Introduction to the Overture Demo

Bill Henshaw

Department of Mathematical Sciences,
Rensselaer Polytechnic Institute,
Troy, New York, USA.

13th Symposium on Overset Grids and Solution Technology,
Future of Flight Aviation Center, Mukilteo, Washington, October 17–20, 2016.

Downloading Overture and the CG (Composite Grid) suite of PDE solvers.

The Overture framework and Composite Grid (CG) PDE solvers are open source and available from (documentation, downloads)

overtureFramework.org

The source-code repositories are hosted at

sourceforge.net/projects/overtureframework

Acknowledgments.

Supported by

Department of Energy, Office of Science

ASCR Applied Math Program

National Science Foundation

Current Overture developers

Kyle Chand

Bill Henshaw

Major Contributors

Don Schwendeman (RPI),

Jeff Banks (RPI),

Longfei Li (RPI)

Overture: a toolkit for solving partial differential equations (PDEs) on overlapping grids.

Top three reasons for using Overture:

1. You need to efficiently solve a PDE on a complex geometry.
2. You need to solve a PDE on a moving geometry.
3. You need to generate an overlapping grid.

You can

- write your own PDE solver using the capabilities provided by Overture.
- use (or change) an existing PDE solver from the CG suite.

Overture: a toolkit for solving partial differential equations (PDEs) on overlapping grids.

Top three reasons for using Overture:

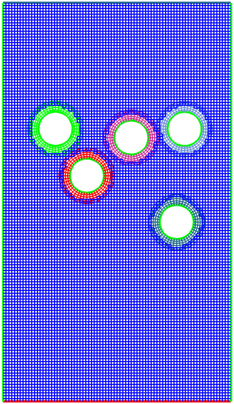
1. You need to efficiently solve a PDE on a complex geometry.
2. You need to solve a PDE on a moving geometry.
3. You need to generate an overlapping grid.

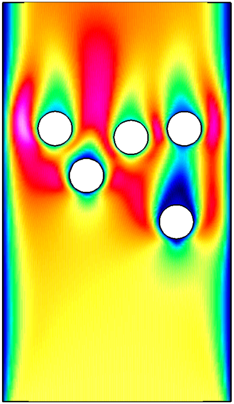
You can

- write your own PDE solver using the capabilities provided by Overture.
- use (or change) an existing PDE solver from the CG suite.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

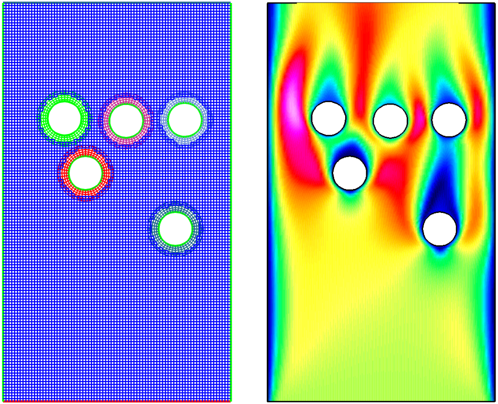




- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

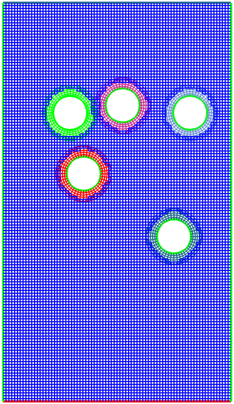
Overlapping grid: a set of structured grids that overlap.

