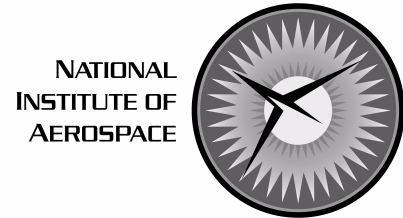
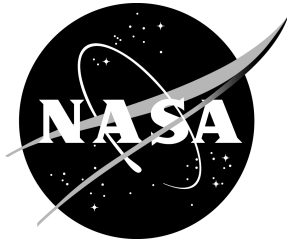


NASA/CR-2004-213031
NIA Report No. 2004-01



On Efficient Multigrid Methods for Materials Processing Flows with Small Particles

Boris Diskin
National Institute of Aerospace, Hampton, Virginia

Vasyl Michael Harik
Swales Aerospace, Hampton, Virginia

July 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

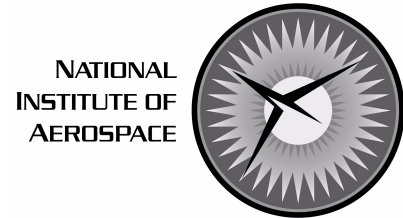
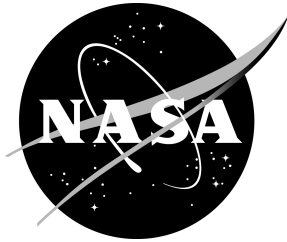
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- Email your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Telephone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/CR-2004-213031
NIA Report No. 2004-01



On Efficient Multigrid Methods for Materials Processing Flows with Small Particles

Boris Diskin

National Institute of Aerospace, Hampton, Virginia

Vasyl Michael Harik

Swales Aerospace, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NCC-1-02043

July 2004

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

ON EFFICIENT MULTIGRID METHODS FOR MATERIALS PROCESSING FLOWS WITH SMALL PARTICLES

Boris Diskin* and Vasyl Michael Harik†

ABSTRACT

Multiscale modeling of materials requires simulations of multiple levels of structural hierarchy. The computational efficiency of numerical methods becomes a critical factor for simulating large physical systems with highly disparate length scales. Multigrid methods are known for their superior efficiency in representing/resolving different levels of physical details. The efficiency is achieved by employing interactively different discretizations on different scales (grids). To assist optimization of manufacturing conditions for materials processing with numerous particles (e.g., dispersion of particles, controlling flow viscosity and clusters), a new multigrid algorithm has been developed for a case of multiscale modeling of flows with small particles that have various length scales. The optimal efficiency of the algorithm is crucial for accurate predictions of the effect of processing conditions (e.g., pressure and velocity gradients) on the local flow fields that control the formation of various microstructures or clusters.

1 INTRODUCTION

Capability to efficiently simulate materials processing flows with small particles is important for optimization and manufacturing control of advanced materials that have tailored physical properties (e.g., stiffness or strength, thermal or electric conductivity). Ability to process polymers with filler particles is essential for design of multifunctional materials since the commonly used low-cost plastics (e.g., polypropylene or polyamide) have low thermal and low electrical conductivity without any additives. These properties are critical for designing novel aerospace structures like flexible wings, structures with active controls, and multifunctional membranes. Applications such as heat sinks, radio frequency interference shielding, electromagnetic shielding and electronic packaging also require new composite materials with higher thermal and electrical conductivities.

Successful control of manufacturing conditions for materials processing may include improving dispersion of particles and reducing the cluster size, applying pressure gradient or shear strain, as well as controlling flow viscosity, microstructural texture or particle orientation. Any particular manufacturing process may include multiple stages and many processing parameters. As a result, optimization of the flow control requires accurate models in order to resolve local particle distributions, their relative motion, and interactions between small particles and the surrounding polymer matrix for different processing parameters. Moreover,

*Staff Scientist, National Institute of Aerospace (NIA), 144 Research Dr., Hampton, VA 23681 Email: bdiskin@nianet.org. This research was supported by the National Institute of Aerospace under NASA LaRC Cooperative Agreement NCC1-02043

†Senior Structural Engineer, Swales Aerospace, NASA Langley Research Center, Hampton, VA 23681 Email: v.m.harik@larc.nasa.gov. This research was supported in part by NASA URETI for bio-inspired nanostructured multifunctional materials (Award NCC1-02037)

such models are needed to predict the formation of various microstructures or clusters and to simulate the effect of processing conditions (e.g., pressure and velocity gradients or flow rates) on materials composition. Two-phase processing flows may involve highly disparate length scales caused by the distribution of particle sizes that may span a range from tens of nanometers to tens of microns (e.g., sub-micron particles or carbon filaments, conglomerates of these particles or clusters). To model the local and global transport properties of such two-phase flows, efficient hierarchical methods for multiscale modeling are required.

The processing flows of two-phase viscous materials can be often described by partial differential equations of elliptic type (e.g., Poisson equation). The processing of filled polymers involves viscous flows of suspended particles and, in some cases, flow control with an electromagnetic field applied. Because the flow is very viscous, and the filler particles are small, the dynamic effects can usually be neglected. Consequently, the viscous flows can be described by either the Stokes or Poisson equations. Electrostatic effects can be also analyzed with these equations. The smallest size of the filaments in this study can be at the sub-micron scale. While accurate constitutive models for nano-structured multifunctional composites are not available, there is a need for the physics-based engineering approximations that would extend the applicability of available models to address some of the nanoscale effects. As a starting point, numerical models sensitive to the materials microstructure (e.g., micromechanical and submicron length scales) are considered. This paper will focus on efficient multiscale methods for solving the Poisson equation for materials systems with significant microstructural hierarchy or multiple length scales.

1.1 Brief Review of Processing Flow Analyses

Multiscale materials modeling methods have wide range of applications. In computational materials science, hierarchical numerical methods are needed to address physical phenomena at sub-micron scale and their influence (or dependence) on macroscopic physical conditions. For the considered case, analysis methods are required to investigate material systems with small particles and their sensitivity to the global physical conditions (e.g., pressure, velocity distribution, shear stresses, etc.). The simplest models developed for heterogeneous material systems with several length scales involve the concepts of effective material properties such as an effective flow viscosity, for example. Classical analysis of the effective elongational viscosity by [Batchelor, 1971] illustrates this approach for suspensions of collimated fibers at dilute and semi-dilute concentrations in a Newtonian fluid. Later, [Shaqfeh and Friedrickson, 1990] have accounted for higher-order interaction between particles and changes in individual fiber orientation.

The need for hierarchical methods is especially acute when local effects such as fiber orientation and cluster formation are addressed for non-dilute suspensions. For concentrated suspensions, the anisotropic flow properties result in effective viscosity being represented by a tensor ([Christensen, 1992, Gibson, 1989]) with three independent viscosity coefficients. Pipes and his students ([Pipes et al., 1994]) have developed an effective constitutive relation for highly anisotropic behavior of concentrated flow systems. In a micromechanical analysis of collimated fiber suspensions, they predicted principal anisotropic viscosities as a function of both the matrix fluid properties and fiber packing geometries. Such analysis applies to materials containing perfectly collimated, discontinuous fibers that have an overlap length equal to one-half of the fiber length. It was also pointed out that the material

microstructure can not be perfect, but it can be tailored to enhance effective material properties ([Pipes et al., 1994]). Furthermore, microstructural variations in fiber length and fiber distribution should be examined.

Analyses of viscous materials systems with inclusions are often carried out by using unit-cell models that simulate a representative volume element and a typical microstructure under periodic boundary conditions ([Harik and Cairncross, 1999, Pipes et al., 1994]). Even processing flows with randomly-distributed filaments of different lengths can be described by a unit cell involving a semi-random fiber distribution and the surrounding polymer matrix (see Figure 4 in Section 5.) Geometry of such a unit cell is well represented by a rectangle. In the representative volume element, the viscous flow field is governed by the Poisson equation

$$\Delta U(x, y) = P_x, \quad (1)$$

where $U(x, y)$ is the flow velocity and P_x is the normalized pressure gradient. The unit cell is usually subjected to periodic boundary at $x = 0$ and $x = L$. Several relevant sets of boundary conditions should be considered at $y = 0$ and $y = H$.

