

EVALUATION OF GRID-BASED AND GRID-FREE METHODS TO MODEL MICROCHANNEL TRANSPORT-REACTION*

K. L. HAMLINGTON[†], Y.-Y. KWAN[‡], H. FUJIOKA[‡], R. CORTEZ[‡], AND
D. P. GAVER III[†]

Abstract. The goal of this paper is to investigate the spectral element method (SEM), a grid-based computational approach, and the grid-free particle strength exchange method (PSEM) for solving the advection-diffusion-reaction equation in a microfluidic channel. We identify the dimensionless parameter regimes at which each algorithm provides a satisfactory solution in terms of accuracy and computational time. We also compare the velocity field computed in microchannel geometries for nonzero and zero Reynolds numbers by solving the full Navier–Stokes equations with the SEM and solving the inertia-free Stokes equations with the boundary element method, which does not require an internal mesh. The methods discussed may be utilized in any appropriate combination to solve high and low Péclet number and zero and nonzero Reynolds number transport problems to allow fast, accurate evaluation of a binding reaction between two species. We show that the grid-free particle and boundary element methods are suitable for solving the convection-dominated, irregular geometry problems expected for microfluidic applications where the Reynolds number is near zero and the Péclet number is very high. The SEM transport solver is appropriate for cases in which diffusion plays a greater role and the concentration gradients are not as steep. These methods may be applied to the design and optimization of microchannel geometries for mixing and other microfluidic applications.

Key words. spectral element, boundary element, particle strength exchange, advection-diffusion-reaction, microfluidic device design

AMS subject classifications. 65M38, 65M70, 65M75, 65Z05

DOI. 10.1137/120880598

1. Introduction. Computational modeling of the advection-diffusion-reaction equation plays an important role in the design of microfluidic components and elucidation of the underlying transport processes at microscale dimensions (100 nm–100 μm). A particular application for which modeling is useful is the design of a microfluidic channel for a miniaturized antibody-based sensor. This type of sensor requires the combination of an antibody solution and its corresponding analyte solution to facilitate a competitive-binding reaction within the microchannel as the solutions flow through the device. The reaction product formation is transduced to a detectable signal that indicates the concentration of an environmental contaminant, pathogen, or molecule of clinical interest [30]. At microscale dimensions, the fluid dynamics and transport are characterized by the vanishingly small Reynolds numbers ($\text{Re} = \rho UL/\mu$) of the laminar flow regime and the high Péclet numbers ($\text{Pe} = UL/D$) of convection-dominated mass transport due to the small diffusivities of the antibody and analyte. For example, if we assume aqueous solutions with density $\rho = 1 \text{ g/cm}^3$ and viscosity $\mu = 1 \text{ cP}$, a mean velocity $U = 10^{-2} \text{ cm/s}$, and length scale $L = 10^{-2} \text{ cm}$, the Reynolds

*Submitted to the journal's Computational Methods in Science and Engineering section June 11, 2012; accepted for publication (in revised form) April 29, 2013; published electronically July 11, 2013. This work was supported by NSF EPS-0701491, NSF CBET-1033619, NSF-RTG (DMS-1043626), NIH R01-HL81266, and the Louisiana Board of Regents LONI Institute PKSFI project.

<http://www.siam.org/journals/sisc/35-4/88059.html>

[†]Department of Biomedical Engineering, Tulane University, New Orleans, LA 70118 (khamling@tulane.edu, dpg@tulane.edu).

[‡]Center for Computational Science, Tulane University, New Orleans, LA 70118 (tonykwan510@gmail.com, fuji@tulane.edu, rcortez@tulane.edu).

number for this system is $\text{Re} = 10^{-2}$. Estimating the molecular diffusivities of an antibody and analyte that may be used in a microfluidic sensor as $D_1 = 2 \times 10^{-8} \text{ cm}^2/\text{s}$ and $D_2 = 2 \times 10^{-7} \text{ cm}^2/\text{s}$, respectively, the Péclet numbers are $\text{Pe}_1 = 5000$ and $\text{Pe}_2 = 500$. Therefore, the combination of the two solutions occurs only by molecular diffusion between adjacent streams, requiring an impractical amount of time and channel length for this application. To passively combine the antibody and analyte solutions without moving parts or actuators [42], we consider geometric modifications to the microchannel to increase transverse diffusion and advection, as well as a range of transport parameter values, $0 \leq \text{Re} \leq 50$ and $10 \leq \text{Pe} \leq 5000$. Convection-dominated behavior associated with $\text{Pe} \gg 1$ leads to steep concentration gradients, and modeling these systems using grid-based techniques such as the finite element method necessitates a fine and/or adaptive mesh and very small time steps to accurately resolve the evolving concentration field [21]. Considering an irregular domain as well, a grid-based computational model may be too computationally expensive to complete.

In this paper, we evaluate two types of computational methods for simulating the advection-diffusion-reaction equation to assess the reaction between two species in a microfluidic channel: (1) the spectral element method (SEM), a grid-based approach, and (2) the grid-free particle strength exchange method (PSEM) coupled with the boundary element method (BEM). Our goal is to identify the dimensionless parameter regimes over which each algorithm provides a satisfactory solution of the advection-diffusion-reaction equation. We provide a comparison of the velocity field in microchannel geometries computed for nonzero and zero Re by solving the full Navier–Stokes equations with the SEM and solving the inertia-free Stokes equations with the BEM. We show that the methods discussed may be utilized in any appropriate combination to solve high and low Pe and zero and nonzero Re transport problems for microfluidic applications.