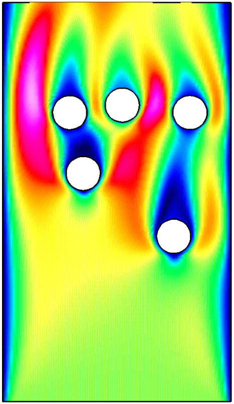


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

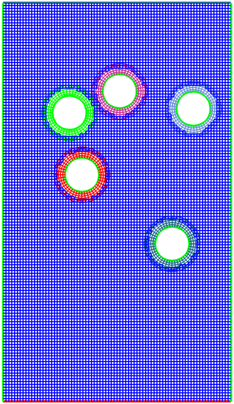


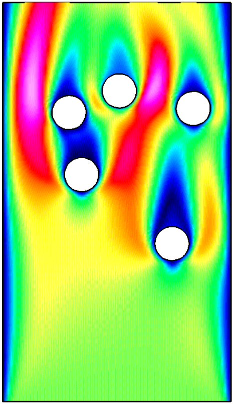


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

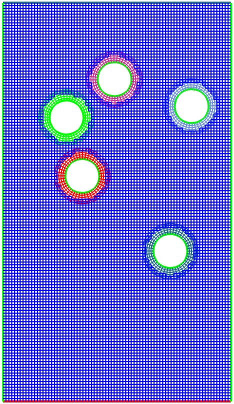


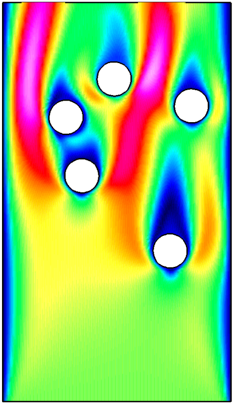


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

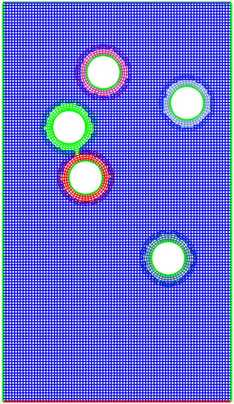


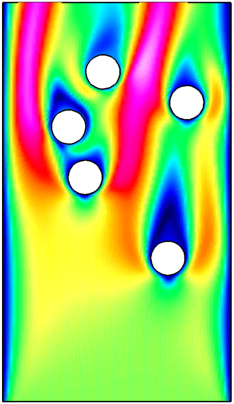


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

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.

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- 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.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

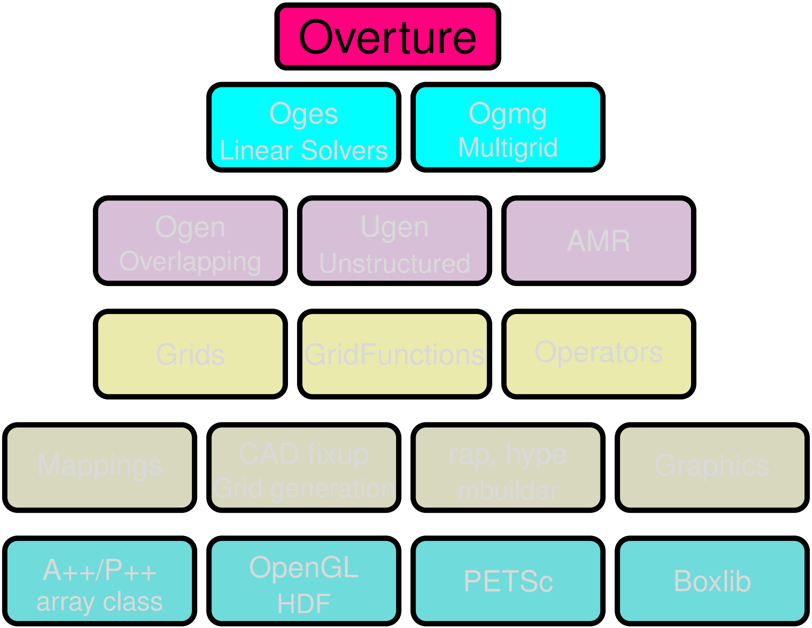
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive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