1. The flow conditions in an extrusion flow or a pipe flow are repressed by the no-slip Dirichlet conditions: $U = 0$ for $y = 0$ and $y = H$.
2. If the processed fluid suspension is highly viscous and the channel walls are perfectly lubricated, then the perfect slip (the Neumann boundary conditions) may be allowed: $\partial_y U = 0$ for $y = 0$ and $y = H$. A partial slip can be simulated by mixed boundary conditions.
3. During manufacturing of thin films, the no-slip condition is maintained on the substrate surface along with the shear-free condition on top free surface: $U = 0$ for $y = 0$ and $\partial_y U = 0$ for $y = H$. Here, the normalized pressure gradient may vary: $P_x = 0, 1, 2, \dots$.
4. For the case of zero pressure gradient: $P_x = 0$, one should consider the processing conditions such as $U = 0$ for $y = 0$ and $\partial_y U = \tau$ for $y = H$, where τ is the applied shear traction that may vary from unity and up. Allowing a partial slip or a shear interaction on the inclusion surface may lead to highly nonlinear problems ([Harik and Cairncross, 1999]).

The above formulations of mathematical problems for polymer systems may be used for various particle distributions and different flow conditions. Later in the paper, examples of viscous flows of a polymer filled with elongated inclusions or short fibers will be presented along with the analysis of numerical efficiency for the examples considered. This paper addresses the need for numerical methods that have reliable and efficient capabilities to simulate various physical phenomena involving significant hierarchical effects.

1.2 On Superior Efficiency of Multigrid Methods

The “golden rule” of the multigrid (or more general multiscale) methods as defined in [Brandt, 1984] is the following: “The amount of computational work should be proportional to the amount of real physical changes in the computed system.”

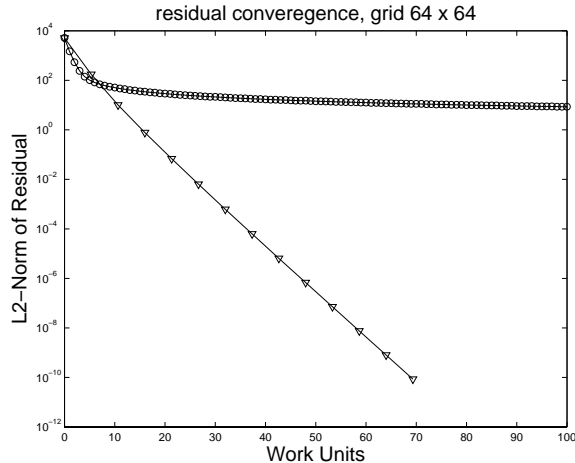


Figure 1: *Computational efficiency: multigrid cycle (triangles) vs. Gauss-Seidel relaxation (circles)*

That is, whenever, the computer grinds very hard for small or slow real physical effects, there must be a better computational way to achieve the same goal. Well known examples of numerical processes suffering a convergence slowdown are the usual single-grid methods as Gauss-Seidel relaxation, alternative-direction iterations, incomplete LU decomposition, (pseudo) time marching, etc., applied for solution of the algebraic equations arising from discretizing partial differential equations, e.g., by finite element, finite difference, finite volume or any other discretization method. The reasons for slowness may be quite different, for example, too small changes in the error from one iteration to the next or too small time steps allowed by the stability requirements. Multigrid methods may help in such problems.

The difficulties are usually related to some “stiffness” in the problem; i.e., to existence of several solution components with different scales which conflict each other. For example, smooth components can be efficiently approximated on coarse grids, but are very slow to converge on fine grids with many grid points per a characteristic scale; the high-frequency components must be approximated on the fine grids. By employing interactively several discretizations on different scales, multigrid techniques resolve such conflicts, avoid stalling, and do away with the computational waste.

The multigrid methods are often employed as fast solvers for discretized systems of partial differential equations [Brandt, 1984, Briggs et al., 2000, Trottenberg et al., 2000]. The multigrid solution usually requires just a few work units, where a work unit is the amount of computational work involved in expressing the algebraic equations (computing residuals or performing a simple relaxation sweep). Figure 1 illustrates the superiority of a multigrid method over the standard pointwise Gauss-Seidel relaxation technique in solving the Laplace equation with the Dirichlet boundary conditions (the solution values are given at the boundary). On this figure, the horizontal axis marks the work units required for computations; on the vertical axis, the L_2 -norm of the residual function is displayed. A random initial approximation in the interior has been chosen. The complexity of one Gauss-Seidel iteration is taken as a work unit, then, the complexity of the multigrid cycle used in the computations is 16/3 work units. After several initial relaxation sweeps, the convergence of

the Gauss-Seidel iterations (circles) is blocked, i.e., the error reduction is virtually invisible. On the contrary, the convergence rates the multigrid method (triangles) are fast and steady.

The applications of the multigrid methods extend far beyond solutions of discretized partial differential equations. Multigrid processes can also cut, often by several orders of magnitude, the requirements for the computer resources to simulate large systems involving multiscale phenomena. The greatest potential of the multigrid methods for nano-technological applications is opportunities for efficient bridging different scales and establishing an adequate communication between different-scale models.

1.3 Review of Multigrid Methods for Problems with Small Inclusions

In viscous suspension flows, differences in the scales of the suspended particles are usually significant. Assuming that even the smallest scales need to be resolved, enormous computational resources are required to solve such problems, and the most efficient algorithms available must be employed.

The fastest solvers for discretized elliptic partial differential equations are the full-multigrid (FMG) algorithms. Such an algorithm can, for example, solve the five-point discretization of the two-dimensional Poisson equation in a general domain to errors below the discretization errors in about 22 computer operations per grid point; each additional multigrid cycle, costing about 15 operations per grid point, reduces the errors by more than another order of magnitude. Similar efficiency, with the number of operations just growing proportionately to the number of operations involved in describing the discretized system, has been obtained for some nonlinear systems of equations, including Stokes and Navier-Stokes equations that represent more involved physical models.

The physical model (1) considered in this paper is a Poisson equation with multiple small inclusions (pieces of the boundary) embedded into the interior of the domain. These inclusions are often so small, occupying just a few points on the fine grid, that they cannot be properly incorporated into a coarse-grid formulation. Efficiency, i.e., convergence rate, of classical multigrid solvers deteriorates significantly because of the poor accuracy of coarse-grid approximations.

Several algorithm modifications has been studied in the literature to reduce (prevent) the degradation of the multigrid efficiency in solving problems with small inclusions:

- A thorough study of the effects of different ways to represent small inclusions on the coarse grids has been performed in [Mikulinsky, 1993]. The general conclusions suggest that inclusions with the Dirichlet boundary conditions represent the most difficult case, while small inclusions with the Neumann boundary conditions may be neglected on the coarse grids without any significant effect on the solver efficiency. In our numerical tests, we consider only the Dirichlet boundary conditions on the inclusions' borders, while the conditions at the external boundary of the computational domain may vary. For the inclusions with the Dirichlet boundary conditions, [Mikulinsky, 1993] proposes an extension on the next coarse-grid. The results of numerical tests presented in section 3 suggest that this approach is not always efficient. Moreover it is becoming even more problematic for the problems where the inclusions are varying in size, numerous, and actually occupying a significant part of the computational domain. Such problems represent the major interest for nanostructural materials applications and require other

solution approaches to be explored.

