INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **38**:1009–1025 (DOI: 10.1002/fld.249)

A domain decomposition method for modelling Stokes flow in porous materials

Guangli Liu and Karsten E. Thompson[*,†]

Gordon A. and Mary Cain Department of Chemical Engineering, Louisiana State University, BatonRouge, LA 70803, U.S.A.

SUMMARY

An algorithm is presented for solving the Stokes equation in large disordered two-dimensional porous domains. In this work, it is applied to random packings of discs, but the geometry can be essentially arbitrary. The approach includes the subdivision of the domain and a subsequent application of boundary integral equations to the subdomains. This gives a block diagonal matrix with sparse off-block components that arise from shared variables on internal subdomain boundaries. The global problem is solved using a biconjugate gradient routine with preconditioning. Results show that the effectiveness of the preconditioner is strongly affected by the subdomain structure, from which a methodology is proposed for the domain decomposition step. A minimum is observed in the solution time versus subdomain size, which is governed by the time required for preconditioning, the time for vector multiplications in the biconjugate gradient routine, the iterative convergence rate and issues related to memory allocation. The method is demonstrated on various domains including a random 1000-particle domain. The solution can be used for efficient recovery of point velocities, which is discussed in the context of stochastic modelling of solute transport. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: boundary element method; domain decomposition; Stokes flow; disordered porous media; preconditioner

1. INTRODUCTION

The problem of two-dimensional (2D) Stokes flow through a random array of discs has been studied in relation to a number of phenomena. In our research, we examine convective transport phenomena in heterogeneous porous media [1] and for certain applications, this 2D problem is used as an analogue to three-dimensional (3D) flows in sphere packings. Also, the problem is equivalent to 3D flow transverse to parallel cylinders, which is important for modelling fluid flow in aligned fibrous materials [2].

Analytic solutions to the Stokes equation exist only for simple geometries; hence, porous media problems generally require numerical approaches. If the structure is regular, then one

[*] Correspondence to: K. E. Thompson, Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA 70803, U.S.A.

[†] E-mail: karsten@che.lsu.edu

Received 28 December 2000
Revised 29 July 2001

has more flexibility in choosing a numerical approach. However, for disordered media, the computational problems are much larger, and solution methods are more restricted. Typical limitations can include size, solid volume fraction, particle shape and particle spacing, depending on the method.

The purpose of this work was to develop an efficient computational algorithm for relatively large domains that have few restrictions associated with pore structure or porosity. Although examples shown in this paper are for 2D arrays of discs, we have tested it using media composed of angular particles, and it could equally well be applied to unstructured domains with nearly arbitrary particle shapes. The solution method combines attributes of both boundary integral techniques and domain decomposition. Use of the boundary element method reduces the dimension of the numerical problem by one, thus allowing for easy discretization of the flow domain and giving a smaller number of unknowns. Domain decomposition vastly improves the sparseness of the resulting matrix equation, thereby allowing one to solve much larger problems than what the standard boundary integral methods would allow.

2. NUMERICAL MODELLING OF STOKES FLOW IN 2D

The problem of Stokes flow in periodic arrays of cylinders has been examined extensively. The general problem was first solved by Sangani and Acrivos [3], who recast the Stokes equation in terms of the stream function and vorticity and employed a truncated series solution for these two variables that satisfied specific boundary conditions on the periodic unit cell. Coefficients in the series were found by approximately satisfying the remaining boundary conditions. Larson and Higdon [4] studied flow in a similar periodic porous medium, but near the medium's interface with a bulk-flowing fluid. The Stokes equation in each periodic cell was solved using a boundary element method. However, an iterative process was used to obtain boundary values as a function of distance from the plane-porous interface. The periodicity in structure allowed a highly efficient solution (comparable to a fully periodic solution) even though the flow was not periodic.

Solving for Stokes flow in random porous media is a more challenging problem. Sangani and Yao [5] first addressed the 2D problem (i.e. transverse Stokes flow over random parallel cylinders) using the stream function formulation and a multipole expansion of the singularly forced Laplace equation. A numerical solution was used to determine coefficients in the resulting series expansion. Sangani and Mo [6] were able to obtain solutions in somewhat larger random domains. They accounted for lubrication forces using multipole expansions along the centreline of near-particle gaps.

The lattice-Boltzmann method provides a flexible method for modelling flow in porous materials. In particular, it is readily applied to arbitrary domain shapes and can be applied to flows having finite Reynolds numbers. The problem of 2D flow through random discs has been modelled by Koch and Ladd [7] for Reynolds numbers of order 100 and lower. Verberg and Ladd [8] have recently developed a time-independent version for low-Reynolds number flow that converges much more quickly than the traditional method. Using the time-independent technique, flow in 2D random arrays of up to 790 particles was modelled. The disadvantages of the lattice-Boltzmann method is slow convergence (using the traditional technique), especially at low porosities [8], and also the need for very fine discretization throughout a domain to accurately resolve velocity in small constrictions or very near particle surfaces.

All the above methods, as well as finite difference methods, are powerful approaches, and the one best suited for a given problem depends on the details of that problem and the specific information one requires from its solution. In this work, we sought to improve certain capabilities that were identified as a part of a larger project to develop hybrid modelling techniques for very large heterogeneous porous media. First, we require the high-resolution part of the modelling (e.g. where streamline resolution of flow is obtained) to be applied over relatively large characteristic lengths without assuming a smaller scale for periodicity. This attribute is crucial when modelling media where spatial correlation exists. Second, it was necessary that the algorithm be effective for arbitrary particle spacing and porosity (and ideally, arbitrary pore structure). Finally, we required a method that allows for very accurate recovery of point velocities in the immediate region of specified solid–fluid interfaces. This last requirement is associated with studies of interfacial transport, where the hydrodynamics within thin concentration (or temperature) boundary layers around particle surfaces must be known accurately.

3. PROBLEM FORMULATION

The method presented in the paper is applicable to general 2D domains and is being extended to three dimensions. For simplicity, however, computations performed during this research were made using two-dimensional disordered arrays of discs as shown in Figure 1. The radii of discs are chosen randomly from a size distribution, and placed in the domain using a 2D analogue to collective rearrangement algorithms for 3D (which remove overlaps) [9]. The collective rearrangement algorithms are powerful because of the control that one has over final porosity and spatial correlation. In two dimensions, a slight decrease in the radii of spheres was necessary following the removal of overlaps to provide for flow paths. The flow computations described below do not require that all particle pairs be overlap free; however, in our case a slight modification would be required for the boundary discretization subroutine that was used.