Key Features of Overture

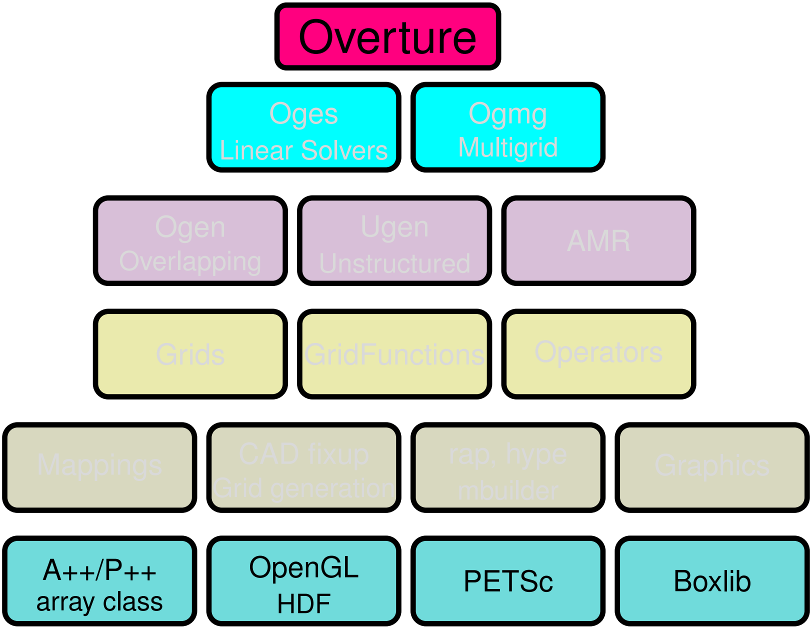
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive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

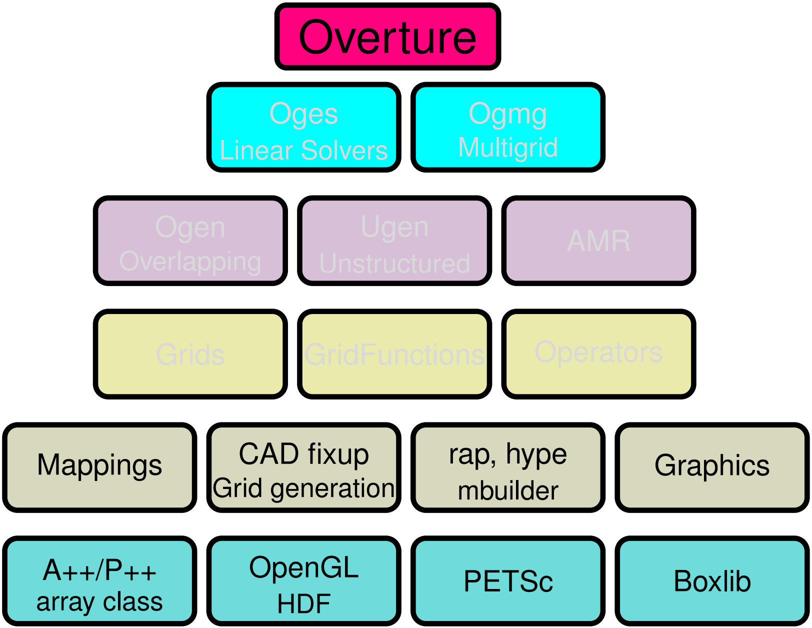
Key Features of Overture

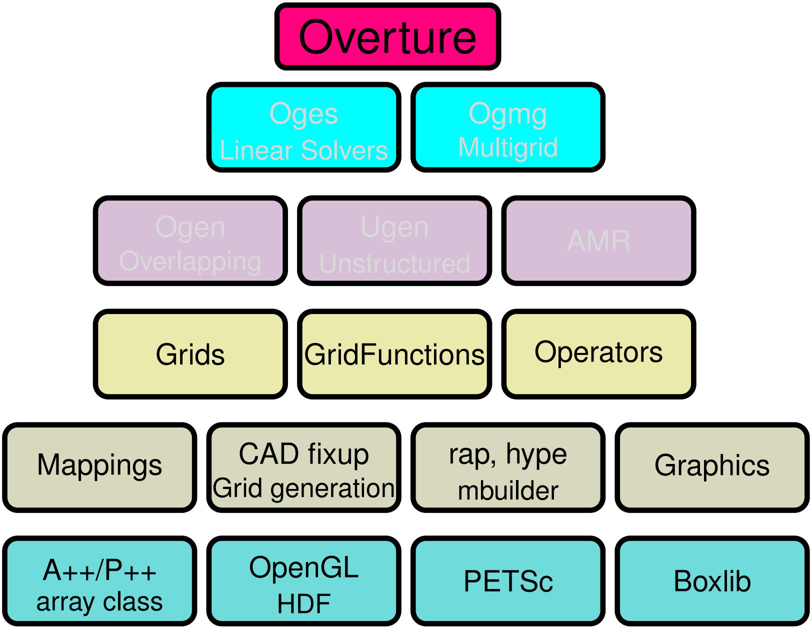
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive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

