On Multigrid for Overlapping Grids

William D. Henshaw

Centre for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA, 94551
henshaw@llnl.gov
http://www.llnl.gov/casc/people/henshaw
http://www.llnl.gov/casc/Overture

May 20, 2011

UCRL-MA-??????

Abstract: We describe various aspects of solving elliptic boundary value problems on overlapping grids. We describe the Overlapping-Grid-MultiGrid-solver, Ogmg, that can be used to obtain solutions to elliptic boundary value problems. Ogmg solves problems in two and three space dimensions on composite overlapping grids. Second and fourth-order accurate approximations are supported. Given an overlapping grid generated from the Ogen grid generator, Ogmg will generate the coarse grid multigrid levels using an automatic coarsening algorithm. The equations on the coarse grids can be determined automatically using a Galerkin averaging procedure. The multigrid solution algorithm has been optimised for some commonly occuring problems such as equations defined with the Laplace operator. Smoothers include Red-Black, Jacobi, Gauss-Seidel, line-zebra and line-Jacobi. Ogmg is particularly efficient when a majority of the grid points belong to cartesian component grids; this is often the case when grids become sufficiently fine. The fourth-order accurate approximations are solved directly with multigrid (as opposed to using a defect correction scheme). Convergence rates for the fourth-order approximations are often nearly as good as the convergence rates for second-order discretizations. Currently only scalar elliptic boundary value problems can be solved.

Contents

1 Introduction **3**

2 The multigrid algorithm for overlapping grids **4**

3 Black box multigrid features **4**

3.1 Automatic coarsening algorithms 5

3.2 Operator averaging 5

3.2.1 Operator averaging at boundaries 5

4 Smoothing overlapping grids **8**

1 Introduction

Ogmg is a multigrid solver for use with Overture [2],[3]. Ogmg can solve scalar elliptic problems on overlapping grids. It has a variety of smoothers including Red-Black, Jacobi, Gauss-Seidel and line smoothers. Second-order accurate and fourth-order accurate approximations are supported. A sparse direct or sparse iterative solver such as GMRES can be used to solve the coarse grid equations – the Overture solver Oges is used for this purpose, thus allowing access to a variety of sparse matrix solvers such as those available with PETSc[1].

In the case of a general elliptic boundary value problem, the system of equations that Ogmg solves are specified as a "coefficient-array" grid function. The coefficient array can be created using the Overture operator classes.

Ogmg has been specifically optimised for a class of commonly occuring problems. These *predefined* equations are

$$\begin{aligned} \Delta u &= f && \text{laplaceEquation} \\ \nabla\cdot(s(\mathbf{x})\nabla)u &= f && \text{divScalarGradOperator} \\ (I + c_0\Delta)u &= f && \text{heatEquationOperator} \\ (I + s(x)\Delta)u &= f && \text{variableHeatEquationOperator} \\ (I + \nabla\cdot(s(\mathbf{x})\nabla))u &= f && \text{divScalarGradHeatEquationOperator} \end{aligned}$$

These equations are augmented with Dirichlet, Neumann or mixed boundary conditions. The above equations can be solved more quickly and with less storage than if the same equation had been defined through a general coefficient matrix.

Ogmg starts with an overlapping grid constructured with the overlapping grid generator Ogen. The coarse grids needed by the multigrid algorithm are automatically generated with a new coarsening algorithm. Very coarse grids can be formed by relaxing the accuracy requirements for interpolation on the coarser grids and allowing the overlap between grids to grow as the grids are coarsened.

The discrete approximations to the equations on the coarse grids are determined automatically using a Galerkin averaging procedure from finer grids.

An adaptive smoothing algorithm is used to improve the convergence rate. The number of sub-smooths performed on each component grid is adjusted to keep the residuals nearly the same.