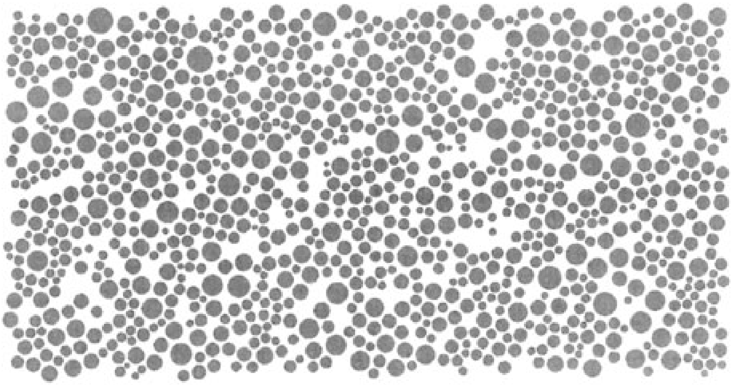

Figure 1. Example of a 1000-particle two-dimensional porous medium.

Within the model porous media, we wish to solve the Stokes equation for low-Reynolds-number flow. The Stokes equation and the continuity equation are

$$-\nabla p + \mu \nabla^2 \mathbf{u} = 0 \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

The Stokes equation is a time-independent boundary-value problem that can be solved using a variety of numerical techniques, given an appropriate combination of boundary velocities and stresses.

Finite rather than periodic boundary conditions are used because the domain models a piece of a larger heterogeneous medium; ultimately, the strategy would be to derive appropriate boundary conditions for the Stokes-flow problem from a large-scale simulation. However, in the problems shown here, we simply apply constant-pressure boundaries along two opposing sides of the domain, with no-flow boundaries parallel to this imposed pressure gradient. (The no-flow boundaries are shown in bold in subsequent figures.) Mathematically, the constant-pressure conditions at the inlet and outlet are imposed by forcing the tangential velocity component equal to zero along these boundaries, which thereby allows one to equate the normal stress to the imposed pressure. These conditions provide a well-posed global problem.

4. SOLUTION METHOD

4.1. Division of the pore space into subdomains

Most techniques for the direct numerical solution of Equation (1) are impractical in a large domain such as the one shown previously. Domain decomposition, which helps address this problem, is shown schematically in Figure 2; each subdomain can be viewed as a smaller

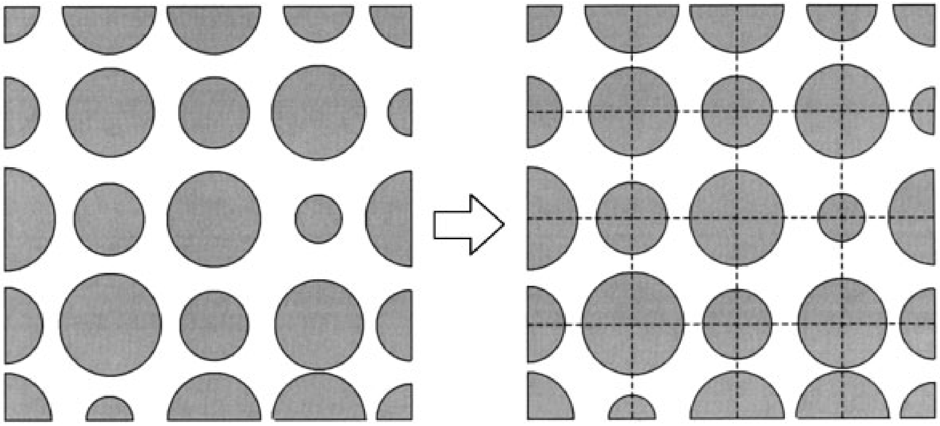

Figure 2. Schematic of domain decomposition on a regular grid, using particle centres as discretization points.

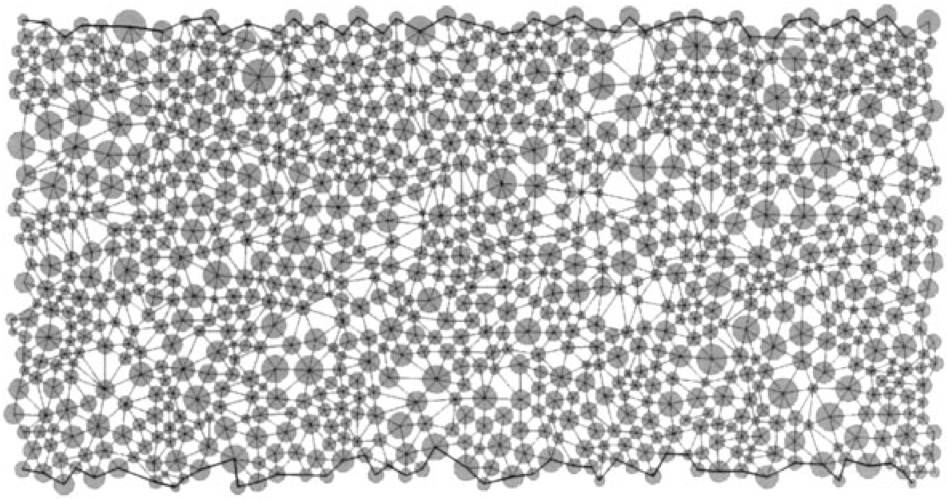

Figure 3. Delaunay tessellation of the random porous medium.

boundary-value problem. In the following descriptions, subdomains are defined by the fluid phase only, rather than the whole region within a dashed box in Figure 2. Hence, each subdomain will typically include sections of zero-velocity boundary (i.e. any particle surfaces) as well as sections of boundary within the fluid phase. Boundary conditions along interior fluid-phase boundaries are not known initially (i.e. on those boundaries created by the subdivision process), which means the flow problem is not immediately solvable in any one subdomain by itself. As such, the objective of the method is to determine these interior boundary values; once they are known, velocity can be found at an arbitrary point by solving the Stokes equation in that point's subdomain.

For simplicity in the schematic, Figure 2 was drawn as a regular structure, which immediately suggests the square sudivisions that are shown. A random array does not provide such an obvious strategy, but the schematic still illustrates some important considerations. One question is whether an advantage is gained by pinning the subdomain lattice to the particle centres. As it turns out, this strategy is beneficial because it helps to minimize the length of interior fluid-phase boundaries (as discussed later). In principle, however, the subdivision process is not restricted, which would be an important consideration if one were to work with media composed of arbitrarily shaped particles.

A Delaunay tessellation is a logical choice for initial discretization of a random domain. It is unique, it joins the particle centres, and the structure can be stored and accessed very efficiently. Figure 3 is an illustration of the Delaunay tessellation applied to the Figure 1 domain. Further refinement of the subdomain can be made by removing internal boundaries from the Delaunay tessellation, and the strategy for this removal process is one of the major points discussed in this paper. Figure 4 is a modified subdomain map that was created by removing the longest fluid-phase boundaries while constraining the maximum subdomain size to 25 particles.