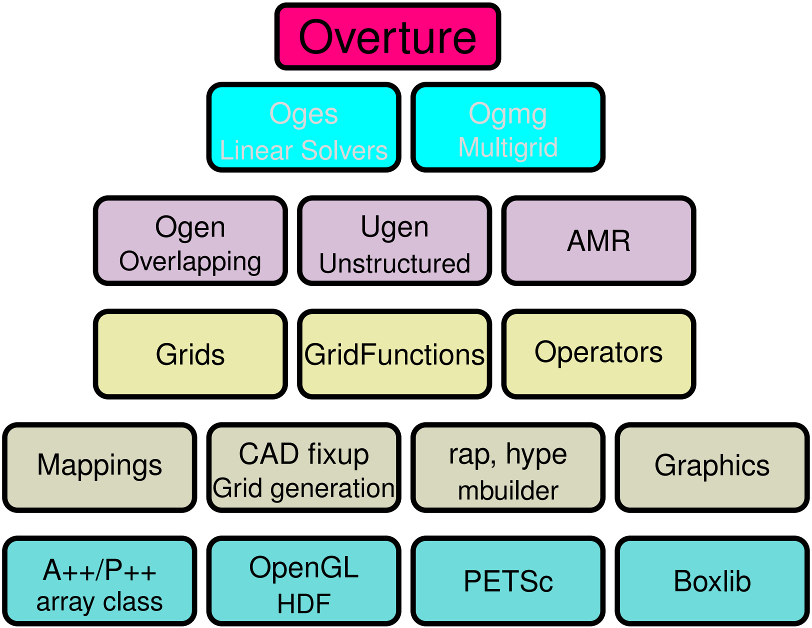


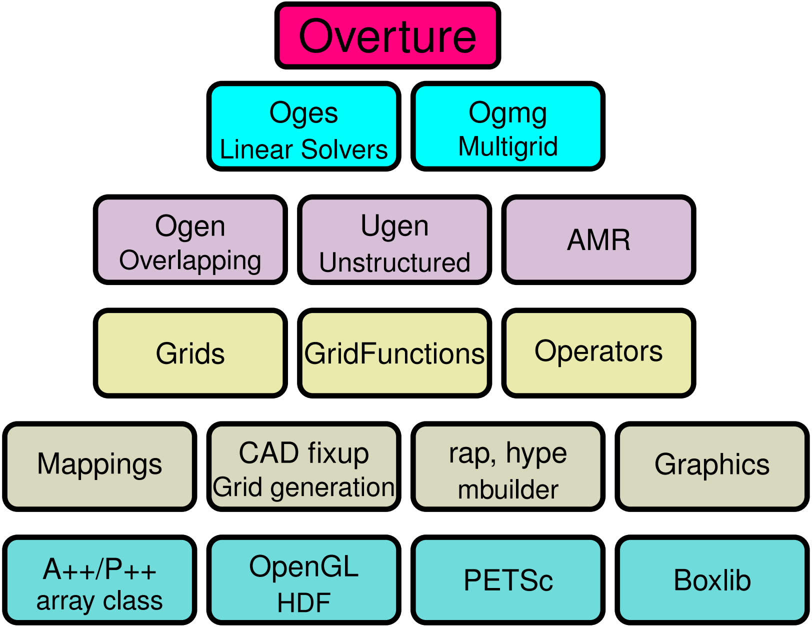
Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