Good multigrid convergence rates are usually obtained. Ogmg is particularly efficient when a majority of the grid points belong to cartesian component grids; this is often the case when grids become sufficiently fine. The overall convergence rates has been significanty improved from the original fortran version of the code [4]. Some of the factors that led to this improved performance were

- an adaptive smoothing algorithm that performs additional sub-smooths on component grids that are converging slowly.
- generating the coarse grid equations through operator averaging improves the convergence rate.
- careful attention to boundary conditions (especially for Neumann boundary conditions and fourth-order accurate discretzations) can improve the convergence rate.
- using over-relaxation for Red-Black smoothers is very helpful, especially in 3D.

Ogmg does not yet efficiently handle the case of singular problems such as a Poisson equation with all Neumann boundary conditions; in this case one must first determine the left null vector to the discrete operator.

2 The multigrid algorithm for overlapping grids

Ogmg uses the standard defect correction algorithm. The implementation on overlapping grids is relatively straight-forward. See the paper [4] for further discussion.

- Typically Jacobi, red-black or line (zebra) smoothers are used.
- The fine to coarse *Restriction* operator is the *full weighting* operator except at boundaries.
- The coarse to fine Prolongation operator is second or fourth order interpolation; second-order by default.
- The **cycle** chosen is either adaptive or can be fixed to a desired one.

while *not converged* **do**
  smooth ν_1 times or until the smoothing rate $> \eta$
    $v_1 \leftarrow S^{\nu_1} v_1$
  form the defect and transfer to the coarser grid
    $f_2 \leftarrow R^{1\to 2}(f - Av_1)$
  "solve" the defect equation (at least to an "accuracy" of δ)
    $A_2 v_2 \approx f_2$
  correct the fine grid solution from the coarse grid solution
    $v_1 \leftarrow v_1 + P^{2\to 1} v_2$
  smooth ν_2 times or until the smoothing rate $> \eta$
    $v_1 \leftarrow S^{\nu_2} v_1$
end while

The **smoothing step** represented by the operator S is a *composite-smooth* where each grid in turn is smoothed:

for *each grid g in a CompositeGrid* **do**
  smooth grid g ν_g times
  interpolate
end for

The smoother and the number of smooths may vary from component grid to component grid. We try to choose n_g, the number of smooths on each component grid, so that the residual stays about the same size on each component grid. The approximate rule we use is that

$$n_g \approx \frac{\| \text{ residual on grid } g\|}{\min_g \|\text{residual on grid } g\|}$$

The grid with the smallest residual will have $n_g = 1$.

3 Black box multigrid features

A new feature of Ogmg is the ability to take a problem defined on a CompositeGrid with only one level and to automatically generate the information for coarser levels. There are two key ingredients to making this work. The first is that we need to generate the coarse grid coefficient matrices automatically. The second is to handle interpolation points on the coarse grid which may or may not sit on interpolation points on the fine grid.

3.1 Automatic coarsening algorithms

3.2 Operator averaging

To generate a coarse grid operator from a fine grid operator we can average the operator on the fine grid and then restrict the result to the coarse grid.

Consider a 3 point stencil operator in one dimension. If we look at the stencil for rows $i-1, i, i+1$ arranged in a matrix then we get

$$\begin{array}{ccccc} a_{i-1}u_{i-2} & b_{i-1}u_{i-1} & c_{i-1}u_i & 0 & 0 \\ 0 & a_iu_{i-1} & b_iu_i & c_iu_{i+1} & 0 \\ 0 & 0 & a_{i+1}u_i & b_{i+1}u_{i+1} & c_{i+1}u_{i+2} \end{array}$$

If we replace row i by the weighted average of rows $i-1, i, i+1$ with weights α, β, α then we get the wide stencil

$$\alpha a_{i-1}u_{i-2} \quad (\alpha b_{i-1} + \beta a_i)u_{i-1} \quad (\alpha(c_{i-1} + a_{i+1}) + \beta b_i)u_i \quad (\alpha b_{i+1} + \beta c_i)u_{i-1} \quad \alpha c_{i+1}u_{i+2}$$

If we distribute the values at point $i-1$ using $u_{i-1} = \frac{1}{2}(u_{i-2}+u_i)$ and at point $i+1$ using $u_{i+1} = \frac{1}{2}(u_{i+2}+u_i)$ then we have a wide stencil only defined at points $i-2, i, i+2$.