1.1. Spectral element method (SEM). The SEM is a high-order finite element method that was first introduced by Patera [35]. The SEM has been used extensively to solve problems involving flow, such as the Navier–Stokes equations [13] and the advection-diffusion equation [3, 31, 36, 46]. Compared to the traditional low-order finite element method, the SEM often results in a more computationally efficient approach because (1) fewer elements are needed because of the higher accuracy, and (2) the number of unknowns can be reduced by applying a substructuring technique to eliminate the unknowns in the interior of elements. The latter is important as the number of unknowns can be reduced from $\sim h^{-d}N^d$ to $\sim h^{-d}N^{d-1}$, where h is the average element size, d is the space dimension, and N is the polynomial degree used. In the SEM, the linear system resulting from the discretization of the differential equation is usually solved by a preconditioned iterative method, and a number of preconditioners have been proposed in the literature [23, 32, 39]. For equations with a time derivative, such as the advection-diffusion equation, we have found that the Jacobi preconditioner usually works well. The computational code we used supports preconditioned conjugate gradient with different preconditioners, including Jacobi, overlapping Schwarz, and balancing domain decomposition methods by constraints (BDDC). More details of these preconditioners can be found in [27].

1.2. Particle strength exchange method (PSEM). To simulate transport processes, we also implement the PSEM, a grid-free Lagrangian method, in which a system of ordinary differential equations (ODEs) is solved to simulate the evolution of the concentration of each substance as particle trajectories follow the local flow

field. A smooth “blob” function is associated with each particle, which overlaps to neighboring particles to provide a smooth representation of the concentration field. Particle methods have been extensively developed and improved for simulating the Navier–Stokes equations [8, 10, 26, 29] and have been applied to advection-diffusion systems [2, 9, 12, 40]. This grid-free approach is advantageous for our advection-dominated application because it is better able to deal with the sharp gradients encountered and less likely to introduce artificial diffusion than with fixed-grid numerical schemes. The PSEM benefits from relatively simple implementation, and recent work has demonstrated execution with parallelization techniques for faster processing time [6]. Because the particles advect to adapt to the flow map, the PSEM is capable of handling complex geometries with irregular boundaries and obstacles. However, this adaptability creates particle distortion in nonuniform flows or regions of high shear such that the particle blob functions no longer overlap and the equations are not solved to the expected degree of accuracy [25]. A redistribution step restores the regular spacing of particles by interpolating the particle concentrations at the new positions. An additional drawback of particle methods is the difficulty of enforcing boundary conditions since boundaries do not exist in the particle calculation. As discussed in section 3.2.2, we implement an image particle technique that reflects particles symmetrically about the boundary to impose a Neumann condition, such that the concentration of the image particle is identical to its in-domain counterpart. In the case when a Dirichlet condition is prescribed, the concentration associated with these image particles may be computed by solving a linear system of equations that enforces the appropriate concentration value at the boundary, as was done in [47].

1.3. Boundary element method (BEM). When $\text{Re} \ll 1$, which is typical in microfluidic systems, we use the BEM based on the boundary integral representation of Stokes flow derived by Ladyzhenskaya [28] to solve the Stokes equations for the velocity field in microchannel domains. The BEM is advantageous particularly for complicated domains because there is no internal mesh, reducing a two-dimensional problem to one dimension. Our BEM implementation, which has been rigorously validated and adapted from a version used extensively in simulations of pulmonary airway reopening [14, 16, 41] and flow over cells in a channel [15], is solved efficiently with parallelization for complicated geometries with fine resolution on the boundaries. The approach for the two-dimensional formulation described in this paper is similar to that of Higdon [18], wherein the boundary integrals are expressed in terms of traction and velocity, as opposed to biharmonic formulations involving the stream function and vorticity equation [22]. Disadvantages of our BEM approach include singularities in the boundary integral solution and the formation of large dense matrices that require significant computational memory. We overcome these difficulties by using special quadrature rules and by implementing the method using multiple CPUs as described in section 3.1.2.

The remaining sections of the paper are organized as follows. In section 2, we review the governing equations and boundary conditions of the model system. We describe the implementation of the SEM and BEM to solve the Navier–Stokes and Stokes equations, respectively, and the SEM and PSEM to solve the advection-diffusion-reaction in section 3. The computational results are presented in section 4. In section 4.1, we establish the accuracy of the transport algorithms by comparison to an analytical solution of the Taylor dispersion problem and use this solution to profile the computational efficiency of the SEM and PSEM for high and low Péclet numbers. In section 4.2, we compare the velocity profiles computed in two example microchannel

geometries with the Navier–Stokes (SEM) and Stokes (BEM) solvers. The advection-diffusion-reaction solutions in the two geometries are computed with the SEM and PSEM for a range of Péclet and Reynolds numbers in section 4.3, and we evaluate the amount of reaction product formed in each case. We show that the grid-free particle method coupled to the BEM works better for the convection-dominated, irregular geometry problems expected for microfluidic applications where the Reynolds number is assumed zero and the Péclet number is high. We demonstrate that the SEM transport solver is better suited for cases in which diffusion plays a greater role and the concentration gradients are not as steep.