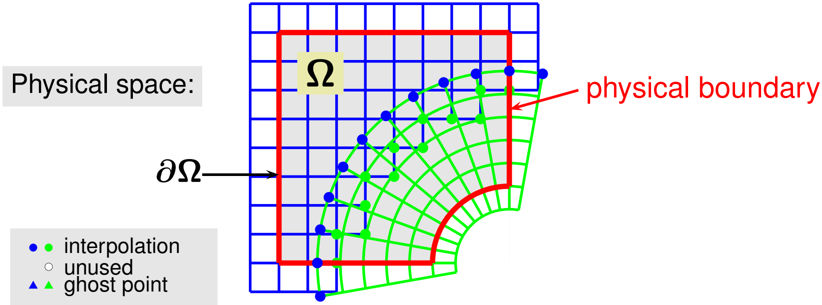
- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

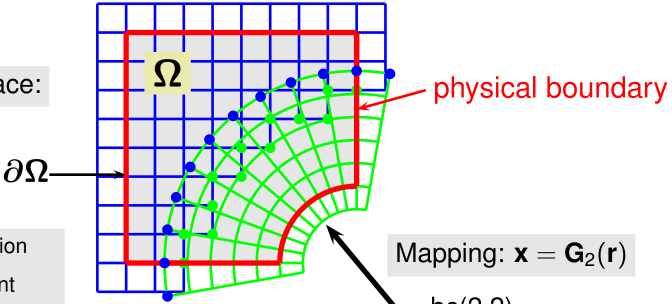
- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgsm: elastic wave equation (linear elasticity).
- cgmx: time domain Maxwell's equations solver.
- cgmp: multi-physics solver (e.g. FSI, conjugate heat transfer).

Components of an Overlapping Grid



Components of an Overlapping Grid

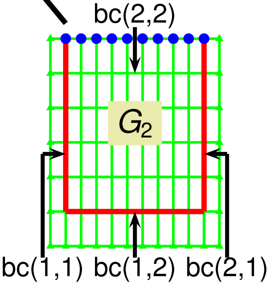
Physical space:



interpolation
unused
ghost point

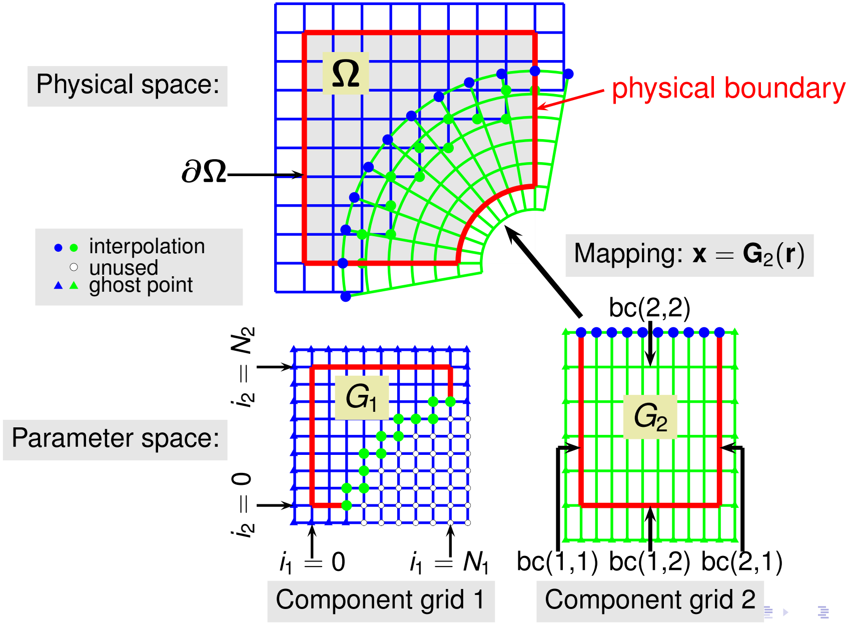
Mapping: $\mathbf{x} = \mathbf{G}_2(\mathbf{r})$

Parameter space:

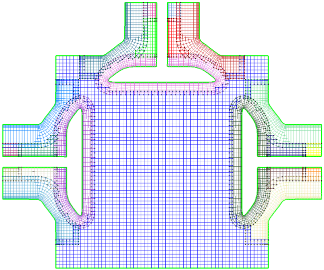


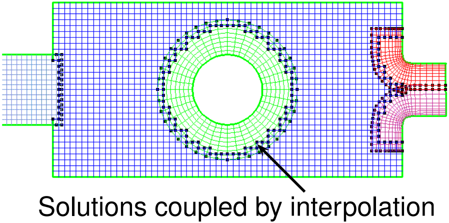
Component grid 2

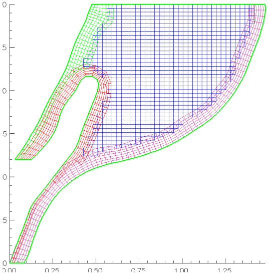
Components of an Overlapping Grid



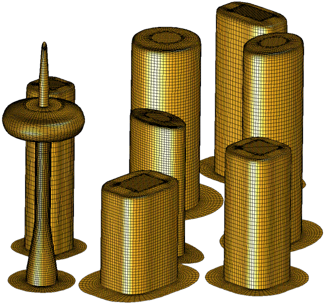
Ogen can be used to build 2D overlapping grids:

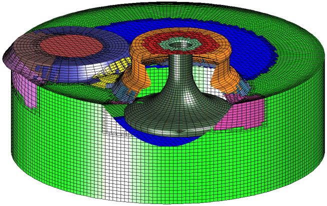


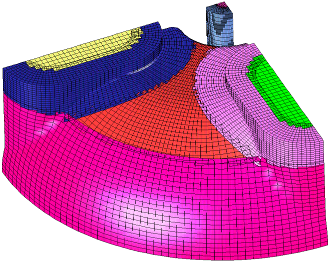


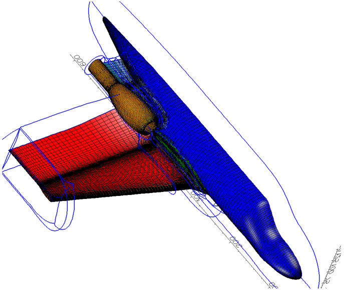


Ogen can be used to build 3D overlapping grids:









Overture supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
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 supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
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 supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
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 supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
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 supports a high-level C++ interface