- An alternative method based on recombination of iterants (other names of the technique are polynomial acceleration and Krylov subspace acceleration) has been proposed in the paper [Brandt and Mikulinsky, 1995]. In this method, small inclusions are neglected on the coarse grids regardless the type of the boundary conditions. The special fine-grid errors that cannot be eliminated in the coarse-grid correction are reduced through recombining iterants obtained at different stages of a multigrid algorithm. Specifically, let us have k approximate solutions (iterants) $u^{(1)}, u^{(2)}, \dots, u^{(k)}$, e.g., each new iterants has been obtained by applying a multigrid cycle to the previous one. The new approximation can be computed as a (linear) combination of these iterants:

$$u = \sum_{n=1}^{k-1} \alpha_n u^{(n)} + \left(1 - \sum_{n=1}^{k-1} \alpha_n\right) u^{(k)}. \quad (2)$$

The coefficients α_n are chosen to minimize a norm of the residual function. In general these coefficients may be smooth functions of spatial variables allowing local recombination of iterants ([Brandt and Mikulinsky, 1995]). The algorithm significantly improves efficiency over classical multigrid methods, restoring fast convergence rates on relatively large (fine) grids. However, the efficiency per a work unit of the algorithm recombining a small number of iterants (say, two) is not optimal. Moreover, the convergence rates of the algorithm are slightly deteriorating for finer grids. The results of numerical experiments with this algorithm are reported in section 3. The convergence properties of the algorithm version with local recombination of iterants, that is the only viable option for the problem with many small inclusions, are even less favorable. Another disadvantage of neglecting inclusions on coarse grids may become prominent if the boundary conditions at an inclusion define the solution values far away from the inclusion. One can think about problems with the Neumann boundary conditions at the external edges of the computational domain and the Dirichlet boundary conditions at the only inclusion in the interior. In this case, the inclusion neglecting makes the problem ill posed.

We have convinced ourselves that recovering the optimal multigrid efficiency in solution of problems with small inclusions is impossible without coarse grids accounting for the presence of the inclusions. Assuming that in the most cases we cannot directly represent small inclusions on coarse grids, we have focused on changing locally the coarse-grid discretization to preserve some information about the inclusions in the fine-grid formulation. The ideal changes should bring enough accuracy to the coarse-grid solution and be cost effective.

The ultimate method that would resolve the problem of a poor coarse-grid accuracy is the highly accurate coarsening scheme proposed in [Brandt, 2000]. This methods provides a way to design a coarse-grid discretization that would approximate the fine-grid discretization with the accuracy that is exponentially improving as function of the characteristic size of the coarse-grid stencil. This algebraic coarsening scheme is extremely accurate but, in the same time, very cost consuming, requiring hundreds of computer operations to derive a discrete operator in one coarse-grid point. The most encouraging feature of this coarsening scheme is that the number of computer operations is constant independent on the grid size,

i.e., the complexity of the algorithm remains linear as function of the number of points. In general case of a very complicated environment, this method represents an attractive option for deriving large-scale approximations. The method (or its cheaper approximations) is still can be used locally for derivation of the coarse-grid equations not everywhere, but only in “problematic” regions.

The coarsening method employed in this paper can be characterized as a local Galerkin coarsening ([Briggs et al., 2000, Trottenberg et al., 2000]). The standard Galerkin coarsening method assumes that an accurate approximation to the fine-grid solution can be obtained by a linear interpolation from a coarse-grid function. The coarse-grid operator is defined by successively applying (1) prolongation operator, that interpolates coarse-grid functions to the fine grid, (2) the fine-grid discretization operator, and (3) the restriction operator, that averages the obtained fine-grid function back to the coarse grid. We call this method local because the Galerkin method is only employed in the neighborhoods of some of inclusions depending on their coarse-grid representation.

The coarse-grid representation of an inclusion used in our tests contains only the coarse-grid points located at the area physically occupied by the inclusion. This representation implies that the number of grid-points representing the inclusion may only decrease on the coarse grids. The inclusion is called fully represented on the coarse grid, if it spans the same number of points on the coarse grid as on the fine grid, otherwise, the inclusion is called contracted or vanished. The relative part of the area that is strongly affected by an inclusion fully represented on the coarse grid is expanded. Our experience shows that such expanded inclusions represent the most difficult case in the numerical computations.

We assume that the target grid is fine enough to accurately represent all the inclusions. Thus, on the target grid, the discrete operator is always the five-point Laplacian at all the interior grid points. On the coarse grids, away from the inclusions or in the vicinity of a fully represented inclusion, the discretization is again the five-point Laplace operator. In the neighborhood of a contracted or vanished inclusion, the Galerkin coarsening is used. The details of the restriction and prolongation operators used in the Galerkin coarsening are given in section 3.

2 PROBLEM FORMULATION

The differential Poisson equation defined in an open domain Ω with boundary $\partial\Omega$ is given by

$$\Delta u(x) = f(x) \quad (x \in \Omega), \quad (3)$$

where f is a known source function, u is the unknown function, and $\Delta \equiv \partial_{xx} + \partial_{yy}$ is the Laplace operator. In addition to the interior equation (3), suitable boundary conditions will be assumed on $\partial\Omega$. In our standard example, this will be either the Dirichlet boundary condition

$$u(x) = g_1(x) \quad (x \in \partial\Omega) \quad (4)$$

or the Neumann boundary condition

$$\partial_n u(x) = g_2(x) \quad (x \in \partial\Omega), \quad (5)$$

where ∂_n is a derivative along the direction normal to the boundary.

2.1 Multigrid Cycles

In this section, multigrid cycles for solving a discrete approximation to the problem (3) are defined. The discrete operators are finite difference approximations to the Laplace operator on Cartesian possibly anisotropic grids. For any grid Ω_{h_x, h_y} covering the domain Ω , where h_x and h_y are meshsizes in the corresponding directions, the discretization of (3) is written as

$$\Delta^{h_x, h_y} u_{i,j} = f_{i,j}, \quad (i, j) \in \Omega_{h_x, h_y}, \quad (6)$$

where $i = x/h_x, j = y/h_y$ are integers and

$$\Delta^{h_x, h_y} u_{i,j} \equiv \frac{1}{h_x^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{h_y^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \quad (7)$$

is the five-point Laplacian. Also for simplicity, we treat here only problems where $\partial\Omega$ coincides with the grid lines of all our grids, in which case the implementation of the boundary condition (4) is straightforward. The discrete boundary condition (5) (together with the interior equation) is implemented at the boundary $j = 0$ as

$$\frac{1}{h_x^2} (u_{i+1,0} - 2u_{i,0} + u_{i-1,0}) + \frac{1}{h_y^2} (2u_{i,1} - 2u_{i,0}) = f_{i,0} + \frac{1}{h_y} g_2(ih_x). \quad (8)$$

Given a discrete equation of the form (6), and an initial approximate solution at the fine grid, a two-level Correction Scheme(CS) cycle can be described in the following six steps [Brandt, 1984, Briggs et al., 2000, Trottenberg et al., 2000].

1. *Pre-relaxation.* Improve the current fine-grid solution approximation by ν_1 relaxation sweeps.
2. *Residual restriction.* Form a coarse-grid approximation to the fine-grid residual function. This intergrid transfer operator is called restriction and is denoted R^h
3. *Coarse-grid problem.* Construct the coarse-grid operator and form the coarse-grid problem with the source function given by the residual restricted at step 2. The coarse-grid operator is denoted Δ^{H_x, H_y} , where H_x and H_y are the coarse-grid meshsizes.
4. *Coarse-grid solution.* Solve the coarse-grid problem. The solution is an approximation to the fine-grid error function. At this stage, we do not specify the solution method. It can be any method (direct or iterative) providing an accurate solution to the coarse-grid problem.
5. *Correction prolongation.* Interpolate the coarse-grid solution to correct the current fine-grid approximate solution. This intergrid transfer operator is called prolongation and is denoted P^h
6. *Post-relaxation.* Improve the current fine-grid solution approximation by ν_2 relaxation sweeps.