2. Governing equations and boundary conditions. The motion of the fluid is governed by the incompressible Navier–Stokes equations

$$(2.1) \quad \text{Re} \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla^2 \mathbf{u} - \nabla P, \quad \nabla \cdot \mathbf{u} = 0,$$

where $\mathbf{u} = (u_1, u_2)$ and ∇P are the fluid velocity and pressure gradient nondimensionalized according to the length L , velocity U , and pressure $\Pi = \mu U/L$ scales, and $\text{Re} = \rho LU/\mu$ is the Reynolds number with ρ and μ the density and viscosity of the fluid, respectively. The indices 1 and 2 refer to the x - and y -directions in Cartesian coordinates. The Reynolds number is typically small in microfluidic systems, and the Stokes flow approximation of $\text{Re} \rightarrow 0$ may be appropriate. We will investigate the accuracy of this approximation in model geometries in section 4.2, where we solve the time-dependent (2.2) and steady-state (2.3) Stokes equations:

$$(2.2) \quad \frac{\partial \mathbf{u}}{\partial t} = \nabla^2 \mathbf{u} - \nabla P, \quad \nabla \cdot \mathbf{u} = 0,$$

$$(2.3) \quad \nabla^2 \mathbf{u} = \nabla P, \quad \nabla \cdot \mathbf{u} = 0.$$

The transport and reaction of each concentration c_i is governed by the dimensionless advection-diffusion-reaction equation:

$$(2.4) \quad \frac{\partial c_i}{\partial t} + (\mathbf{u} \cdot \nabla) c_i = \frac{1}{\text{Pe}_i} \nabla^2 c_i + R_i,$$

where $\text{Pe}_i = LU/D_i$ is the Péclet number for concentration c_i with diffusion coefficient D_i , and R_i is the term describing the reaction kinetics of the concentrations.

The domains Ω_A and Ω_B used for the computations in sections 4.2 and 4.3 are shown in Figure 2.1. The boundary of the domain is composed of three inlets and one outlet connected by solid channel walls: $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{wall}} \cup \Gamma_{\text{out}}$. Unattached solid-walled obstructions may also be included within the domain, but such geometries are not discussed here. No slip and no penetration boundary conditions for velocity are imposed on the channel walls, while the no-flux boundary condition is enforced for the concentrations:

$$(2.5) \quad \text{On } \Gamma_{\text{wall}} : \mathbf{u} = \mathbf{0},$$

$$(2.6) \quad \text{On } \Gamma_{\text{wall}} : \frac{\partial c_i}{\partial n} = 0.$$

A Dirichlet condition is prescribed for both the velocity and concentrations between the lower and upper boundaries of each inlet Γ_{in} . We assume that the outlet is long

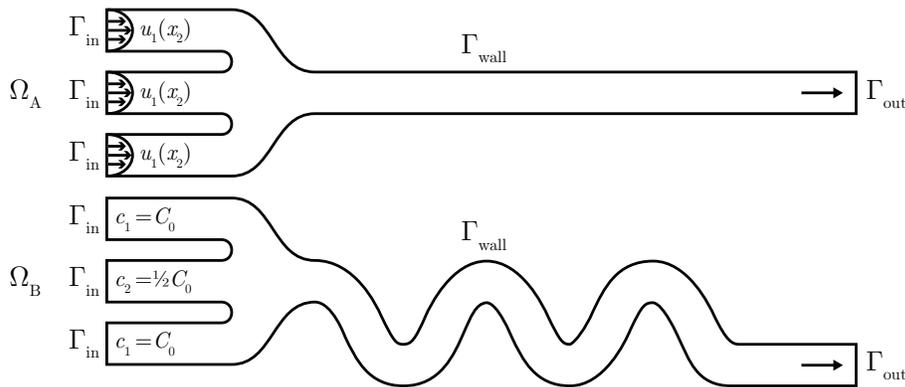


FIG. 2.1. Computational domains. The advection-diffusion-reaction equation is solved in straight (Ω_A) and serpentine (Ω_B) domains of length 18 with three inlets (Γ_{in}) and one outlet (Γ_{out}) of unit width. A parabolic velocity profile and Dirichlet concentration condition are prescribed at the inlets.

enough such that there is no tangential velocity along Γ_{out} , and the normal component of the velocity follows the Neumann boundary condition:

$$(2.7) \quad \text{On } \Gamma_{out} : \mathbf{u} \cdot \mathbf{t} = 0, \quad \frac{\partial \mathbf{u} \cdot \mathbf{n}}{\partial n} = 0,$$

where \mathbf{n} is the unit outward normal vector and \mathbf{t} is a unit vector tangential to the boundary. Early on, when the concentration front is far away from the outlet, the concentrations satisfy the homogeneous Dirichlet boundary condition. Otherwise, the Neumann boundary condition is used:

$$(2.8) \quad \text{On } \Gamma_{out} : c_i = 0 \quad \text{or} \quad \frac{\partial c_i}{\partial n} = 0.$$

For advection-dominant problems, the concentration at the latter part of the channel is zero until the concentration front arrives. Hence, for grid-based methods, such as SEM, one can save computational time by using a smaller domain at early times, and extend the domain as time increases. In this case, the homogeneous Dirichlet boundary condition can be used at the end of the domain, such that extending the domain can be accomplished by simply setting the concentration to be zero in the extended part. Similarly for the PSEM, particles with an initial zero concentration always exist beyond the front so that diffusion may occur downstream.

3. Numerical formulation and implementation. In this section, we describe the methods implemented to compute the transport of substances in microchannel geometries. We use both grid-based and grid-free formulations for solving the Navier–Stokes (2.1) and advection-diffusion-reaction (2.4) equations. All methods are programmed in FORTRAN 90 language and compiled with the PGI 7.2 compiler (The Portland Group, Inc., Lake Oswego, OR). The programs are executed on 2.4 GHz AMD Opteron processors with 2GB memory per CPU. The methods executed in parallel also use the MVAPICH2 MPI software (The Ohio State University).

3.1. Velocity calculations. We compute the velocity solution using the grid-based SEM for cases when $Re > 0$. When it is appropriate to assume that $Re \rightarrow 0$, a parallelized version of the grid-free BEM is used. We also use the SEM for $Re = 0$ for comparison.