$$(\alpha(a_{i-1} + \tfrac{1}{2}b_{i-1}) + \tfrac{1}{2}\beta a_i)u_{i-2} \qquad (\alpha(\tfrac{1}{2}b_{i-1} + \tfrac{1}{2}b_{i+1} + c_{i-1} + a_{i+1}) + \beta(b_i + \tfrac{1}{2}a_i + \tfrac{1}{2}c_i))u_i \qquad (\alpha(c_{i+1}\tfrac{1}{2}b_{i+1}) + \tfrac{1}{2}\beta c_i)u_{i+2}$$

Typically we take $\alpha = \gamma = 1/4$, and $\beta = 1/2$. In more than one space dimension we can apply the above averaging procedure sequentially in each direction.

Now consider a 5 point operator (we could also average 5 adjacent equations rather then 3)

$$\begin{array}{ccccccc} a_{i-1}u_{i-3} & b_{i-1}u_{i-2} & c_{i-1}u_{i-1} & d_{i-1}u_i & e_{i-1}u_{i+1} & 0 & 0 \\ 0 & a_iu_{i-2} & b_iu_{i-1} & c_iu_i & d_iu_{i+1} & e_iu_{i+2} & 0 \\ 0 & 0 & a_{i+1}u_{i-1} & b_{i+1}u_i & c_{i+1}u_{i+1} & d_{i+1}u_{i+2} & e_{i+1}u_{i+3} \end{array}$$

Averaging in a similar fashion we obtain a coarse grid operator

$$a_i^c u_{i-2} \quad b_i^c u_{i-1} \quad c_i^c u_i \quad d_i^c u_{i+1} \quad e_i^c u_{i+2}$$

where

$$\begin{aligned} a_i^c &= \frac{1}{2}\alpha a_{i-1} \\ b_i^c &= \alpha b_{i-1} + \beta a_i + \frac{1}{2}\alpha a_{i-1} + \frac{1}{2}(\alpha c_{i-1} + \beta b_i + \alpha a_{i+1}) \\ c_i^c &= \alpha d_{i-1} + \beta c_i + \alpha b_{i+1} + \frac{1}{2}(\alpha c_{i-1} + \beta b_i + \alpha a_{i+1}) + \frac{1}{2}(\alpha e_{i-1} + \beta d_i + \alpha c_{i+1}) \\ d_i^c &= \alpha d_{i+1} + \beta e_i + \frac{1}{2}(\alpha e_{i-1} + \beta d_i + \alpha c_{i+1}) + \frac{1}{2}\alpha e_{i+1} \\ e_i^c &= \frac{1}{2}\alpha e_{i+1} \end{aligned}$$

3.2.1 Operator averaging at boundaries

At a boundary we will typically have a boundary condition such as a dirichlet, neumann or mixed boundary condition. We need to decide how to average near the boundary and on the boundary or ghost line.

Ogmg is aware of two types of boundary conditions. These 'boundary conditions' actually just indicate how the **ghost line** should be updated. The condition `extrapolation` indicates the ghost line is extrapolated and usually means that a dirichlet boundary condition is applied on the boundary. The `equation` boundary condition indicates that some equation is applied on the ghost line; this is usually associated with a neumann or mixed boundary condition.

dirichlet : In this case we just impose a dirichlet BC in the coarse grid operator.

neumann : (or neumann like condition) coarse grid operator. The coarse grid operator is obtained by distributing the fine grid ghost line equation to the coarser grid but not averaged in the tangential direction.

$$(\alpha(a_i + \tfrac{1}{2}b_i) + \tfrac{1}{2}\beta a_i)u_{i-2} \qquad (\alpha(b_i + c_i + a_i) + \beta(b_i + \tfrac{1}{2}a_i + \tfrac{1}{2}c_i))u_i \qquad (\alpha(c_i + \tfrac{1}{2}b_i) + \tfrac{1}{2}\beta c_i)u_{i+2}$$

extrapolation : coarse grid operator is also extrapolation.

equation : ghost line has some equation on it. The coarse grid operator is obtained by averaging the ghostline equations (i.e. averaging is only done in the tangential directions).