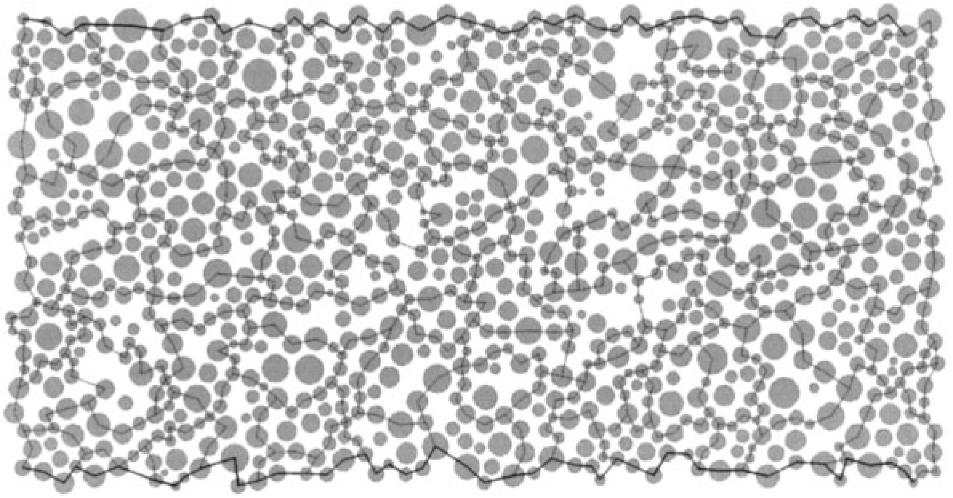

Figure 4. Final subdomain map, created by removing the largest internal fluid-phase boundaries from the Delaunay tessellation and limiting subdomain size to 25 particles.

4.2. Solution of the Stokes equations within the subdomains

A boundary integral method is used to solve Equations (1) and (2) within each subdomain. The numerical equations are derived from the following integral equation, written for a pole point $\mathbf{x}_0$ on a boundary of one of the subdomains, where S indicates the perimeter of the fluid phase in a subdomain [10]:

$$u_j(\mathbf{x}_0) = -\frac{1}{2\pi\mu}\int_S f_i(\mathbf{x})G_{ij}(\mathbf{x},\mathbf{x}_0)\,\mathrm{d}S + \frac{1}{2\pi}\int_S u_i(\mathbf{x})T_{ijk}(\mathbf{x},\mathbf{x}_0)n_k(\mathbf{x})\,\mathrm{d}S \tag{3}$$

The terms $\mathbf{f}$ and $\mathbf{u}$ are the stress and velocity along the boundary. The Stokeslet $\mathbf{G}$ and its associated stress tensor $\mathbf{T}$ depend on the boundary geometry (see Reference [10]). For numerical solution, the boundary integral is approximated using a series of N finite boundary elements:

$$u_j(\mathbf{x}_0) = -\frac{1}{2\pi\mu}\sum_{n=1}^{N} f_i^{(n)}\int_{S_n} G_{ij}(\mathbf{x},\mathbf{x}_0)\,\mathrm{d}S + \frac{1}{2\pi}\sum_{n=1}^{N} u_i^{(n)}\int_{S_n} T_{ijk}(\mathbf{x},\mathbf{x}_0)n_k(\mathbf{x})\,\mathrm{d}S \tag{4}$$

In this discrete form, the boundary values (stress and velocity) are assumed to be constant over each element and the integrals are evaluated analytically. In two dimensions, one typically must know two of the four boundary values on each element, while the boundary integral equations result in a set of linear equations for the remaining two unknowns. (The four boundary conditions referred to are two velocity plus two stress components.) If velocity is specified everywhere, then the equation can be solved only within an arbitrary pressure. We do not encounter this problem since pressure is specified over the inlet and outlet of the overall flow domain.

In a subdivided domain, (e.g. Figure 4), two boundary values (no-slip) are known along any solid–fluid boundary because of the no-slip condition. Internal fluid-phase boundaries have no

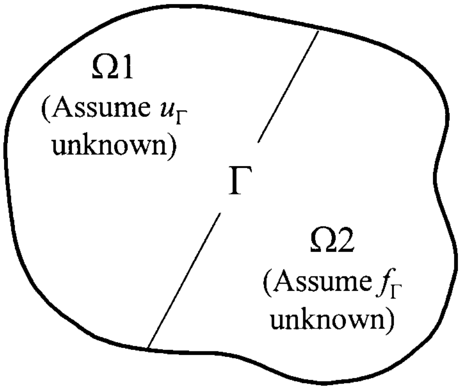

Figure 5. Hypothetical subdivided domain with split boundary conditions along the common boundary.

known boundary values, but are subject to four equations (i.e. the two vector components of Equation (4) written for the two neighbouring subdomains). Hence, the subdivided problem, like the original, is well-posed.

The global problem becomes one of solving for all unknown boundary values (two on some boundaries, four on others). Once these are determined, the velocity at an arbitrary point can be determined by computing a relatively small boundary integral defined by the point's associated subdomain. A slight variation of Equation (4) (i.e. with different constants) is used to calculate these internal velocities [10].

4.3. Structure and solution of the global problem

The structure of the coefficient matrix in the global problem impacts the method and efficiency of solution. Using the domain decomposition procedure described here, the matrix has a block-diagonal structure with sparse off-diagonal components. Terms in the blocks are integral coefficients for variables associated with the subdomain whose boundary is being integrated. Off-diagonal terms are integral coefficients for variables on that same boundary, but which have been assigned to neighbouring subdomains. Non-zero terms in the right-hand-side vector are from boundary integral terms where the velocity or stress is known.

The resulting block-diagonal structure is of considerable benefit for solution purposes. Although it arises naturally by setting up the boundary integrals one subdomain after another, the four unknowns on internal fluid-phase boundaries must be assigned with some care. Specifically, the blocks must remain square (because of the preconditioner used in our algorithm), and one must incorporate known stresses into each subdomain to avoid creating singular blocks. The logistics of these variable assignments are dealt with using a depth-first search [11].