A multilevel cycle is obtained from the two-level cycle by recursively applying the same cycle to the solution of the coarse-grid problem (step 4). The coarsest grid where the problem is solved precisely is the grid having few mesh spaces in at least one of the spatial directions. One can introduce the *the cycle index* parameter γ that is defined as the number of coarser-grid cycles per fine grid cycle. Cycles with $\gamma = 1$ are called *V* cycles and denoted $V(\nu_1, \nu_2)$; those with $\gamma = 2$ are termed *W* cycles and denoted $W(\nu_1, \nu_2)$. For uniformly elliptic problems either of these is used, usually with $\nu_1 + \nu_2 = 2$ or 3 . Each such cycle typically reduces the error at least by an order of magnitude.

In the full multigrid (FMG) algorithm, an initial fine-grid approximation to the solution of (6) is first obtained from the coarser grid, and then it is improved by multigrid cycles. Thus, the algorithm $FMG_m(\nu_1, \nu_2)$ for solving (6) is recursively defined as the following four steps.

1. *Coarse-grid Solution.* The coarse-grid equation is formed by choosing the coarse-grid operator and restricting the fine-grid source function $f_{i,j}$ to the coarse grid. In the FMG algorithm, the source function at the next coarse level is obtained by properly averaging the corresponding fine-grid function; only at the target finest level, the source function is calculated directly from f . This guarantees the proper performance of the algorithm for *any* (not just smooth) f . The discretization of the boundary conditions could be handled similarly: in particular the values $g_{i,j}$ of the boundary conditions on the coarse grids could be obtained by fully averaging functions g from the fine grids.
2. *Recursion.* For the coarsest grid, solve the problem directly, otherwise apply $FMG(\nu_1, \nu_2)$ for solving the current-grid problem.
3. *Solution interpolation.* Interpolate the solution obtained after the FMG algorithm to the next fine grid to form the initial approximation.
4. *Multigrid Cycle.* Improve the initial approximation with m multigrid cycles (say, $V(\nu_1, \nu_2)$).

For uniformly elliptic problems, a cheap algorithm such as $FMG_1(1, 1)$ is easily enough to obtain a solution error substantially smaller than the discretization error.

3 MODEL PROBLEM: ONE INCLUSION

In this section we test different computational approaches to handle the problem associated with one small inclusion.

Let the computational domain $(x, y) \in [0, 4] \times [0, 1]$ be covered with an anisotropic Cartesian mesh with mesh spaces h_x and h_y in the corresponding directions. The grid indexes are changed as $i = 0, 1, \dots, n_x$, $n_x = 4/h_x$ and $j = 0, 1, \dots, n_y$, $n_y = 1/h_y$. The standard second-order accurate five-point discretization for the Laplace operator is applied in the interior of the domain. The discrete periodic conditions are applied at the vertical edges of the external boundary as $u(0, j) = u(n_x, j)$. The type of boundary conditions (Dirichlet or Neumann) may vary at the horizontal pieces of the domain boundary, $j = 0$ and $j = n_y$. The coarse grids are built by semicoarsening, i.e., the mesh sizes are doubled in the x -direction ($H_x = 2h_x, N_x = n_x/2$) and remain the same in the y -direction ($H_y = h_y, N_y = n_y$).

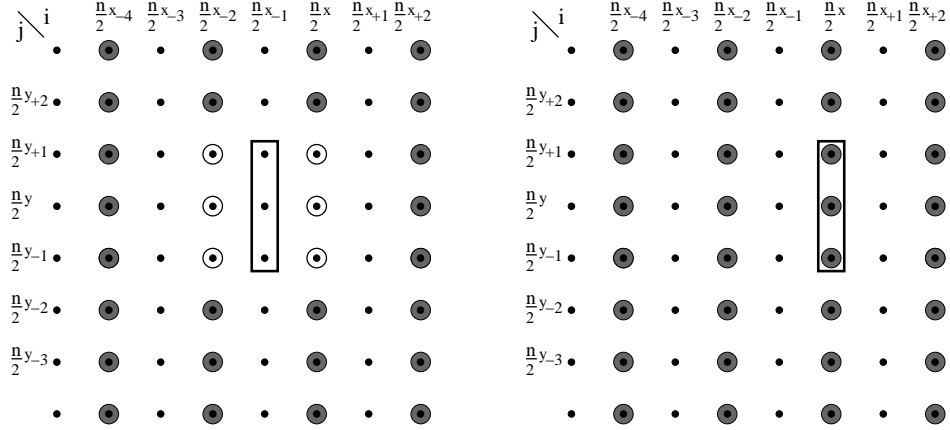


Figure 2: *The inclusion location*

The coarsest grid where the problem is solved precisely is the grid with eight mesh spaces in the horizontal direction. A small inclusion with the Dirichlet boundary condition is located in the middle of the domain occupying three grid points on the fine grid, (\tilde{i}, \tilde{j}) , $\tilde{j} = (\frac{n_y}{2} - 1), \frac{n_y}{2}, (\frac{n_y}{2} + 1)$. The x -coordinate of the inclusion may vary, being either $\tilde{i} = \frac{n_x}{2} - 1$ or $\tilde{i} = \frac{n_x}{2}$, to represent both the important cases: (1) the inclusion is totally neglected at the coarse-grid and (2) the inclusion is fully represented on all the coarse grids. Figure 2 illustrates the possible inclusion location. The small bullets denote the fine-grid points. The inclusion is shown as a rectangle. At the points within the rectangle, the Dirichlet boundary conditions are applied. The gray color denotes the coarse-grid points with the standard five-point discretization of the Laplace operator. The white circles are the coarse-grid points where the Galerkin operator is employed. The left figure shows the inclusion to be neglected on the coarse grid; on the right figure, the inclusions is fully represented on the coarse grid.

4 NUMERICAL IMPLEMENTATION

4.1 Modified Multigrid Cycle

In this paper we propose a modification of a standard multigrid cycle to be used in computations with the small inclusions. The ingredients of the modified cycle are the following:

1. *Relaxation.* The relaxation method is the red-black vertical line (zebra) Gauss-Seidel relaxation [Briggs et al., 2000, Trottenberg et al., 2000]. In this method, the domain is swept twice. In the first sweep, all the solution $u_{i,j}$ at the grid points with odd sub-indexes i are simultaneously updated to satisfy local discrete equations. After this sweep, the residuals become zero at the every other vertical line. The second sweep updates solution at the vertical lines with even sub-indexes i . For proper discretized uniformly elliptic problems, this relaxation scheme is usually very efficient in reducing high-frequency error components. A multigrid algorithm combined line relaxation with semicoarsening has been studied earlier (e.g., in [J. E. Dendy Jr. et al., 1992])

and is known for an excellent efficiency in serial and parallel computations. This combination has also been successful for efficient multigrid solution of nonelliptic problems [Brandt and Diskin, 2000, Diskin, 2001, Llorente et al., 2001], forming a good basis for future extensions to Navier-Stokes equations. The relaxation used in our tests is supplemented with a local relaxation procedure described in the paragraph 5 below.

2. *Restriction.* The residual restriction at the most of the domain is given by a symmetric, horizontal, full weighting operator

$$R_{i,j} = R^h r_{i,j} = \frac{1}{4} \left(r_{2i-1,j} + 2r_{2i,j} + r_{2i+1,j} \right), \quad (9)$$

where $r_{i,j} = f_{i,j} - \Delta^{h_x, h_y} u_{i,j}$ is the fine-grid residual function, and $R_{i,j}$ is the coarse-grid residual function. The same restriction, R^h , is used for computing the coarse-grid source function within the FMG algorithm.