Remark: The coarse grid matrix A^2 can also be defined from the fine grid matrix A^1 using the prologation and restriction operators

$$A^2 = RA^1P$$

The first step above where the rows were combined corresponds to premultiplying A^1 by R. The second step where the values at points $i+1$ and $i+1$ were removed corresponds to the post-multiplication by P.

Tables (1-2) compare the convergence results for Poisson's equation with dirichlet boundary conditions when using operator averaging to construct the coarse grid equations to no operator averaging. The notation used to label the columns in defined at the start of section **??**. In this example the operator-averaged case is significantly better. There was a negligible difference in CPU times between the two cases.

With Operator Averaging				
i	res	rate	WU	ECR
1	$1.4e+02$	0.036	5.0	0.52
2	$3.4e+00$	0.024	5.0	0.48
3	$9.3e-02$	0.027	5.0	0.49
4	$2.5e-03$	0.027	5.0	0.49
5	$7.1e-05$	0.028	5.0	0.49
6	$2.0e-06$	0.029	5.0	0.49
7	$5.9e-08$	0.029	5.0	0.50

Without Operator Averaging				
i	res	rate	WU	ECR
1	$2.3e+02$	0.054	5.0	0.56
2	$9.4e+00$	0.041	5.0	0.53
3	$5.2e-01$	0.055	5.0	0.56
4	$2.9e-02$	0.056	5.0	0.57
5	$1.7e-03$	0.058	5.0	0.57
6	$1.1e-04$	0.065	5.0	0.58
7	$7.4e-06$	0.067	5.0	0.58

Table 1: Second-order accuracy. Left: operator averaging. Right: no operator averaging. Multigrid convergence rates, 5 levels, smoother rb[2,1]. Grid square256, trigonometric solution.

With Operator Averaging				
i	res	rate	WU	ECR
1	$3.0e+02$	0.048	5.1	0.55
2	$1.4e+01$	0.046	5.1	0.54
3	$7.0e-01$	0.051	5.1	0.55
4	$3.6e-02$	0.051	5.1	0.56
5	$1.9e-03$	0.052	5.1	0.56
6	$9.7e-05$	0.052	5.1	0.56
7	$5.1e-06$	0.053	5.1	0.56

Without Operator Averaging				
i	res	rate	WU	ECR
1	$9.4e+02$	0.111	5.1	0.65
2	$8.8e+01$	0.094	5.1	0.63
3	$9.8e+00$	0.111	5.1	0.65
4	$1.2e+00$	0.120	5.1	0.66
5	$1.4e-01$	0.122	5.1	0.66
6	$1.8e-02$	0.124	5.1	0.66
7	$2.2e-03$	0.126	5.1	0.66

Table 2: Fourth-order accuracy. Left: operator averaging. Right: no operator averaging. Multigrid convergence rates, 5 levels, smoother rb[2,1]. Grid square256.order4, trigonometric solution.