The resulting matrix structure is shown schematically for the simple subdivided domain in Figure 5. Assume that stresses are known on the external boundaries of Ω1 and Ω2 (and that no-slip conditions exist somewhere in each subdomain). Assume also that unknown stresses on Γ are assigned to the Ω1 subdomain while unknown velocities on Γ are assigned to the

$\Omega 2$ subdomain. Then, the resulting block-diagonal matrix has the form

$$\begin{bmatrix} (1-\mathbf{T}_{11}) & \mathbf{G}_{1\Gamma} & -\mathbf{T}_{1\Gamma} & \\ -\mathbf{T}_{\Gamma 1} & \mathbf{G}_{\Gamma\Gamma} & (1-\mathbf{T}_{\Gamma\Gamma}) & \\ & -\mathbf{G}_{\Gamma\Gamma} & (1-\mathbf{T}_{\Gamma\Gamma}) & -\mathbf{T}_{\Gamma 2} \\ & -\mathbf{G}_{2\Gamma} & -\mathbf{T}_{2\Gamma} & (1-\mathbf{T}_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{u}_{S1} \\ \mathbf{f}_{\Gamma} \\ \mathbf{u}_{\Gamma} \\ \mathbf{u}_{S2} \end{bmatrix} = \begin{bmatrix} -\mathbf{G}_{11}\mathbf{f}_{S1} \\ -\mathbf{G}_{\Gamma 1}\mathbf{f}_{S1} \\ -\mathbf{G}_{\Gamma 2}\mathbf{f}_{S2} \\ -\mathbf{G}_{22}\mathbf{f}_{S2} \end{bmatrix} \qquad (5)$$

where S_1 and S_2 are the outer boundaries of the two subdomains, **G** and **T** are matrices containing the discretized boundary integrals, and **u** and **f** are vectors containing the discretized velocities and forces. The upper half of the matrix represents the boundary integral equations for $\Omega 1$, and the off-block terms (shown in the first two rows of column three) are from velocities along Γ, because $\mathbf{u}_\Gamma$ terms were assigned to $\Omega 2$.

A biconjugate gradient method with preconditioning is used to solve the system of equations [12]. The preconditioner is a multidomain multiplicative Schwarz method. The Schwarz method, which is equivalent to a block Gauss–Seidel technique, can sometimes be used iteratively to solve the system of equations. If it were applied in this way (using Equation (5) as an example), one would iterate in a two-step process, first solving

$$\begin{bmatrix} (1-\mathbf{T}_{11}) & \mathbf{G}_{1\Gamma} \\ -\mathbf{T}_{\Gamma 1} & \mathbf{G}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{S1} \\ \mathbf{f}_{\Gamma} \end{bmatrix} = \begin{bmatrix} -\mathbf{G}_{11}\mathbf{f}_{S1} + \mathbf{T}_{1\Gamma}\mathbf{u}_{\Gamma} \\ -\mathbf{G}_{\Gamma 1}\mathbf{f}_{S1} - (1-\mathbf{T}_{\Gamma\Gamma})\mathbf{u}_{\Gamma} \end{bmatrix} \qquad (6)$$

for $\mathbf{u}_{S1}$ and $\mathbf{f}_\Gamma$, and then solving

$$\begin{bmatrix} (1-\mathbf{T}_{\Gamma\Gamma}) & -\mathbf{T}_{\Gamma 2} \\ -\mathbf{T}_{2\Gamma} & (1-\mathbf{T}_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\Gamma} \\ \mathbf{u}_{S2} \end{bmatrix} = \begin{bmatrix} -\mathbf{G}_{\Gamma 2}\mathbf{f}_{S2} + \mathbf{G}_{\Gamma\Gamma}\mathbf{f}_{\Gamma} \\ -\mathbf{G}_{22}\mathbf{f}_{S2} + \mathbf{G}_{2\Gamma}\mathbf{f}_{\Gamma} \end{bmatrix} \qquad (7)$$

for $\mathbf{u}_\Gamma$ and $\mathbf{u}_{S2}$. This approach is non-convergent for the larger problems that we tested. However, the Schwarz method still proved to be a good preconditioner for these cases. When used as a preconditioner, the coefficient matrices in Equations (6) and (7) are inverted and applied to blocks of the global matrix during iterations of the biconjugate gradient method. (Specifically, the equations $Ky = p_i$ and $Kz = s$ appear in the preconditioned BI-CGSTAB algorithm in Reference [12]; we approximate y and z by performing a single pass of the Schwarz method, where K is the entire block-diagonal matrix.)

5. RESULTS AND ANALYSIS

5.1. *Format of the solution*

The solution consists of all boundary values that were originally not known: velocities over the inlet and outlet, stresses on all particle surfaces, stresses on no-flow boundaries, and both velocity and stress on the internal fluid-phase boundaries. One advantage of this approach is that the final data set, which might be stored for subsequent stochastic simulations, is very

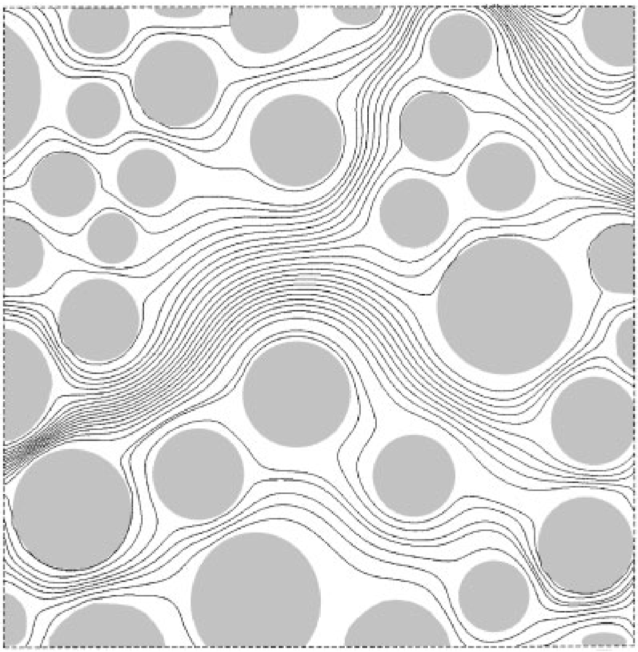

Figure 6. Streamlines in a small region of the 1000-particle domain.

compact, requiring only the pore and subdomain structure, the boundary discretization, and the boundary values. The latter two data sets are relatively small because of the reduction in dimension associated with the boundary integral method.

Once the global solution is found (or loaded from a previous run), the velocities at arbitrary points in the fluid phase can be recovered by calculating local boundary integrals. Because analytic expressions for the integrals in Equation (4) are available, this calculation is quite efficient and works well when coupled with stochastic algorithms for solute transport.

Figure 6 shows a streamline plot within a small area of the larger porous medium. This plot was generated by recovering velocities in the manner described above and integrating along a streamline using Euler's method.

5.2. Effect of subdomain structure

For a random packing there exists a unique Delaunay tessellation (in the absence of degeneracies and/or numerical error). The final subdomain structure, on the other hand, is not unique since it is generated by removing internal boundaries from the original Delaunay tessellation. The strategy used to create this map is of critical importance for the convergence behaviour.