3.1.1. Spectral element method. For this implementation, the Navier–Stokes equations (2.1) are discretized in time by the second-order rotational pressure correction scheme [45]. In the first substep, we solve for an intermediate velocity $\tilde{\mathbf{u}}^{k+1}$ from

$$(3.1) \quad \frac{3\tilde{\mathbf{u}}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\Delta t} + 2\mathbf{N}^k - \mathbf{N}^{k-1} - \frac{1}{\text{Re}} \nabla^2 \tilde{\mathbf{u}}^{k+1} + \frac{1}{\text{Re}} \nabla p^k = \mathbf{0},$$

with boundary conditions (2.5) and (2.7), where $\mathbf{N}^k = (\mathbf{u}^k \cdot \nabla)\mathbf{u}^k$ is the nonlinear term. In the second substep, the incompressibility of the velocity field is enforced by projecting $\tilde{\mathbf{u}}^{k+1}$ onto an incompressible space:

$$(3.2) \quad \begin{aligned} \tilde{\mathbf{u}}^{k+1} &= \mathbf{u}^{k+1} + \nabla \phi^{k+1}, \\ \nabla \cdot \mathbf{u}^{k+1} &= 0, \\ \mathbf{u}^{k+1} \cdot \mathbf{n} &= \tilde{\mathbf{u}}^{k+1} \cdot \mathbf{n} \quad \text{on } \partial\Omega. \end{aligned}$$

Finally, the pressure is updated through

$$(3.3) \quad p^{k+1} = p^k + \frac{3\text{Re}}{2\Delta t} \phi^{k+1} - \nabla \cdot \tilde{\mathbf{u}}^{k+1}.$$

Note that this approach can be adopted to solve the Stokes equations. This is accomplished by solving the steady-state solution of the time-dependent Stokes equations (2.2). The pressure correction scheme results in a number of Poisson-type equations to be solved in each time step. These equations are solved using an SEM. Let $T = \{\Omega_i\}$ be a triangulation of the domain Ω . Then the spectral element space is

$$(3.4) \quad V_N(T) := P_N(T) \cap V,$$

where

$$(3.5) \quad P_N(T) := \{u \in C^0(\Omega) : u|_{\Omega_i} \in P_N(\Omega_i) \quad \forall \Omega_i \in T\}$$

is the space of continuous functions whose restriction in each subdomain Ω_i is a polynomial of degree at most N , and V is the solution space of the differential equation. The resulting linear system can be solved by a direct method or a preconditioned iterative method. For equations resulting from the discretization of a time derivative, the Jacobi preconditioner works very well. Otherwise, one may use a more sophisticated preconditioner, such as the overlapping Schwarz preconditioner [39]. After testing, we found that for the purposes of this paper, conjugate gradient with Jacobi preconditioner works best when solving the velocity, and a direct solver with substructuring technique works best when solving the pressure.

3.1.2. Boundary element method. When $\text{Re} \rightarrow 0$, (2.1) reduces to the Stokes equations (2.3). We use the following formulation of the BEM described by Halpern and Gaver [16]. Taking Fourier transforms of (2.3) and applying Green’s theorem yields the integral equation

$$(3.6) \quad u_k(\mathbf{x}) = \int_{\Gamma} f_{ik}(\mathbf{x}, \mathbf{y}) u_i(\mathbf{y}) d\Gamma_y - \int_{\Gamma} g_{ik}(\mathbf{x}, \mathbf{y}) \tau_i(\mathbf{y}) d\Gamma_y,$$

where the traction vector τ_i is defined as $\tau_i = \sigma_{ij} n_j$, the product of stress tensor σ_{ij} and the unit vector n_j normal to the boundary Γ directing away from the fluid, with

indices i and j equal to 1 (x -direction) or 2 (y -direction). The traction and velocity kernels, f_{ik} and g_{ik} , derived from the free-space Green's function are

$$(3.7) \quad \begin{aligned} f_{ik} &= -\frac{1}{\pi} \left(\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)n_j(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^4} \right), \\ g_{ik} &= -\frac{1}{4\pi} \left(\delta_{ik} \log |\mathbf{x} - \mathbf{y}| - \frac{(x_i - y_i)(x_k - y_k)}{|\mathbf{x} - \mathbf{y}|^2} \right). \end{aligned}$$

The integrals are evaluated over the positions \mathbf{y} that describe the boundary Γ ; the solution vector $u_k(\mathbf{x})$ describes the velocity at the points \mathbf{x} within the bounded domain. In the limit as \mathbf{x} approaches a point on the boundary surface, singularities arise in the kernels f_{ik} and g_{ik} . Thus, the integral solution (3.6) becomes

$$(3.8) \quad C_{ki}u_k(\mathbf{x}) = \int_{\Gamma} f_{ik}(\mathbf{x}, \mathbf{y})u_i(\mathbf{y}) d\Gamma_y - \int_{\Gamma} g_{ik}(\mathbf{x}, \mathbf{y})\tau_i(\mathbf{y}) d\Gamma_y,$$

where $\mathbf{x} \in \Gamma$. The tensor C_{ki} accounts for stress discontinuities at the surface. If the boundary is smooth at \mathbf{x} , $C_{ki} = 1/2 \delta_{ki}$, but C_{ki} has a more complicated structure if the domain has corners [5].

