by a dielectric cylinder

Known solution as a Mie series. Maximum errors at $t = 1$. 

| grid | $|e^{E_x}|_{\infty}$ | $|e^{E_y}|_{\infty}$ | $|e^{H_z}|_{\infty}$ | $\delta_{E}$ |
|------|---------------------|---------------------|---------------------|-------------|
| $G_1$ | 1.4e−1             | 2.9e−1             | 3.0e−1             | 6.7e−2     |
| $G_2$ | 1.0e−2             | 2.1e−2             | 2.2e−2             | 4.5e−3     |
| $G_4$ | 6.8e−4             | 1.4e−3             | 1.4e−3             | 2.9e−4     |
| rate $\sigma$ | 3.86         | 3.87            | 3.88             | 3.92       |
Scattering by a 3d material interface

- Uses newly developed 4th-order accurate 3D material interface approximations.
- Scattering of a plane wave by an interface with a bump, glass-to-air.
- 1 billion grid points, 32 nodes (8 processors per node) of a Linux cluster.
Cgsm: a solid-mechanics solver (in Overture.v24).

- linear elasticity on overlapping grids, with adaptive mesh refinement,
- conservative finite difference scheme for the second-order system,
- upwind Godunov scheme for the first-order-system.


**Conjugate heat transfer**: coupling incompressible flow to heat conduction in solids.

- overlapping grids for each fluid or solid domain,
- a partitioned solution algorithm (separate physics solvers in each sub-domain),
- (cgins) incompressible Navier-Stokes equations (with Boussinesq approximation) for fluid domains,
- (cgad) heat equation for solid domains,
- a key issue is interface coupling.

Each fluid or solid sub-domain is covered by an overlapping grid.
Fluid sub-domains: cgins. Solid sub-domains: cgad.
Coupled problem: cgmp.
Deforming composite grids for FSI

**Goal:** Couple overlapping grid techniques for modeling fluids and gases (using moving grids and AMR) with linear and non-linear solid mechanics codes.

**Approach:**

- Fluids: Overlapping grid fluid-mechanics solver.
- Solids: unstructured grid or overlapping-grid solid-mechanics solver.
- Boundary fitted deforming grids are used at the fluid-solid interfaces.

**Strengths of the approach:**

- Maintains high quality grids for large deformations and displacements.
- Uses efficient structured grid methods optimized for Cartesian grids.

**Current status:**

- Solve Euler equations in the fluid domains on moving grids.
- Solve equations of linear elasticity in the solid domains.
- Fluid grids at the interface deform over time.
Deforming composite grids for FSI

Shock focusing in a fluid cavity surrounded by an elastic solid.

- Solving the Euler equations in the fluid, linear elasticity in the solid.

The figures show results from preliminary work to model an experiment by Veronica Eliasson.
Conclusions

- Overlapping grids have been used to solve a wide class of problems.
- Smooth boundary fitted grids for accuracy.
- Structured grids for efficiency.
- Rapid grid generation for moving geometry.
- Overture is a toolkit for grid generation and solving PDEs.
- The CG set of PDE solvers solve a variety equation in continuum mechanics.

Open problem: automatic grid generation for complex geometry.