Our results show that the most effective strategy is to remove internal boundaries that span the largest fluid-phase gaps (thus leaving internal boundaries across the smallest particle–particle gaps). At first, removal of large internal boundaries will create more 'pore-like' subdomains because many boundaries in the tessellation span large void spaces (see Figure 3).

As removal continues, the subdomains tend to become large convoluted shapes that no longer resemble pores, and the remaining internal boundaries span only the smallest particle–particle gaps in the domain (see Figure 4).

The effect that this removal strategy has on convergence is related to the generation of off-block matrix coefficients. Specifically, all internal fluid-phase boundaries generate off-block coefficients because of the boundary-value assignments described above, and the preconditioner does not operate above the diagonal (or more precisely, above the blocks). In essence, removing an internal fluid-phase boundary eliminates a set of unknowns that appear explicitly in the global problem. Their effect is still felt implicitly in the boundary integral equations and therefore in the structure of the blocks. This implicit presence seems to be preferable since blocks are inverted directly during preconditioning.

The suggestion made above, that subdomains should be created by removing larger boundaries, is illustrated quite dramatically by comparing the two subdomain maps shown in Figure 7. Both maps are generated on the identical 81-particle porous medium (very slightly disordered), and both maps contain 64 subdomains. The boundary-element discretization was performed so as to generate the same number of equations and the same number of off-block coefficients in both cases. However, convergence was achieved in 19 iterations for the case shown in Figure 7(a) versus 75 iterations when the longer internal boundaries were chosen (Figure 7(b)).

The equivalent structure of the matrices in the above example suggests that conditioning is the dominant factor in solution efficiency. This effect can be quantified using the six-particle flow cell shown in Figures 8(a) and 9(a). Like the larger example, it can be subdivided in the two different ways shown. The small size of this problem allows us to set up the two-block global matrix equation and examine the conditioning directly. Figures 8(b) and 8(c) show eigenvalues for the short-boundary case before and after preconditioning using the Schwarz method. Figures 9(b) and 9(c) show eigenvalues for the long-boundary case before and after preconditioning. The condition numbers remain large for both matrices after preconditioning. However, the eigenvalues are more closely clustered around $\lambda = 1$ for the short-boundary case, which improves the convergence of Krylov-subspace-based iterative techniques [13].

5.3. Effect of subdomain size on convergence

The other aspect of boundary removal is the size of the resulting subdomains. Larger subdomains result in larger but fewer blocks in the coefficient matrix, which one would expect to affect solution times. We quantify the size of these blocks using the number of particles per subdomain (either on or inside the boundary) for a couple of reasons. First, this parameter is used to monitor and control subdomain size in the boundary-removal subroutine. Second, since the boundary element method requires a boundary discretization, the number of particles per subdomain is more or less proportional to the number of equations per block that end up in the matrix.

Figure 10 shows the number of iterations (for convergence) versus subdomain size for a 100-particle random medium. The sharp improvement in convergence for small subdomains (i.e. changes from three particles with the Delaunay tessellation to four or five particles per subdomain) is observed for two reasons. First, long fluid-phase boundaries are removed (which contributes to the improved convergence and fewer equations). Second, there appears to be

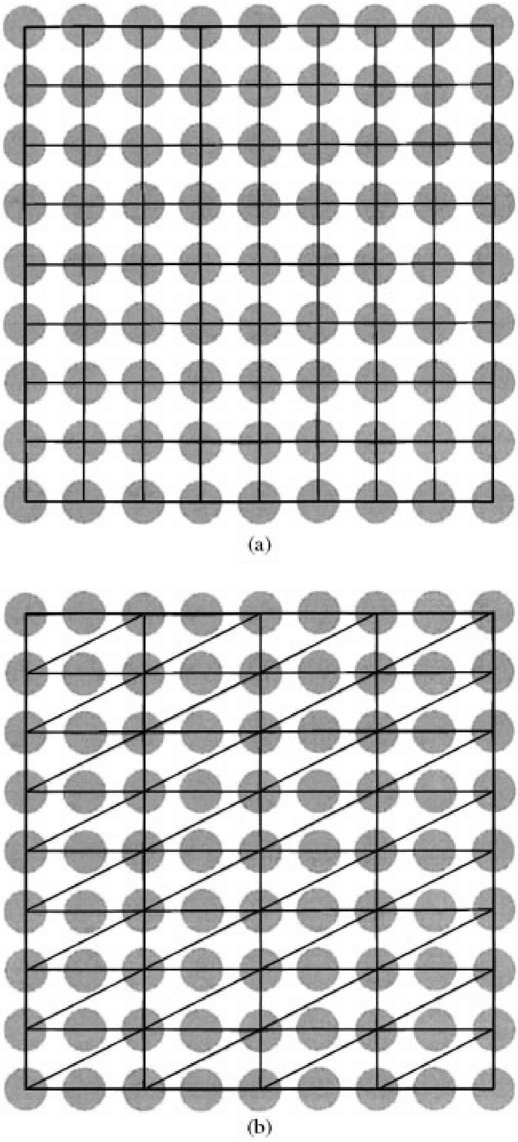

Figure 7. Eighty-one-particle domain: (a) short interior subdomain boundaries; and (b) long interior subdomain boundaries.

a beneficial effect that occurs because larger pieces of the flow problem are being solved directly during the preconditioning step. (Recall that blocks are inverted for preconditioning, which is equivalent to directly solving subdomain flow problems using current guesses for boundary values.)