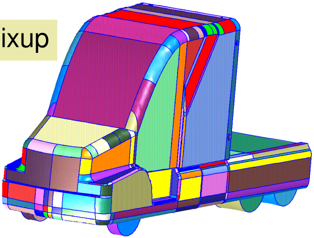
But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
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();
}
```

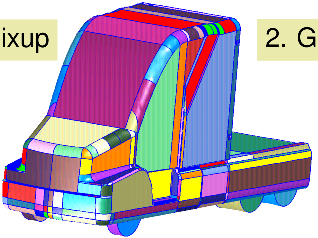
From CAD to Mesh to Solution with Overture

1. Cad fixup

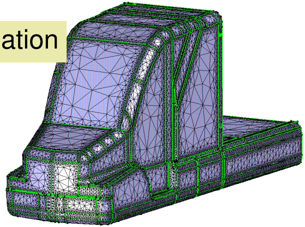


From CAD to Mesh to Solution with Overture

1. Cad fixup

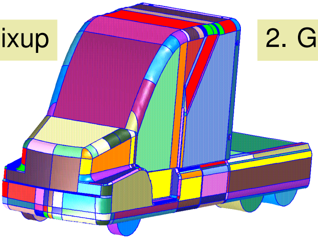


2. Global triangulation

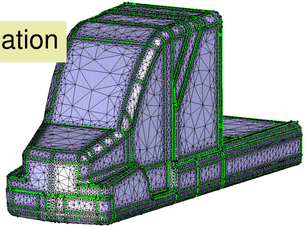


From CAD to Mesh to Solution with Overture

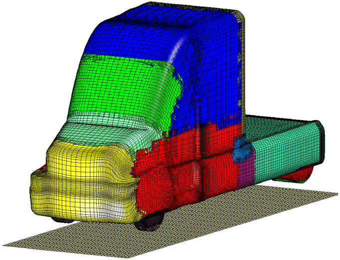
1. Cad fixup



2. Global triangulation

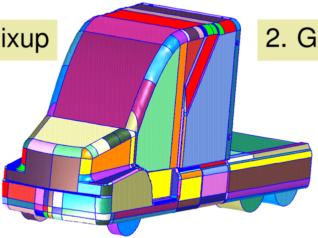


3. Overlapping grid

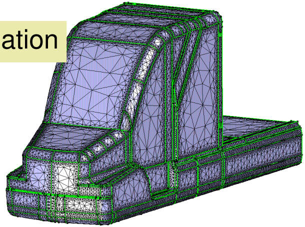


From CAD to Mesh to Solution with Overture

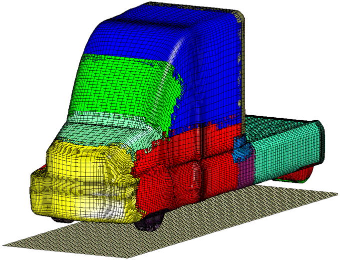
1. Cad fixup



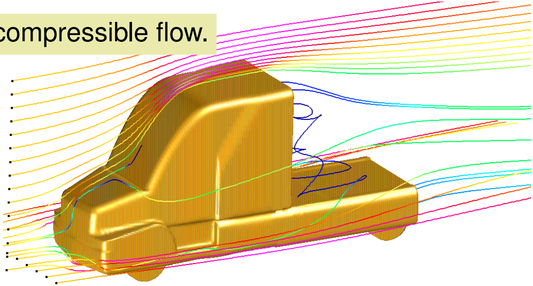
2. Global triangulation



3. Overlapping grid



4. Incompressible flow.

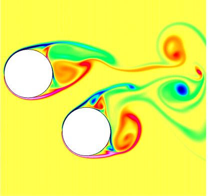


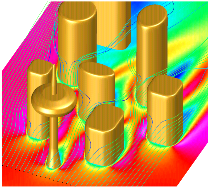
Overture is used by research groups worldwide

Typical users are graduate students and University/Lab researchers.

- Blood flow and blood clot filters (Dr. Mike Singer).
- Flapping airfoils, mircro-air vehicles (Prof. Yongsheng Lian, U. of Louisville).
- Wave-energy devices (Dr. Robert Read, Prof. Harry Bingham, Technical U. of Denmark).
- Plasma physics (Dr. Jeff Banks, Dr. Richard Berger, LLNL).
- Flapping airfoils (Dr. Joel Guerrero, U. of Genoa).
- High-order accurate subsonic/transonic aero-acoustics (Dr. Philippe Lafon, CNRS, EDF).
- Tear films and droplets (Dr. Kara Maki, RIT, and Prof. Richard Braun, U. Delaware).
- Elastic wave equation (Dr. Daniel Appelö, UNM).
- Compressible flow/ice-formation (Graeme Leese, Prof. Nikos Nikiforakis, U. Cambridge).
- Relativistic hydrodynamics and Einstein field equations (Dr. Philip Blakely, U. Cambridge).
- Converging shock waves, shock focusing (Prof. Veronica Eliasson, USC).
- Wind farms (Dr. J. Sitaraman, U. of Wyoming).
- Hypersonic flows for reentry vehicles, (Dr. Bjorn Sjögreen, LLNL, Dr. Helen Yee NASA).
- High-order accurate, compact Hermite-Taylor schemes (Prof. Tom Hagstrom, SMU).
- High-order accurate aero-acoustics (Dr. Ramesh Balakrishnan, ANL).
- Incompressible flow in pumps (Dr. J.P. Potanza, Shell Oil).

Cgins: incompressible Navier-Stokes solver.

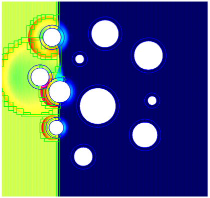


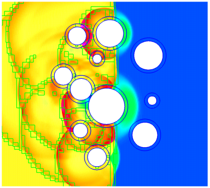


- 2nd-order and 4th-order accurate (DNS).
- support for moving rigid-bodies (deforming bodies in progress).
- heat transfer (Boussinesq approximation).