Equation (3.8) is solved numerically by discretizing only the boundaries into N_{elm} three-node quadratic elements and evaluating the discrete version of the equation at each node, where the velocity u and stress τ vectors are approximated along each element m using quadratic polynomials and expressed in terms of a local arc-length coordinate. Numeric evaluation of the integrand in (3.8) for each element using standard Gaussian quadrature techniques yields the linear system of equations, $\underline{\mathbf{F}}\mathbf{w} = \underline{\mathbf{G}}\mathbf{v}$, where the vectors \mathbf{w} and \mathbf{v} contain the velocity and stress components. $\underline{\mathbf{F}}$ and $\underline{\mathbf{G}}$ are $2N_b \times 2N_b$ and $2N_b \times 3N_b$ matrices, respectively, where N_b is the number of boundary nodes. Matrix $\underline{\mathbf{G}}$ is larger than $\underline{\mathbf{F}}$ to allow the end-points of each node to have two distinct stress values because of two possible orientations of the normal vector, specifically at corner points. Both the velocity and stress must be specified in the x_1 - and x_2 -directions to avoid stress discontinuities at corner points; only two of the four degrees of freedom, u_1 , u_2 , τ_1 , and τ_2 , are applied on other node points [16].

The elements of $\underline{\mathbf{F}}$ and $\underline{\mathbf{G}}$ are computed using a 10-point regular Gaussian quadrature if \mathbf{x} does not coincide with one of the points on Γ_m , while a 10-point logarithmic quadrature is used to evaluate the portions of the integrals in (3.8) that contain the logarithmic singularity from g_{ik} . The kernel f_{ik} also contains a singularity, which are the diagonal coefficients of $\underline{\mathbf{F}}$. These components are computed indirectly by imposing a uniform flow in both the x_1 - and x_2 -directions using rigid body considerations as described in Brebbia and Dominguez [5]. The known velocity and stress boundary conditions are then applied, and the system is rearranged so that $\underline{\mathbf{A}}\mathbf{z} = \mathbf{b}$. In this equation, $\underline{\mathbf{A}}$ is a $2N_b \times 2N_b$ matrix, \mathbf{z} is a $2N_b$ vector containing the unknown velocities and stresses, and \mathbf{b} contains the known stress or velocity information. The matrices $\underline{\mathbf{F}}$ and $\underline{\mathbf{G}}$ are dense and need a large memory space to store element data. We parallelize the code with MPI (message passing interface) and utilize the two-dimensional block-cyclic data layout scheme, as described in *ScaLAPACK Users' Guide* [4], to distribute the large matrices $\underline{\mathbf{F}}$ and $\underline{\mathbf{G}}$ across a CPU grid so that the memory requirements on each CPU are reduced. This allows the use of the parallel ScaLAPACK routine PDGESV from the Netlib Repository (<http://www.netlib.org>) to solve the linear system of equations for \mathbf{z} .

The velocity vector at any internal point is computed after obtaining the boundary solution by evaluating the discrete form of the integral equation (3.6):

$$(3.9) \quad u_k(\mathbf{x}) = \sum_{m=1}^{N_{\text{elm}}} \left[\int_{S_m} f_{ik}(\mathbf{x}, \mathbf{y}) u_i(\mathbf{y}) dS_m - \int_{S_m} g_{ik}(\mathbf{x}, \mathbf{y}) \tau_i(\mathbf{y}) dS_m \right].$$

The internal velocity solution u_k at a given position \mathbf{x} requires a numerical integration over all boundary elements, which can be computationally expensive for large domains with dense node spacing. We have parallelized this computation by dividing the boundary elements into the number of CPUs (N_c); each CPU performs the integration for one group of elements. The discrete formulation (3.9) is then the summation of the integrals over N_c CPUs.

3.2. Advection-diffusion-reaction calculations. We compute the concentration profile of substances in microchannel geometries using the SEM and the grid-free PSEM.

3.2.1. Spectral element method. In the first method to solve advection-diffusion-reaction, (2.4) is discretized in time by the second-order scheme

$$(3.10) \quad \frac{3c_i^{k+1} - 4c_i^k + c_i^{k-1}}{2\Delta t} + (\mathbf{u}(t^{k+1}) \cdot \nabla) c_i^{*,k+1} = \frac{1}{\text{Pe}_i} \nabla^2 c_i^{k+1} + R_i^{*,k+1},$$

with boundary conditions (2.6) and (2.8) for the walls and outlet and a Dirichlet condition at the inlet. The second-order approximations to the concentrations and reaction terms at the current time step are

$$(3.11) \quad c_i^{*,k+1} = 2c_i^k - c_i^{k-1}, \quad R_i^{*,k+1} = 2R_i^k - R_i^{k-1},$$

which are solved with the SEM.

For an advection-dominant system, i.e., $\text{Pe} \gg 1$, the resultant linear system can be solved efficiently using the Jacobi preconditioner. In this case, the work per time step is proportional to the number of unknowns. Hence,

$$(3.12) \quad \text{CPU time} \propto (h^{-1}N)^d \Delta t^{-1},$$

where h is the typical size of an element, N is the polynomial degree, and $d = 2$ is the space dimension. Moreover, the element size h is proportional to the slope of the concentration front, $\text{Pe}^{1/2}$, and the time step Δt is proportional to the element size h due to the CFL condition. We note that the CFL condition is relevant here because the advection part is treated explicitly. Hence, (3.12) becomes

$$(3.13) \quad \text{CPU time} \propto N^d \text{Pe}^{(d+1)/2}.$$

3.2.2. Particle strength exchange. In the second method to solve (2.4), the concentration field is discretized as a group of “particles,” a pointwise distribution of the concentration continuum. Each of these particles is associated with a concentration value for each of the reactants in the system and initialized in the domain with uniform spacing h . Diffusion of the reagents is simulated by the exchange of concentrations, or strengths, with neighboring particles as the particles are advected through the domain by the prescribed flow condition.