To guarantee the optimal efficiency, the residual restriction should be modified in the vicinity of the boundary because, for the Dirichlet boundary, the coefficients of the operator R^h should depend on the distance from the boundary. It is a difficult task to derive the precise residual restriction in the case of a complicated boundary shape. The general remedy is to include local relaxation sweeps reducing the residuals in the vicinity of the boundary to the level much lower than the residual level typical for the interior of the domain. If the residuals become small, their precise restriction is not important any more. For example, [Brandt and Mikulinsky, 1995] suggests not to transfer these residuals at all. In our tests, we employ a local relaxation procedure in the vicinity of the inclusion and modify the residual restriction near the inclusions depending on how the inclusion is represented on the coarse-grid. If the inclusion is fully represented on the coarse grid, i.e., \tilde{i} is even ($\tilde{i} = 2k$), the restriction operator remains unchanged, otherwise ($\tilde{i} = 2k - 1$), the restriction operator is changed at the adjacent coarse-grid points as

$$\begin{aligned} R_{k,\tilde{j}} &= \frac{1}{3} \left(2r_{2k,\tilde{j}} + r_{2k+1,\tilde{j}} \right), \\ R_{k-1,\tilde{j}} &= \frac{1}{3} \left(2r_{2k-2,\tilde{j}} + r_{2k-3,\tilde{j}} \right). \end{aligned} \quad (10)$$

This change of the restriction does not significantly affect the source function on the coarse grid (because of the local relaxation involved), but it is important for definition of the Galerkin operator.

3. *Prolongation.* The prolongation operator for the coarse-grid correction is a second-order accurate linear interpolation defined as

$$v = P^h V; \quad \begin{cases} v_{2i,j} &= V_{i,j}, \\ v_{2i-1,j} &= \frac{1}{2} \left(V_{i-1,j} + V_{i,j} \right), \end{cases} \quad (11)$$

where V is the coarse-grid solution and v is the correction to the fine-grid solution approximation. The solution prolongation within an $FMG_1(1, 1)$ algorithm usually requires a higher order of accuracy to guarantee the convergence of the errors below

the level of discretization error. However, practically, a linear interpolation is often good enough; in our numerical tests, the same linear prolongation operator (11) is used for the FMG solution interpolation.

4. *Coarse-Grid Operator.* Away from the inclusion, the five-point Laplacian is always used on the coarse grids. If the inclusion is fully represented on the coarse-grids, this five-point discretization is used everywhere in the interior. If the inclusion is neglected on the coarse grids, the discrete coarse-grid operators on the first coarse grid where the inclusion become invisible are changed at the points located at the less-than-meshsize distance from the inclusion. The operators are constructed by the Galerkin method as

$$\Delta^{H_x, H_y} \equiv R^h \Delta^{h_x, h_y} P^h, \quad (12)$$

but use the boundary values at the inclusion explicitly. Because of the specific form of the restriction operator R^h , this construction leads to a six-point one-sided discrete stencil. The Galerkin method is also used for the coarser grids in this region, always applied to the points adjacent to the inclusion location. For the next coarsening steps, the symmetric restriction operator R^h (9) is used.

5. *Local Relaxation.* The local relaxation performs additional relaxation sweeps in the region adjacent to the inclusion. The major goal of the local relaxation is to reduce residuals here to facilitate their restriction to the coarser grid. The local relaxation sweeps implemented as a part of a global relaxation procedure. In the numerical tests, all the points located within a rectangular box $\tilde{i} - 3 \leq i \leq \tilde{i} + 3, \frac{n_y}{2} - 4 \leq j \leq \frac{n_y}{2} + 4$ are relaxed with ten additional sweeps per each global relaxation step. At the pre-relaxation stage, the local sweeps follow the global one; at the post relaxation stage, the local sweeps precede the global one. The local relaxation is performed on all the grids where the inclusion is represented.

4.2 Numerical tests

In this section we compare the performance of three different multigrid algorithms: (I) The classical multigrid cycle that uses the standard coarse-grid discretization (7) and residual restriction (9) everywhere, (II) the accelerated multigrid cycle employing recombination of two iterants, and (III) the modified cycle using the local Galerkin coarsening. The algorithms (I) and (III) are identical when the inclusion is fully represented on all the grids.

Table 1: *Asymptotic convergence of $V(1, 1)$ multigrid cycles with the Dirichlet boundary conditions at the upper boundary and the Neumann boundary conditions at the lower boundary. The inclusion is neglected on the coarse grids*

$n_x \times n_y$	Ref.	Alg. I	Alg. II	Alg. III
64×32	0.09	0.63	0.03 (0.17)	0.10
128×64	0.09	0.98	0.04 (0.20)	0.10
256×128	0.09	1.31	0.05 (0.23)	0.10

Table 2: *Asymptotic convergence of multigrid cycles with the Dirichlet boundary conditions at the upper boundary and the Neumann boundary conditions at the lower boundary. The inclusion is fully represented on all the coarse grids*

$n_x \times n_y$	2-level Ref.	V(1,1)	2-level V(1,1)	W(1,1)	FV(1,1)
64×32	0.05	0.11	0.05	0.05	0.05
128×64	0.05	0.19	0.07	0.06	0.08
256×128	0.05	0.27	0.07	0.07	0.10

In the table 1 we compare the asymptotic convergence of $V(1,1)$ multigrid cycles for target grids of different sizes. The zero values for the source function f and for the boundary conditions are chosen to ensure the zero exact solution. Initial solution values in the interior are chosen randomly. The asymptotic rate is considered established when the differences in the convergence rates of three consecutive cycles are less than 0.005. As a reference point, we use the convergence of the standard multigrid $V(1,1)$ cycle in the domain without inclusions. Because the algorithm recombining iterants need to perform two cycles at each iteration, the numbers in parentheses show the convergence rate normalized per work in one cycle (square root of the first value). One can see that the classical algorithm does not work at all diverging for the fine grids. The algorithm with recombination provides good convergence rates, however, the (normalized) rates are significantly slower than the rates of the reference algorithm and seem to slightly deteriorate on the fine grids. Moreover, the FMG method with the recombination cycle is significantly more expensive because it requires at least two cycles to be performed at each level. The modified algorithm fully restores the convergence rates demonstrated by the reference algorithm.

Table 2 contains the results of the tests performed for the problem where the inclusion is fully represented on all the grids. The full coarse-grid representation of the inclusion leads to an expansion of the area where the solution is strongly affected by the inclusion. The reference algorithm used for the tests is a two-level $V(1,1)$ cycle for the problem with no inclusions. We do not present here the results for the algorithm with recombination of iterants; the algorithm neglects the inclusion on the coarse grids and behaves in the same way as in previous tests (table 1). Instead we study the performance of the modified algorithm depending on the cycle index. The reason for this study is the observation that the convergence rates of the classical V-cycle (for this problem, the modified and classical V-cycles coincide) is significantly slower than the convergence rates of the reference cycle (see data in columns 2 and 3 of table 2). On the other hand, the convergence rates for a two-level cycle applied to the problem with a small inclusion (column 4 in table 2) are (nearly) as good as for referenced one. Thus, one can expect to improve the convergence rates with cycles of higher indexes. Indeed, $W(1,1)$ -cycle is capable to recover the convergence rates of the two-level algorithm (column 5 in table 2). However, for the two dimensional algorithms with semicoarsening, the complexity of W cycles is not linear any more. In fact, the amount of work performed at a coarse grid becomes approximately the same as the amount of work performed at the target fine grid. This increased amount of work on the coarse grids leads to a logarithmic factor, (i.e., the number-of-levels factor) in the complexity formula. To keep

the complexity linear, we tested a much cheaper cycle, called FV-cycle, that can be derived from the two-level algorithm by applying an FMG method to solve the coarse-grid problem. FV cycle is 50% more expansive than V -cycle and has a linear complexity. The results show that FV cycle demonstrates (nearly) optimal convergence rates for the problem with the inclusion fully represented on the coarse grids.

5 SIMULATIONS OF PROCESSING FLOWS WITH MULTIPLE INCLUSIONS

The modified multigrid FV cycle is applied for solution of the Poisson equation with many inclusions differing in length and occupying an essential part of the domain.

For simplicity of programming, we make the following assumptions about the inclusions' locations and size distribution; none of these assumptions is mandatory for the applicability and efficiency of the numerical algorithm.