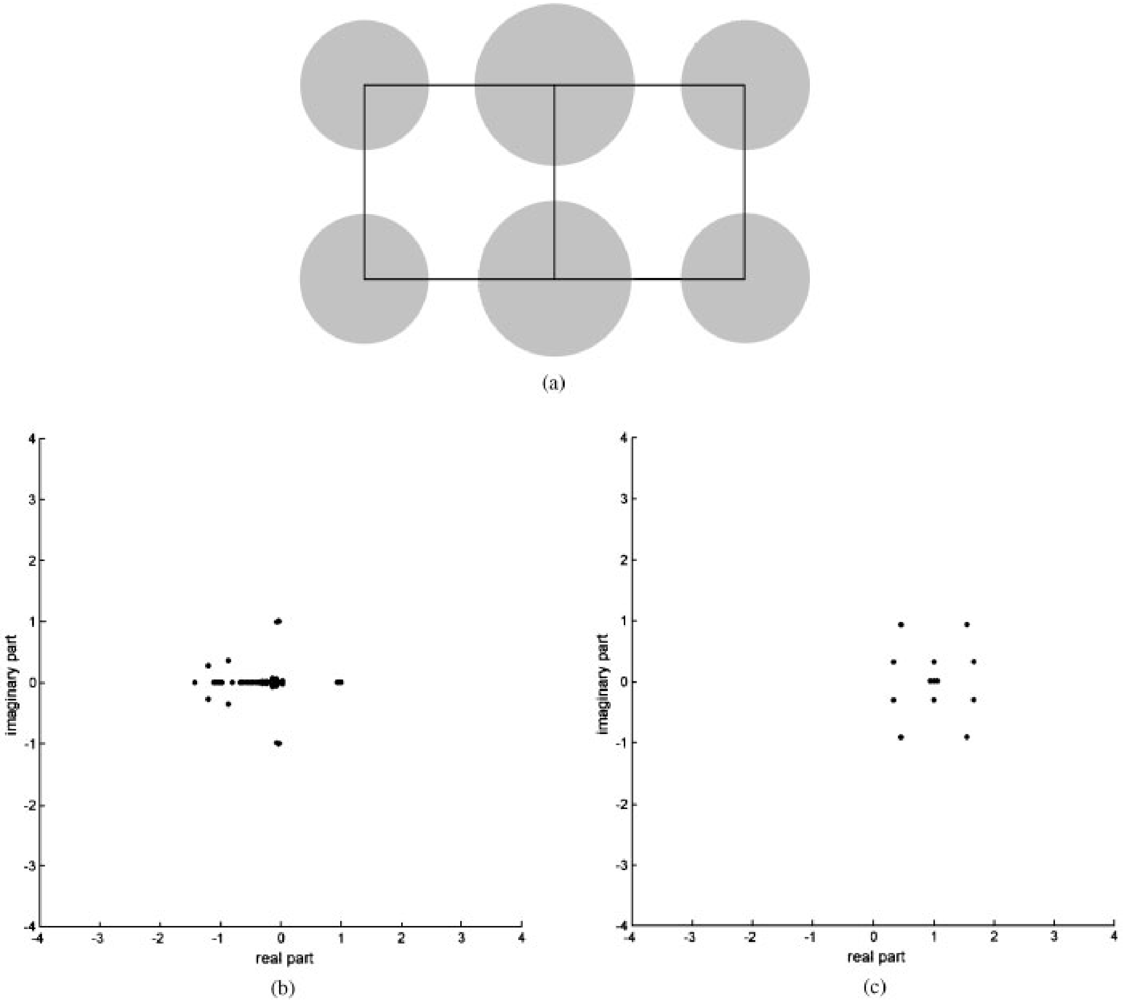

Figure 8. (a) Six-particle domain with short subdomain boundary; (b) eigenvalue distribution for block matrix before preconditioning; and (c) eigenvalue distribution for block matrix after preconditioning.

5.4. Overall efficiency

The overall efficiency is more complicated to analyse because it is related not only to subdomain size and shape, but also to practical issues such as memory limits and whether parallel processing is used.

Consider first the overall efficiency in the absence of memory limits, assuming sequential rather than parallel processing (single-processor computations have been used in this work). The overall time for solution is consumed mostly by two operations: the inversion of all blocks (to be used for subsequent preconditioning steps) and the iterative solution process. Inversion of the blocks in the matrix must be performed only once, and the time requirement

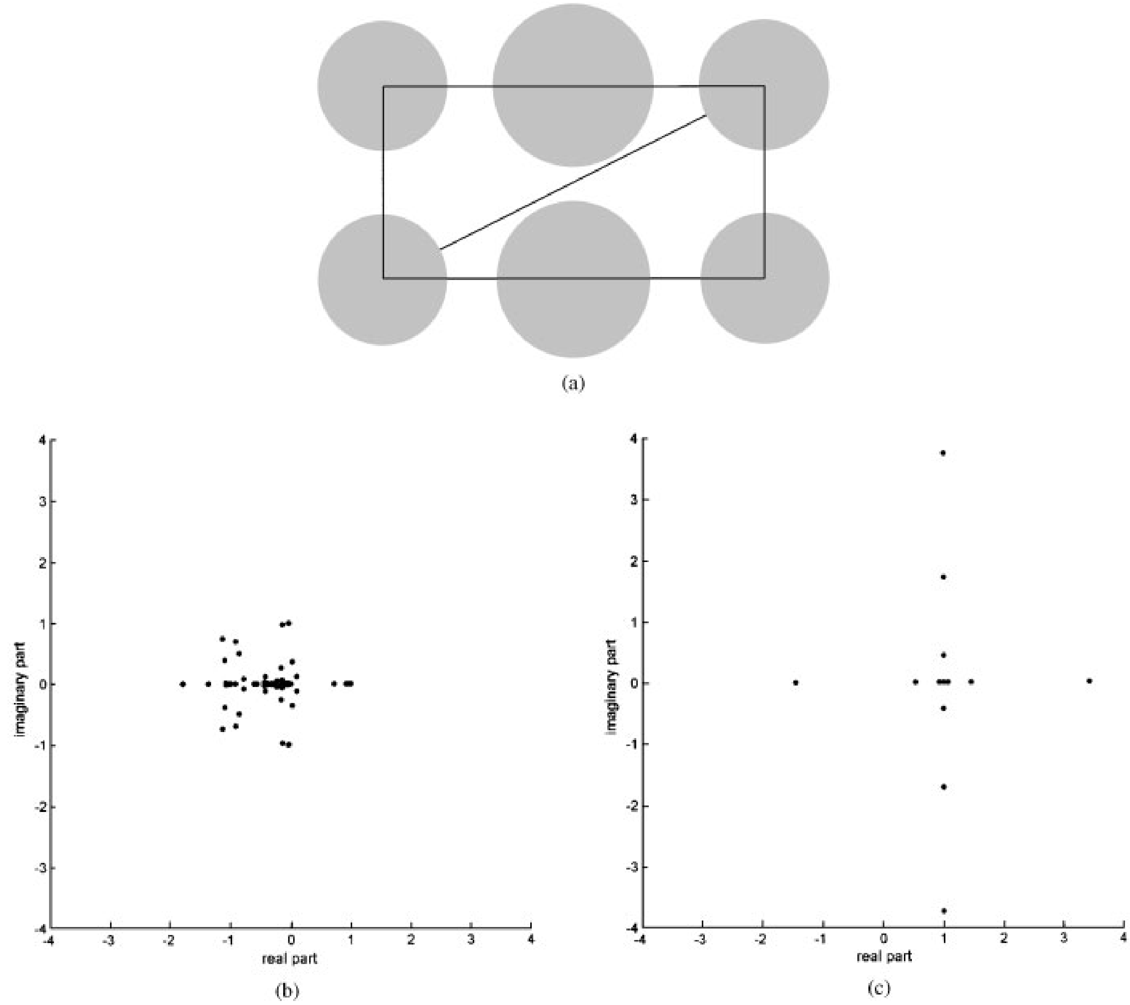

Figure 9. (a) Six-particle domain with long subdomain boundary; (b) eigenvalue distribution for block matrix before preconditioning; and (c) eigenvalue distribution for block matrix after preconditioning.