The method is mathematically formulated by approximating the Laplacian term $\nabla^2 c_i$ of the transport equation (2.4) with an integral operator Q ($Q \sim \nabla^2$),

$$(3.14) \quad Qc_i(\mathbf{x}) = \frac{1}{\delta^2} \int [c_i(\mathbf{y}) - c_i(\mathbf{x})] \Lambda_\delta(\mathbf{x} - \mathbf{y}) d\mathbf{y},$$

which contains the diffusion kernel $\Lambda_\delta(\mathbf{x}) = \delta^{-2} \Lambda(|\mathbf{x}|/\delta)$. This two-dimensional kernel satisfies the moment condition

$$(3.15) \quad \delta^{-2} \iint_{R^2} x_i^2 \Lambda_\delta(\mathbf{x}) d\mathbf{x} = 2$$

to ensure that the integral operator approximates the Laplacian. There are many choices for the diffusion kernel [12]; for the simulations described in section 4 we use the fourth-order kernel derived by Cortez [9]:

$$(3.16) \quad \Lambda_\delta(\mathbf{x}) = \frac{4}{\pi\delta^2} (3 - r^2) \exp(-r^2), \quad r = \frac{|\mathbf{x}|}{\delta}.$$

The parameter δ defines the size of the Gaussian-shaped function (3.16) and is chosen relative to the initial particle spacing parameter h . The value of δ must be sufficiently large so that the solution spreads between the particles; however, δ much greater than the interparticle distance will artificially smooth the concentration. For the simulations in this study we simply use $\delta = h$. For precise conditions that ensure convergence of the method, see [9].

Using the approximation (3.14), equation (2.4) is decoupled into a set of discretized ODEs that describe the convection and diffusion-reaction of each particle p :

$$(3.17) \quad \frac{d}{dt} \mathbf{x}_p(t) = \mathbf{u}(\mathbf{x}_p(t)),$$

$$(3.18) \quad \frac{d}{dt} c_i(\mathbf{x}_p, t) = \frac{1}{\text{Pe}_i} \frac{h^2}{\delta^2} \sum_{j=1}^{N_p} [c_i(\mathbf{x}_j, t) - c_i(\mathbf{x}_p, t)] \Lambda_\delta(\mathbf{x}_p(t) - \mathbf{x}_j(t)) + R_i.$$

The ODEs are solved numerically with the Livermore Solver for Ordinary Differential Equations (LSODE; available from Netlib), which utilizes Adams methods (predictor-corrector) for nonstiff cases and backward differentiation methods in the stiff case [38]. Although LSODE is a serial solver, we have parallelized a portion of the calculation using MPI. Because the computation of the right-hand side of (3.18) for each particle is independent and requires only information from the previous time step, the particles are divided among CPUs to perform this computation. The result from each CPU is then broadcast to the other CPUs, and the set of ODEs is solved for the current time step. The particle velocity is computed from either the steady-state BEM solution or the time-dependent Navier–Stokes solution solved with the SEM. In the case of a steady-state velocity field or if the velocity reaches steady state quickly, the SEM or BEM velocity is computed beforehand on a grid and interpolated at each particle location at subsequent time steps.

A Neumann boundary condition (2.6) is imposed on the domain walls to create an insulated boundary. This condition is enforced by reflecting particles located within a specified distance of the walls symmetrically about the boundary to the outside of

the domain. The concentration values of the new particles equal those of the reflected ones and are included in the sum of (3.18). The inclusion of the additional “image” particles increases the number of operations required, but their use decreases the error such that a larger spacing parameter h may be used, thus reducing the total number of particles in the simulation and computational time required overall.

For the simulations carried out with this method, particles are regularly spaced at distance h initially, with fluid flowing from left to right. Particles continually advect into the domain following (3.17) from upstream with the prescribed initial concentration held constant until entering the PSEM computation at $x_1 = 0$. Particles with zero concentration are added directly downstream of the concentration front throughout the calculation to allow diffusion downstream and to fill in the transverse space between the higher velocity particles and the walls caused by the parabolic velocity profile. Particles are no longer included in the computation after exiting the end of the domain. Redistribution of particles to the initial spacing h is necessary if the particles become highly disorganized such that the interparticle distance becomes much smaller or greater than h . For example, recirculation regions or stagnation points reduce the density of particles in portions of the domain while increasing the particle density in other areas. When particle redistribution is necessary, the particles are replaced with a new set of regularly spaced particles where the concentration values are interpolated from the concentration values at the previous particle positions. The concentration at the new location \mathbf{x}_p^0 is obtained from the equation

$$(3.19) \quad c_i(\mathbf{x}_p^0) = \sum_{j=1}^{N_p} c_i(\mathbf{x}_j) \phi_\delta(\mathbf{x}_p^0 - \mathbf{x}_j) h^2,$$

where the blob function $\phi_\delta(\mathbf{x}) = \delta^{-2} \phi(|\mathbf{x}|/\delta)$ satisfies the condition $\iint_{R^2} \phi_\delta(\mathbf{x}) d\mathbf{x}$. We use the following blob function derived from (3.16) for the interpolation:

$$(3.20) \quad \phi_\delta(\mathbf{x}) = \frac{1}{2\pi\delta^2} (3 - r^2) \exp(-r^2), \quad r = \frac{|\mathbf{x}|}{\delta}.$$

The δ used for the interpolation blob (3.20) is not required to be identical to δ in the PSEM equations (3.16) and (3.18), but in our computations we use the same δ .

This interpolation scheme introduces an approximate 1% loss of mass near the boundaries in our calculations because the particle concentration is not allowed to spread beyond the imposed boundary. To conserve mass, we can again utilize image particles and interpolate the concentration on these particles outside the domain. The concentration is then distributed to the boundary particles and the reciprocal particles within the domain. While this technique is relatively simple for straight boundaries, the implementation for an irregularly bounded domain such as Ω_B is much more complicated because regularly spaced particles may not exist directly on the boundary.