- The inclusion borders coincide with the grid lines on all the grids, i.e., all the inclusions are rectangular with a horizontal/vertical orientation; no oblique inclusions are allowed.
- The inclusions are varying in the lengths only; all the inclusions have the same width of 1/64 of the vertical size of the computational domain.
- The inclusions cannot touch the horizontal pieces of the external boundary.
- If we separate the computational domain into imaginary 64 equal horizontal levels, the inclusions are allowed to occupy only odd levels, but not the 1-st level adjacent to the external boundary. This assumption prevents the coarse-grid collisions of the inclusions located on different levels. The inclusions located on the same level, still may collide on the coarse grids.

The borders of the coarse-grid representation of an inclusion are given by intersection of the coarse grid lines with the area occupied by the inclusion. Therefore, an inclusion may contract or occupy the same area. If the inclusion contracts, the Galerkin method is used to form the coarse-grid operator in the neighborhood of the contacted side of the inclusion. Otherwise, the standard five-point Laplacian is used.

The statistical distribution of inclusion lengths is shown in the Figure 3. The lengths are measured in the width units. The inclusions spatial distribution is shown in the figure 4. The volume ratio is equal 0.13, i.e., about 13% of the computational domain is occupied by the inclusions.

Three model problems are considered:

1. The homogeneous Dirichlet boundary conditions at both the horizontal pieces of the external boundary ($u = 0$ for $y = 0$ and $y = 1$) and a unit normalized pressure gradient in the interior of the domain ($f = -1$).
2. The homogeneous Neumann boundary conditions at the top boundary ($\partial_y u = 0$ at $y = 1$) and the homogeneous Dirichlet boundary conditions at the bottom ($u = 0$ at $y = 0$); the normalized pressure gradient is one in the interior of the domain ($f = -1$).

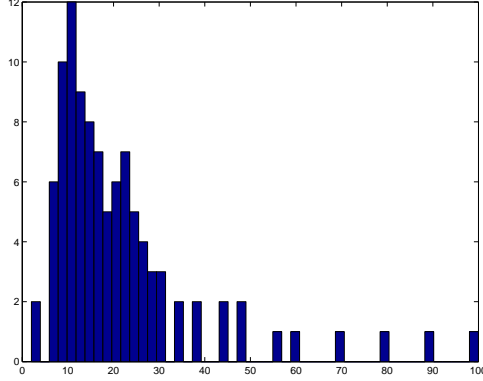


Figure 3: *The distribution of inclusions' lengths.*

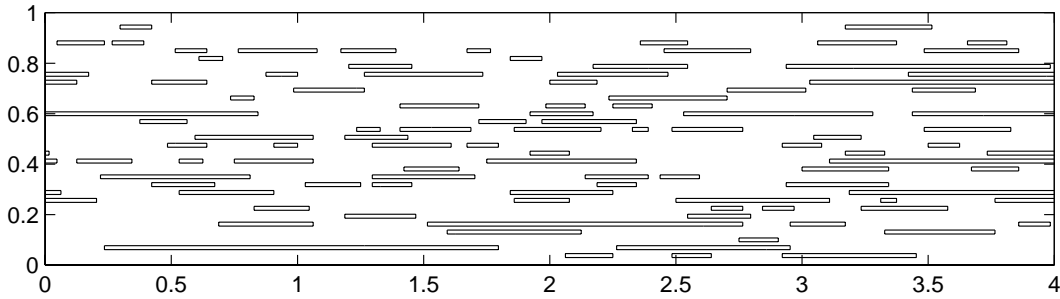


Figure 4: *The inclusions' distribution*

3. The unit Neumann boundary conditions at the top boundary ($\partial_y u = 1$ at $y = 1$) and homogeneous Dirichlet boundary conditions at the bottom ($u = 0$ at $y = 0$); the pressure gradient is zero ($f = 0$).

The modified $FV(1, 1)$ is applied for the three problem. The target grid has the mesh spacing $h_x = 2^{-8}$, $h_y = 2^{-9}$. The computations are terminated when the L_2 -norm of the residual function become less than 10^{-8} . The convergence in the last cycle performed is considered as the asymptotic convergence rate. For all the three problems the asymptotic convergence rate was 0.12.

6 SIMULATION RESULTS

In the processing flows with viscous matrix and rigid inclusions, the dispersion of filler particles often represents a major problem. Consequently, the concentration of particles can not be expected to be uniform. As a result of the particle distribution considered (Figure 4), the unit cell in numerical simulations has polymer-rich regions (e.g., around points $[x = 0.5, y = 0.2]$ and $[x = 2.7, y = 0.5]$ as well as along the upper boundary $[y = 1]$) and the regions with high density of inclusions. These microstructural features of this particular processing flow lead to a specific flow field that is dependent on the local distribution of obstacles. The fluid-rich regions have higher velocities and velocity gradients, while the inclusion-rich regions have much lower velocities.

The first example of a flow field is shown in Figure 5. It corresponds to a case of a multiphase flow in a long duct with the no-slip boundary conditions along its lower and

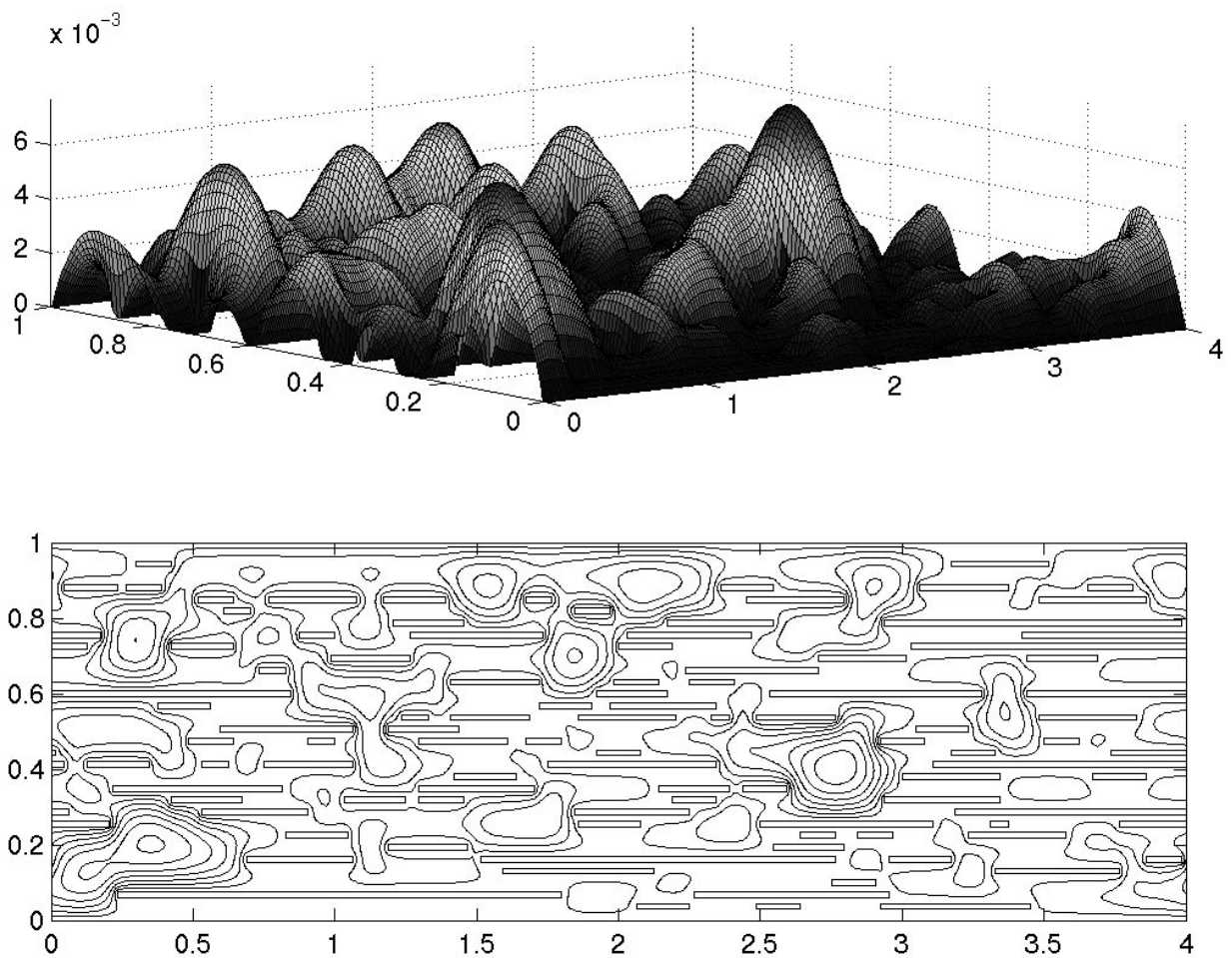


Figure 5: *Solution for the problem 1 with the Dirichlet boundary conditions and the unit pressure gradient.*