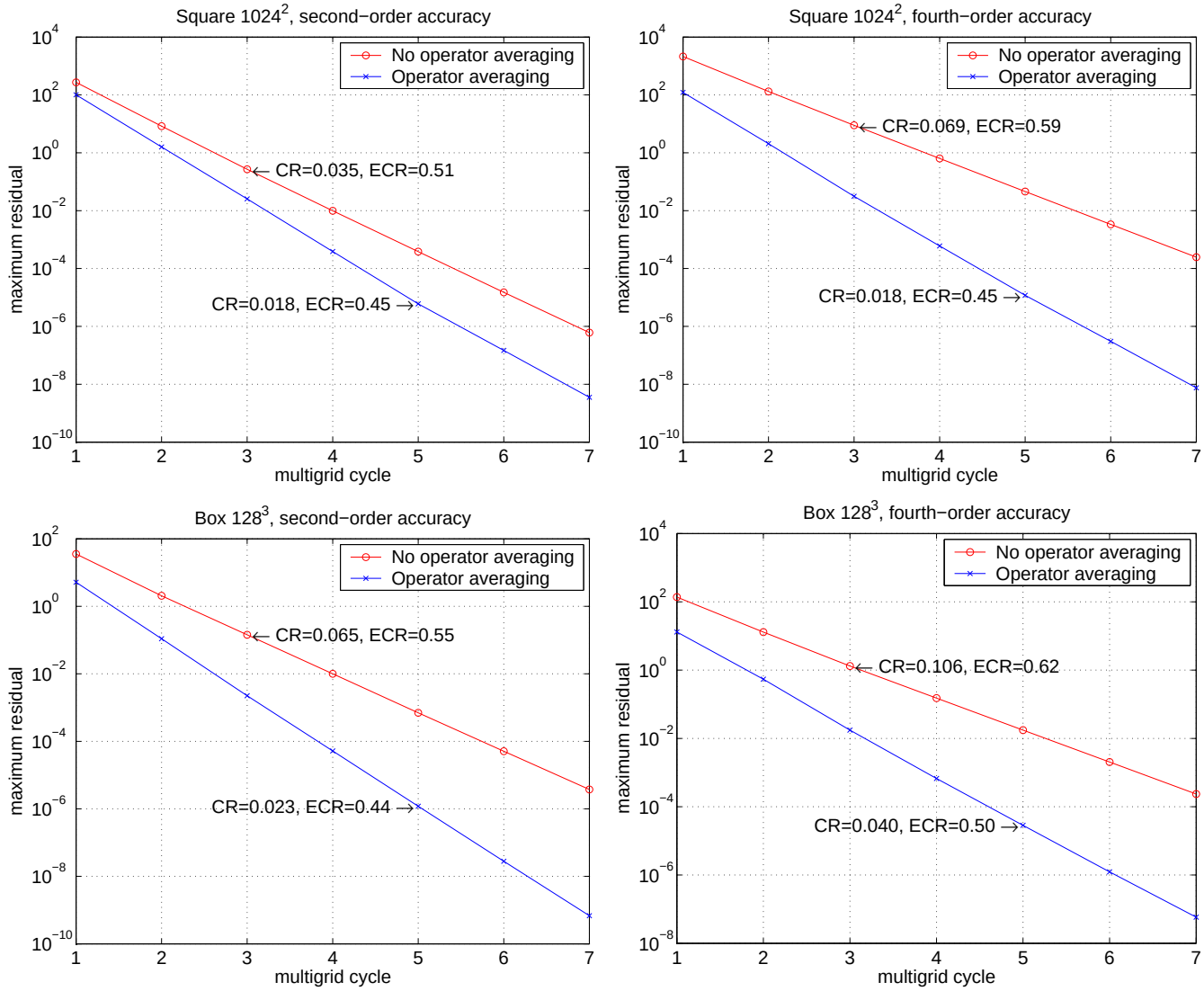

Figure 1: The convergence rate is improved when the coarse grid operators are generated with operator averaging. Results are shown for a V[2,1] cycle.

4 Smoothing overlapping grids

There are a few issues that must be addressed when smoothing an overlapping grid. Some care must be taken to ensure that the composite-smooth operator (a smoothing step over all component grids) retains similiar smoothing rates to that of a smoother for a single component grid. The underlying principle for smoothing an overlapping grid is that the defect after smoothing should be smooth enough to be represented on the next coarser levels.

References

[1] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *The portable extensible toolkit for scientific computation*, Tech. Rep. http://www.mcs.anl.gov/petsc/petsc.html, Argonne National Laboratory, 1999.

[2] W. D. Henshaw, *Overture: An object-oriented framework for solving PDEs in moving geometries on overlapping grids using C++*, in Proceedings of the Third Symposium on Overset Composite Grid and Solution Technology, 1996.

[3] ——, *Oges user guide, a solver for steady state boundary value problems on overlapping grids*, Research Report UCRL-MA-132234, Lawrence Livermore National Laboratory, 1998.

[4] W. D. Henshaw and G. S. Chesshire, *Multigrid on composite meshes*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 914–923.

Index

multigrid, 3
    overlapping grid algorithm, 4