One must carefully choose the appropriate time for redistribution by considering the error caused by particle disorganization and the smoothing error introduced with each interpolation: redistributing the particles too frequently contributes to unnecessary interpolation error, while infrequent particle redistribution in the case of an irregular flow field can lead to large errors. For the computations in section 4.3, we base the criterion for redistribution on the distance to the nearest neighbor of each particle. When this distance is smaller than $h/2$ or greater than $2h$ for more than a set percentage of the total number of particles in the computation, redistribution

occurs. The percentage is decided from “a priori” knowledge of the flow map, and for the computations in this paper we use 7–17%. A similar criterion based on a measure of distortion is described in [17]. See [11, 33, 34] for further details on particle redistribution schemes.

4. Results and discussion of the advection-diffusion-reaction schemes.

We first demonstrate the accuracy and efficiency of the SEM and PSEM for the advection-diffusion equation by solving the Taylor–Aris dispersion problem. The computational solution from each method is compared to analytical solutions valid for small and large Pe. Upon establishing accuracy of the two methods, we solve this problem for a range of Pe to elucidate regimes at which either method is best suited in terms of computational efficiency. We then investigate the difference between the zero Re steady-state velocity field solution computed with the BEM Stokes solver and the Navier–Stokes solution computed with the SEM for varying Re in straight and serpentine geometries. We solve the advection-diffusion-reaction equation in these geometries for a range of Pe and Re and determine the amount of reaction product formed to compare the two channel configurations.

4.1. Validation of method: Taylor–Aris dispersion.

4.1.1. Problem definition. To verify the validity of the parallel PSEM with image particles and the SEM implementation, we solve two-dimensional Taylor–Aris dispersion [37, 1, 24, 44] wherein a solute of concentration c_0 with diffusivity D is introduced at $x_1 = 0$ into fully developed laminar flow between two infinite parallel plates separated by a distance $2a$. The channel centerline is located at $x_2 = 0$. Here, in order to test the transport solvers only, we prescribe the velocity $u_1(\mathbf{x}) = 1.5U(1 - (x_2/a)^2)$, $u_2(\mathbf{x}) = 0$, where U is the mean axial velocity. Symmetry about the centerline permits modeling of the half domain $0 \leq x_2 \leq a$. For this scenario the governing equation,

$$(4.1) \quad \frac{\partial c}{\partial t} + 1.5U \left(1 - \left(\frac{x_2}{a} \right)^2 \right) \frac{\partial c}{\partial x_i} = D \left(\frac{\partial^2 c}{\partial x_1^2} + \frac{\partial^2 c}{\partial x_2^2} \right),$$

is subject to the following initial and boundary conditions representing a step-function input between insulated walls:

$$(4.2) \quad \begin{aligned} c(x_1 > 0, x_2, t = 0) &= 0, \\ c(x_1 \leq 0, x_2, t \geq 0) &= c_0, \\ c(x_1 \rightarrow \infty, x_2, t) &= 0, \\ \frac{\partial c}{\partial x_2} \Big|_{x_2=0} &= \frac{\partial c}{\partial x_2} \Big|_{x_2=a} = 0. \end{aligned}$$

From Kolev and van der Linden [24], the analytical solution to (4.1) is

$$(4.3) \quad \begin{aligned} c_a(x_1, x_2, t) &= \frac{1}{2} \operatorname{erfc} \left(\frac{x_1 - Ut}{2\sqrt{tD_{\text{eff}}}} \right) \\ &\quad - \frac{Ua^2}{8D} \left(-\frac{7}{30} + \frac{x_2^2}{a^2} - \frac{x_2^4}{2a^4} \right) \frac{1}{\sqrt{\pi t D_{\text{eff}}}} \exp \left(\frac{-(x_1 - Ut)^2}{4tD_{\text{eff}}} \right), \end{aligned}$$

where D_{eff} is the effective diffusivity

$$(4.4) \quad D_{\text{eff}} = D + \frac{2U^2 a^2}{105D}.$$

This solution is valid only at large times $t \gg \theta_D$, where $\theta_D = a^2/D$ is a characteristic transverse diffusion time, and is thus used only for small $Pe = aU/D$. At larger Péclet numbers, a very long channel length is required to reach the long-time solution (4.3). Thus, for large Pe we compare the computed solution to the short-time average concentration [44]

$$(4.5) \quad c_{ma}(x_1, t) = \frac{1}{2} \int_{-1}^1 c(x_1, x_2, t) dx_2 \\ \approx \begin{cases} 1, & x_1 \leq 0, \\ \sqrt{1 - x_1/(1.5Ut)}, & 0 < x_1 \leq 1.5Ut, \\ 0, & x_1 > 1.5Ut. \end{cases}$$

4.1.2. Computational settings. With $c_0 = a = U = 1$, we compute the solution to the Taylor dispersion problem and compare the solution $c(x_1, x_2, t)$ of both methods to (4.3) for $Pe = [10, 50, 100]$ at time $T = 5\theta_D$ and x_1 position $L = UT$ using the absolute and relative L^2 -errors Δc and δc :

$$(4.6) \quad \Delta c_{L^2}|_{x_1=L, t=T} = \sqrt{\int (c_a(x_2) - c(x_2))^2 dx_2}, \\ \delta c_{L^2}|_{x_1=L, t=T} = \frac{\Delta c_{L^2}}{\sqrt{\int (c_a(x_2))^2 dx_2}}.$$