- time-stepping options: compact factored-scheme (AFS), semi-implicit (time accurate) (IM), predictor-corrector (PC), pseudo steady-state (efficient line solver), full implicit.

- WDH., *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids*, J. Comput. Phys, **113**, no. 1, (1994) 13–25.

Cgcns: compressible N-S and reactive-Euler.

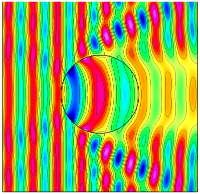


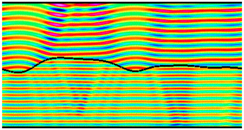


- reactive and non-reactive Euler equations.
- compressible multiphase flow (BN type model).
- multi-fluid flow.
- compressible Navier-Stokes.
- adaptive mesh refinement and moving grids.

• WDH., D. W. Schwendeman, *Parallel Computation of Three-Dimensional Flows using Overlapping Grids with Adaptive Mesh Refinement*, J. Comp. Phys. **227** (2008).
• WDH., DWS, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005).
• WDH., DWS, An adaptive numerical scheme for high-speed reactive flow on overlapping grids, J. Comp. Phys. **191** (2003).

Cgmx: electromagnetics solver.





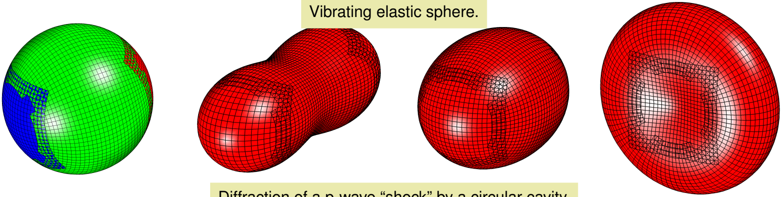
- fourth-order accurate, 2D, 3D.
- Efficient time-stepping with the modified-equation approach
- High-order accurate symmetric difference approximations.
- High-order-accurate *centered* boundary and interface conditions.

• WDH., *A High-Order Accurate Parallel Solver for Maxwell's Equations on Overlapping Grids*, SIAM J. Scientific Computing, **28**, no. 5, (2006).

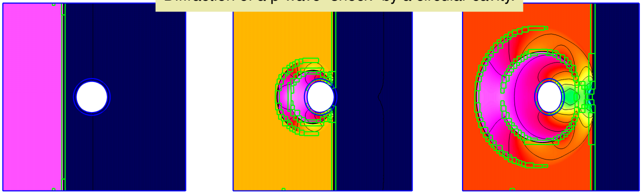
Cgsm: solve the elastic wave equation.

- 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.

Vibrating elastic sphere.



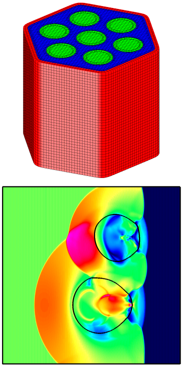
Diffraction of a p-wave “shock” by a circular cavity.



- D. Appelö, J.W. Banks , WDH, D.W. Schwendeman, *Numerical Methods for Solid Mechanics on Overlapping Grids: Linear Elasticity*, JCP, 2012.

Cgmp: a multi-domain multi-physics solver.

Cgmp couples different fluids and solid solvers



- overlapping grids for each fluid or solid domain,
- a partitioned solution algorithm (separate physics solvers in each sub-domain),
- accurate and stable interface treatments.
- conjugate heat transfer (cgins+cgad, cgcns+cgad).
- fluid-structure interactions (cgcns + cgsm). Current work: incompressible flow (cgins) + beams and bulk solids.

• J. W. Banks, WDH, D.W. Schwendeman, *Deforming Composite Grids for Solving Fluid Structure Problems*, J. Comput. Phys, 2012.
•WDH., K. K. Chand, *A Composite Grid Solver for Conjugate Heat Transfer in Fluid-Structure Systems*, J. Comput. Phys, 2009.

Summary.

- Overture: a toolkit for solving PDEs on overlapping grids.
- CG : a suite of PDE solvers for overlapping grids.

`overtureFramework.org`