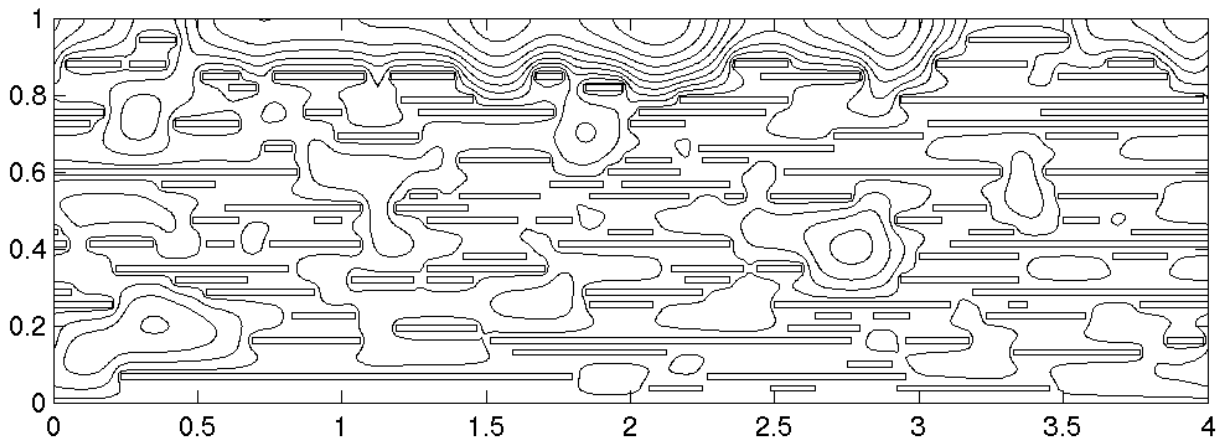
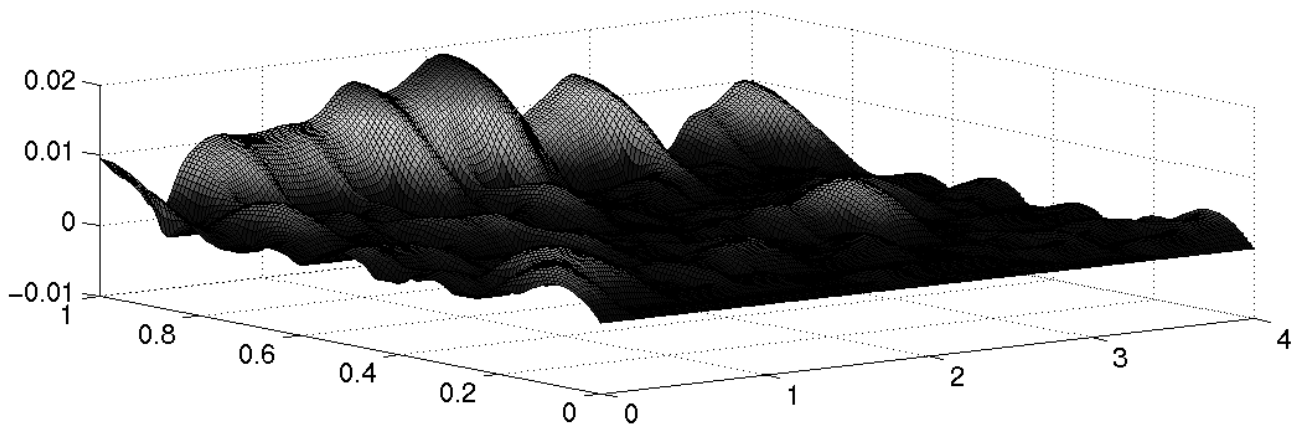


Figure 6: *Solution for the problem 2 with the Neumann boundary conditions at the top, the Dirichlet boundary conditions at the bottom, and the unit pressure gradient.*

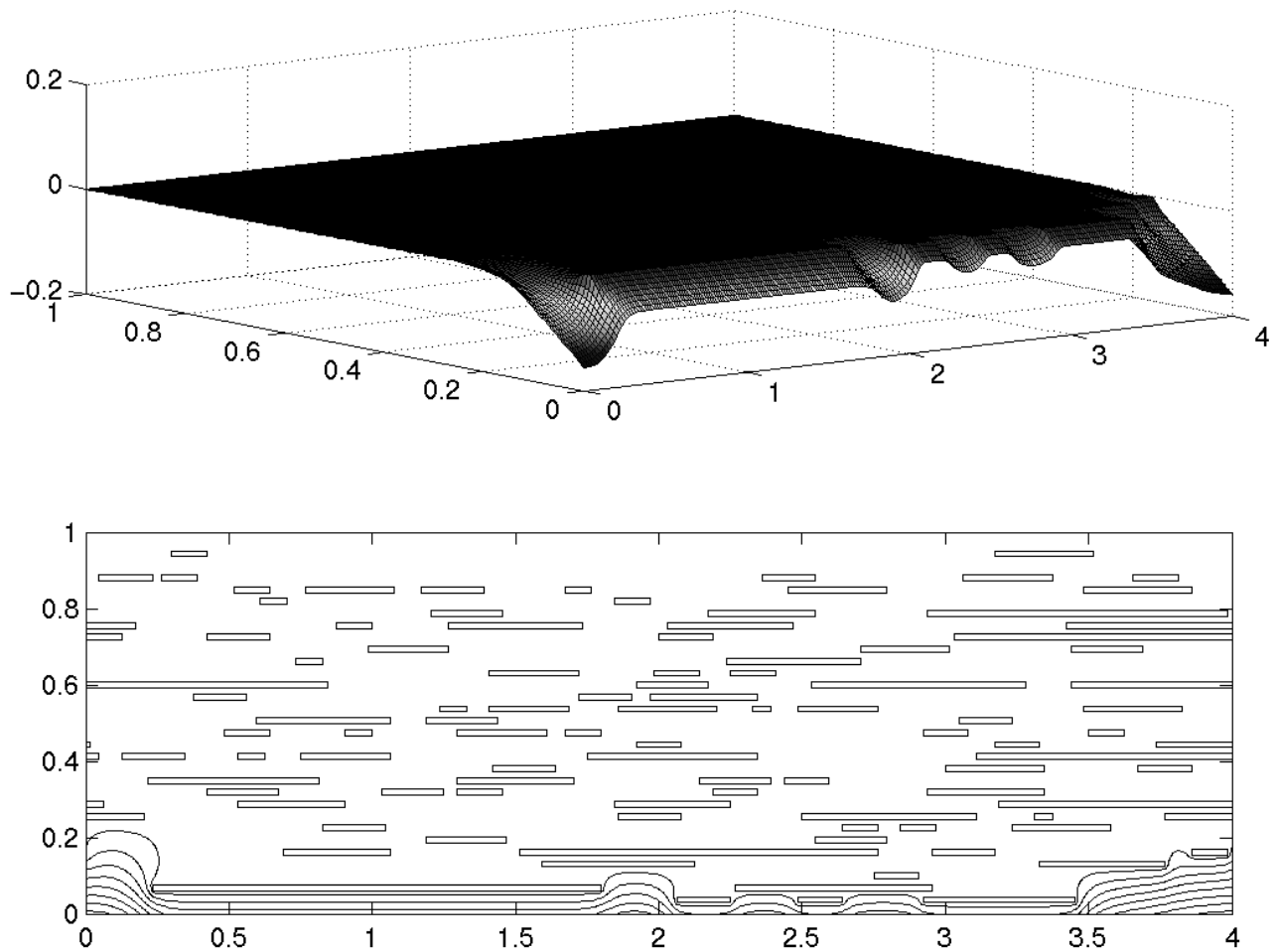


Figure 7: *Solution for the problem 3 with the unit Neumann boundary conditions at the bottom, the Dirichlet boundary conditions at the top, and the zero pressure gradient.*