is proportional to N^3, where N is the number of equations per block (which we assume for the moment to be constant). Once the inversions are complete, the iterative part of the solution requires an additional time

$$t_i = \text{number of iterations} \times \text{number of subdomains} \times \text{calculation time per subdomain}$$

The calculation time per subdomain scales as N^2 for the vector operations in the biconjugate gradient algorithm. However, this penalty is essentially offset by two competing factors: the improved efficiency (i.e. fewer iterations) as the blocks become larger and the decrease in the total number of subdomains on which to operate. There is a rather complicated relationship

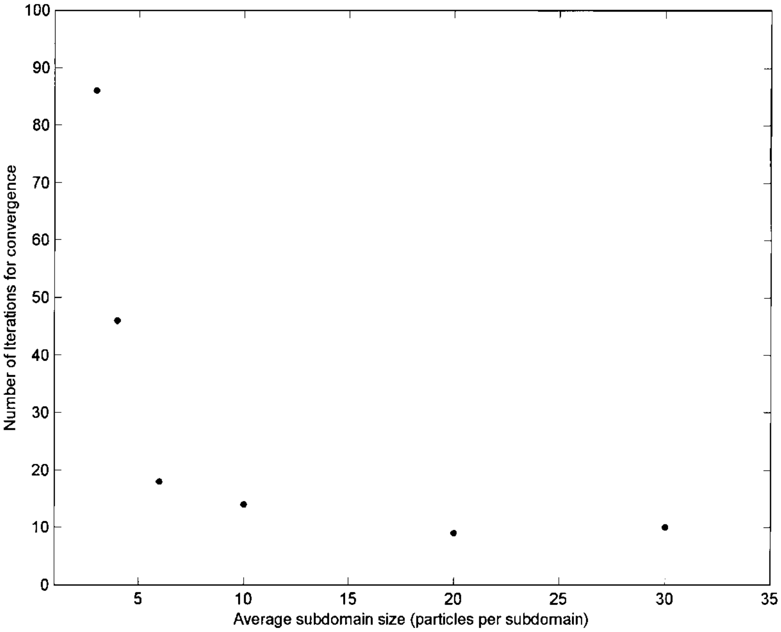

Figure 10. Number of iterations for convergence versus subdomain size for a 100-particle random domain.

between subdomain size (in the number of particles) and the total number of subdomains in the system. Hence, the efficiency and number of subdomains were measured empirically from our calculations.

These empirical data were combined with the scaling arguments to produce the plot of approximate solution time versus subdomain size shown in Figure 11. The y-axis is non-dimensionalized using the hypothetical time that would be required with only a single subdomain (i.e. inversion of the entire full matrix). For small subdomains, a rapid improvement with size occurs because of two effects: first, the removal of large internal boundaries (as discussed above), and second, changes in subdomain shape (e.g. from two triangles to a square), which often reduce the total perimeter and therefore the number of equations considerably. However, for larger subdomains, performance becomes slower with increasing size because as two large subdomains are combined, the improvement in efficiency and the very small reduction in the total number of equations cannot offset the cost of doubling the block size (which is a one-time N^3 penalty for preconditioning plus a repeated N^2 penalty during iteration). The net effect of these factors produces an optimum theoretical performance at around six particles per subdomain for the 100-particle domain used in this example. (To provide a computational benchmark, the 1000-particle solution using six-particle subdomains required 13 264 s CPU time on a 600 MHz PC. The wall clock time was much longer due to memory limitations discussed below.)

Recalling that the above arguments neglected memory limitations, they must be tempered with practical considerations. In the single-processor solutions performed in our work, the

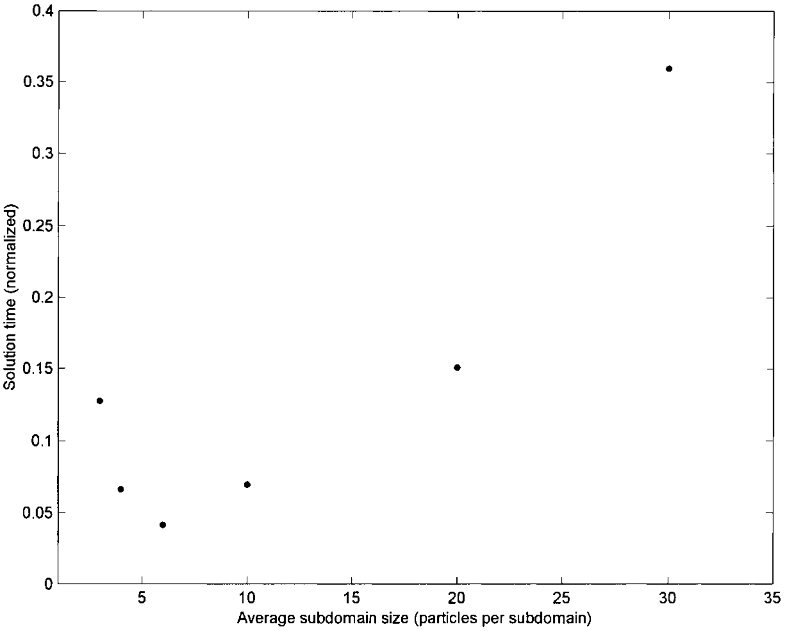

Figure 11. Theoretical solution time versus subdomain size for a 100-particle random domain.

non-zero matrix coefficients as well as the inverted blocks are stored on disk because of their large size, and repeated retrieval of these data dominates the total solution time (90–95%). Consequently, we observe continued improvements in actual computation time well past the subdomain size shown in Figure 11, because it is more efficient to read coefficients from a large block and its inverse at once, whereupon they are used for multiple vector operations in the biconjugate gradient method. An optimum still occurs in these single-processor calculations at the point where data associated with a single block exceed the random-access memory of the processor, and it begins using virtual memory.

For parallel processing (which is being used for future work), the considerations are slightly different. Uniformity of subdomain size and the number of subdomains should be considered. With sufficiently large distributed memory, optimum performance should occur nearer the theoretical minimum (calculated for a given problem) than it did in our computations.

5.5. *Accuracy and mesh refinement*

One of the most attractive aspects of this algorithm is its amenability to local mesh refinement. This approach can be used to increase the resolution in critical parts of a large domain (e.g. in very tight particle–particle gaps, or in the vicinity of a specific particle in which detailed hydrodynamics are being studied), while imposing less-stringent accuracy criteria elsewhere.