upper boundary ($y = 0$ and $y = 1$, respectively) and periodic boundary conditions on the left and right boundaries that simulate an infinite flow geometry along the x -direction. The flow is subjected to a pressure gradient that controls the magnitude of the velocity peaks seen in the top plot of the figure. All of the fluid-rich regions are clearly marked by the velocity peaks and the corresponding dense distribution of velocity contour lines. Although the velocity is zero along the top boundary of the cell ($y = 1$), there is enough fluid-filled space subjected to a pressure gradient so the flow can develop.

The second example of a suspension flow is shown in Figure 6. It describes a case of a fluid flow on an infinite substrate with the no-slip boundary condition on the bottom ($y = 0$) and shear free flow on top. The pressure gradient applied across this flow results in the predominant flow along the shear free surface that simulates a free surface. Under such conditions, there is little flow in the sub-surface region filled with particles that populate the cell toward the substrate. The near-surface fluid-rich region of the unit cell is sensitive to the applied pressure, and having a fluid-rich region is sufficient for developing a local fluid motion.

The third example of a processing flow (Figure 7) represents a rheological flow between two plates. A non-zero shear is applied to the bottom plate ($y = 0$), while the other plate is stationary (i.e. the no-slip condition is enforced along $y = 1$). The fact that the substrate-like surface is along the top boundary of the cell has no difference as far as gravity is concerned. The gravity is not accounted for due to its vanishing influence on the viscous flow in narrow channels or film flows, both of which involve cell geometries with high aspect ratios. As one could expect, when a tangential shear is applied to a fluid with suspended particles, the near-by region is most affected, while the rest of the particle-filled fluid stays undisturbed toward the substrate. The fluid-rich region near the substrate ($y = 1$) is subject to the zero-flow velocity on one side, while the other side is "shielded" from the applied shear by several layers of particles. This case can be further investigated to study the effective viscosity of such suspensions.

7 CONCLUSIONS

The paper presents a new multigrid method for efficient solution of flows with multiple small particles. The method employs a novel coarsening procedure based on the Galerkin coarsening that allows to preserve on the coarse grids an essential information about small-scale particles that cannot be directly represented on the coarse grids. Standard multigrid methods usually fail for such problems because of inability to handle inclusions with different length scales. The new method fully recovers the optimal efficiency of the fastest multigrid solvers. Multiple levels of hierarchy are linked so that the number of length scales does not hamper the computational efficiency.

Examples of the simulation results illustrate the key advantage of the proposed multigrid method that provides high resolution and an effective handling of the large number of physical details at various length scales. Numerical resolution allows one to simulate 1) the tiniest particles and local details of the flow field geometry; 2) conglomeration of particles, formation of particle clusters and subsequent deformation of such clusters; 3) local flow gradients and velocity peaks and 4) global interactions of particle clusters. These capabilities may assist in optimizing the processing conditions that would control the distribution of particles in the processing flows with filler additives, which is critical for improving dispersion of particles

and achieving desired material properties through optimal microstructure.

ACKNOWLEDGMENTS

The authors thank Dr. Cheol Park for useful discussions.

REFERENCES

- [Batchelor, 1971] Batchelor, G. K. (1971). The stress generated in a non-dilute suspension of elongated particles by pure straining motion. *J. Fluid Mech.*, 46:813–829.
- [Brandt, 1984] Brandt, A. (1984). Multigrid techniques: 1984 guide with applications to fluid dynamics. In *Lecture Notes for the Computational Fluid Dynamics, Lecture Series at the Von-Karman Institute for Fluid Dynamics*. The Weizmann Institute of Science, Rehovot, Israel. ISBN-3-88457-081-1, GMD-Studien Nr. 85, Available from GMD-AIW, Postfach 1316, D-53731, St. Augustin 1, Germany. Also available from Secretary, Department of Mathematics, University of Colorado at Denver, CO 80204-5300.
- [Brandt, 2000] Brandt, A. (2000). General highly accurate algebraic coarsening. *Electronic Trans. Num. An.*, 10:1–32.
- [Brandt and Diskin, 2000] Brandt, A. and Diskin, B. (2000). Multigrid solvers for non-aligned sonic flows. *SIAM J. Sci. Comp.*, 21(2):473–501.
- [Brandt and Mikulinsky, 1995] Brandt, A. and Mikulinsky, Vladimir (1995). On recombining iterants in multigrid algorithms and problems with small islands. *SIAM J. Sci. Comput.*, 16(1):20–28.
- [Briggs et al., 2000] Briggs, W. L., McCormick, Steve F., and E.Henson, Van (2000). *Multigrid Tutorial, 2nd edition*. SIAM, USA.
- [Christensen, 1992] Christensen, R. M. (1992). Effective viscous flow properties for fiber suspensions under concentrated conditions. TR UCRL-JC-110344, Lawrence Livermore National Laboratory.
- [Diskin, 2001] Diskin, B. (2001). Efficient multigrid methods for solving upwind-biased discretizations of the convection equation. *Applied Mathematics and Computation*, 123(3):343–379. (Also ICASE Report 99-25, NASA CR-1999-209355).
- [Gibson, 1989] Gibson, A. G. (1989). Die entry of reinforced polymers. *Composites*, 20:57–64.
- [Harik and Cairncross, 1999] Harik, V. M. and Cairncross, R. A. (1999). Evolution of interfacial voids around a cylindrical inclusion. *J. Applied Mech.*, 66:310–314.
- [J. E. Dendy Jr. et al., 1992] J. E. Dendy Jr., Ida, M. P., and Rutledge, J. M. (1992). A semi-coarsening multigrid algorithm for SIMD machines. *SIAM J. Sci. Stat. Comput.*, 13:1460–1469.

- [Llorente et al., 2001] Llorente, Ignacio M., Prieto-Matias, Manuel, and Diskin, Boris (2001). An efficient parallel multigrid solver for 3-d convection and convection-diffusion problems. *Parallel Computing*, 27:1715–1741.
- [Mikulinsky, 1993] Mikulinsky, Vladimir (1993). *Multigrid Treatment of Small Islands*. PhD thesis, The Weizmann Institute of Science.
- [Pipes et al., 1994] Pipes, R. B., Coffin, D. W., Simacek, P., Shuler, S. F., and Okine, R. K. (1994). Rheological behavior of collimated fiber thermoplastic composite materials. In Advani, S. G., editor, *Flow and Rheology in Polymer Composites Manufacturing*. Elsevier, London.
- [Shaqfeh and Friedrickson, 1990] Shaqfeh, E. S. G. and Friedrickson, G. H. (1990). Hydrodynamic stress in a suspension of rods. *Phys. Fluids A*, 2:7–24.
- [Trottenberg et al., 2000] Trottenberg, Ulrich, Oosterlee, C. W., and Schüller, A. (2000). *Multigrid*. Academic Press, London.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)			2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSORING/MONITOR'S ACRONYM(S)		
				11. SPONSORING/MONITORING REPORT NUMBER		
12. DISTRIBUTION/AVAILABILITY STATEMENT						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT						
15. SUBJECT TERMS						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)	