Of specific interest is the error at internal subdomain boundaries because the subdivisions create sharp corners in the boundary structure that would not exist otherwise, and errors are larger near corners unless special methods are used [14]. To evaluate this error, we computed the flowrate across internal subdomain boundaries (by integrating velocity along

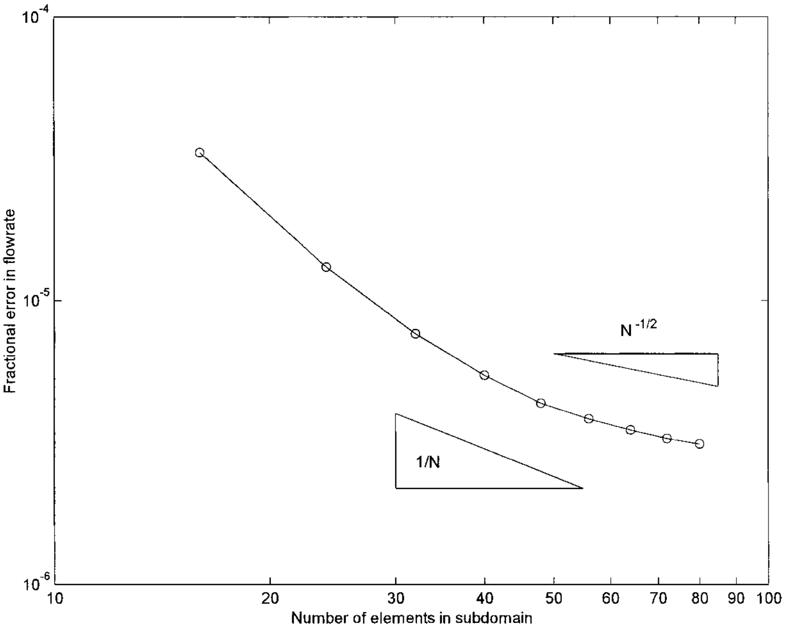

Figure 12. Error in flowrate across an internal subdomain boundary during grid refinement.

the boundary) for increasing numbers of elements in the subdomain. Flowrates in the same gaps were calculated after removing the internal subdomain boundaries and using a high level of grid refinement, and this value was treated as exact. Figure 12 shows the fractional error in flowrate versus the number of elements. The rate of convergence is rapid in the range where truncation error is being improved. However, the rate tails off at higher resolutions, which, we suspect, occurs when roundoff errors and/or error from the iterative matrix solution become significant in relation to the truncation error.

A second consideration is the scaling of convergence time with increasing number of elements. Our tests show that for a given subdomain structure, increasing the number of elements per subdomain has no effect on the number of iterations during the solution of the system of linear equations. Hence, solution time scales only according to the vector operations in the preconditioner and biconjugate gradient method (N^2 for the algorithms used here).

6. SUMMARY AND CONCLUSIONS

While lattice-Boltzmann methods have become the method of choice for solving low-Reynolds-number flow problems in unstructured porous materials, efficient algorithms for direct numerical solution may have advantages in certain situations. The technique discussed here combines important attributes of domain decomposition and boundary-integral methods. The approach is used on a comparatively large, very heterogeneous problem, which demonstrates its flexibility and efficiency.

Numerically, the method gives a largely block-diagonal matrix that is solved using a bi-conjugate gradient method with block preconditioning. We show that the subdomain structure strongly affects the solution efficiency. In particular, minimizing the length of subdomain boundaries in the fluid phase is important. This effect is related to the generation of off-block terms in the coefficient matrix, some of which are not operated upon by the block preconditioner. Subdomain size is the second important parameter. We show, using scaling arguments and empirical results, that an optimum subdomain size exists (in the absence of memory limitations). Our runs were performed on a single processor; consequently, data retrieval time dominated the solution time and the optimum subdomain size was much larger than the predicted optimum.

The large problems required relatively few iterations for convergence, given the very large matrix sizes (e.g. 45 iterations for the 1000-particle problem shown above, which produced $\sim 5\times 10^5$ equations). We expect that in the parallel framework being used in future work, iterations for this size problem will require to be of the order of 10–20 s. Hence, the overall time requirements for solution are very reasonable.

A couple of other issues are of interest in the context of stochastic modelling of transport processes (for which these simulations are being used). First, the complete solution set can be stored very compactly since it consists of only the boundary discretization and boundary values. Second, the recovery of point velocities is efficient since it requires only a local boundary-integral calculation once the global solution is known. Third, local grid refinement is trivial in this algorithm because of the one-dimensional discretization. This allows one to obtain very accurate velocities in critical regions of the domain such as tight particle–particle gaps or near solid–fluid interfaces undergoing heat or mass transfer.

REFERENCES

1. Guo G, Thompson KE. Experimental analysis of local mass transfer in packed beds. *Chemical Engineering Science* 2001; **56**:121–132.
2. Spaid MAA, Phelan FR Jr. Lattice Boltzmann methods for modelling microscale flow in fibrous porous media. *Physics of Fluids* 1997; **9**:2468–2474.
3. Sangani AS, Acrivos A. Slow flow past periodic arrays of cylinders with applications to heat transfer. *International Journal of Multiphase Flow* 1982; **8**:193–206.
4. Larson RE, Higdon JJL. Microscopic flow near the surface of two-dimensional porous media. Part 2. Transverse flow. *Journal of Fluid Mechanics* 1987; **178**:119–136.
5. Sangani AS, Yao C. Transport processes in random arrays of cylinders. II. Viscous flow. *Physics of Fluids* 1988; **31**:2435–2444.
6. Sangani AS, Mo G. Inclusion of lubrication forces in dynamic simulations. *Physics of Fluids* 1994; **6**:1653–1662.
7. Koch DL, Ladd AJC. Moderate Reynolds number flows through periodic and random arrays of aligned cylinders. *Journal of Fluid Mechanics* 1997; **349**:31–66.
8. Verberg R, Ladd AJC. Simulation of low-Reynolds-number flow via a time-independent lattice-Boltzmann method. *Physical Review E* 1999; **60**:3366–3373.
9. Liu G, Thompson KE. Influence of computational domain boundaries on internal structure in low-porosity sphere packings. *Powder Technology* 2000; **113**:185–196.
10. Pozrikidis C. *Boundary Integral and Singularity Methods for Linearized Viscous Flow.* Cambridge University Press: New York, 1992.
11. Chartrand G, Oellermann OR. *Applied and Algorithmic Graph Theory*. McGraw Hill: New York, 1993; 68–85.
12. Van Der Vorst HA. BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Statistical Computing* 1992; **13**:631–644.
13. Trefethen LN, Bau D III. *Numerical Linear Algebra.* SIAM: Philadelphia, PA, 1997; 266–275.
14. Brebbia CA, Telles JCF, Wrobel LC. *Boundary Element Techniques.* Springer: Berlin, 1984; 137–138.