For $Pe = [500, 1000, 5000]$, we compare the PSEM and SEM mean concentration solutions $c_m(x_1, t)$ to (4.5) at time $t = 10$ and position $x_1 = 10$ using the absolute and relative errors Δc_m and δc_m :

$$(4.7) \quad \Delta c_m = |c_{ma} - c_m|_{x_1=10, t=10}, \\ \delta c_m = \left| \frac{c_{ma} - c_m}{c_{ma}} \right|_{x_1=10, t=10}.$$

The PSEM solution is obtained with particle spacing $h = 0.1$ for the low Pe set and $h = 0.05$ for the high Pe set, and $\delta = h$. We use a smaller h for the set of larger Pe to resolve the steep concentration gradient and avoid oscillations at early time. Particles within a distance of 0.2 of the walls are mirrored about the boundary to enforce the boundary conditions. Redistribution is not needed for this case because the particles do not become irregularly spaced in this parabolic flow. The time step size Δt is determined adaptively by the ODE solver and increases as Pe increases. To compute the SEM solution, a regular triangular grid is used with uniform triangle area $h^2/2$, and the polynomial degree is $N = 6$. The time step is controlled by the CFL condition, which is dependent on the element size h (see section 3.2.1 for how h and Δt change with respect to Pe). Note that h here refers to the size of the triangles. With polynomial degree N there are around $(N/h)^2$ grid points per unit area. We use $h = 1$, $\Delta t = 1e-2$ for $Pe = [10, 50, 100]$; $h = 0.2$, $\Delta t = 2e-3$ for $Pe = [500, 1000]$; and $h = 0.1$, $\Delta t = 1e-3$ for $Pe = 5000$.

4.1.3. Comparison to analytical solution. We show the error in the computed solutions and the computational time required on the same single processor in Table 4.1. The clock time for the PSEM computation on 8 processors is also included. The processor efficiency $E = T_S/(T_{Nc}N_c)$ of the computations is 94%, where T_S is the

TABLE 4.1

Comparison of the PSEM and SEM solutions to Taylor–Aris dispersion analytical solutions at large times for low Pe (A) and small times for high Pe (B). The absolute and relative errors (4.6), (4.7) and the computational clock time required T_{N_c} for $N_c = 1$ and $N_c = 8$ are shown.

(A) Long-time	Δc_{L^2}		δc_{L^2}		$T_{N_c=1}$		$T_{N_c=8}$
	SEM	PSEM	SEM	PSEM	SEM	PSEM	PSEM
Pe = 10 $t = x_1 = 50$	1.18e-2	4.94e-3	2.36%	0.99%	00:00:19	04:24:12	00:35:38
Pe = 50 $t = x_1 = 250$	5.80e-3	1.65e-3	1.16%	0.33%	00:06:19	78:17:07	10:24:37
Pe = 100 $t = x_1 = 500$	5.60e-3	1.66e-3	1.11%	0.33%	00:24:44	251:42:17	33:28:17

(B) Short-time	Δc_m		δc_m		$T_{N_c=1}$		$T_{N_c=8}$
	SEM	PSEM	SEM	PSEM	SEM	PSEM	PSEM
Pe = 500 $t = x_1 = 10$	5.87e-3	4.44e-3	1.02%	0.83%	00:01:20	00:07:16	00:00:58
Pe = 1000 $t = x_1 = 10$	2.95e-3	1.82e-3	0.51%	0.32%	00:01:19	00:04:46	00:00:38
Pe = 5000 $t = x_1 = 10$	6.03e-4	8.15e-5	0.10%	0.01%	00:14:16	00:03:08	00:00:25

clock time required for one processor and T_{N_c} is the clock time required for N_c processors. In section A of the table, the solutions are compared to the long-time Taylor dispersion solution (4.3). In section B, the solutions are compared to the short-time mean solution (4.5). The error between the computed solutions and analytical solutions decreases with increasing Péclet number for both methods. The error is less than 1% for the PSEM and less than 2% for the SEM in all cases except $Pe = 10$. The clock time required by the SEM to compute the solutions at $Pe = 10, 50,$ and 100 is negligible compared to the time required by the PSEM. We note that the length of the domain increases with time in both methods, and the PSEM requires the continual addition of particles such that the number of ODEs to be solved also increases with time. At high Pe , the PSEM computation time decreases with increasing Pe , while the SEM time increases by 90% for $Pe = 5000$. The PSEM clock time is 4.5 times faster than the SEM at $Pe = 5000$ on a single processor.

4.1.4. Comparison of computational time. We have shown that the SEM and PSEM accurately compute advection-diffusion. To fairly compare the timing of the two methods, we compute the solution to the Taylor–Aris dispersion problem to $T = 100$ for $Pe = [10, 50, 100, 500, 1000, 5000]$. The computation parameters are the same as in section 4.1.2 except for the PSEM particle spacing. We use $h = 0.1$ for all cases so that the number of particles is constant, although the larger spacing results in a small loss of accuracy at the highest Pe . We show in Figure 4.1 that the CPU clock time T_{N_c} decreases with Pe using the PSEM, while the time required increases with Pe using the SEM. We note that the sharp jumps in the SEM plot are due to the changes in the number of elements and time step size in the computation. A finer mesh and smaller time steps are required for the SEM due to stability issues and cannot be held constant. From this figure and the time results in Table 4.1, we see that in terms of computational time the SEM is a suitable solver choice for $Pe < 500$, while the PSEM is appropriate for high Pe , convection-dominated problems. For the advection-diffusion-reaction computations in section 4.3 we use the PSEM only for $Pe = 5000$ and both methods for $Pe = 50$ and $Pe = 500$ to compare the solutions.

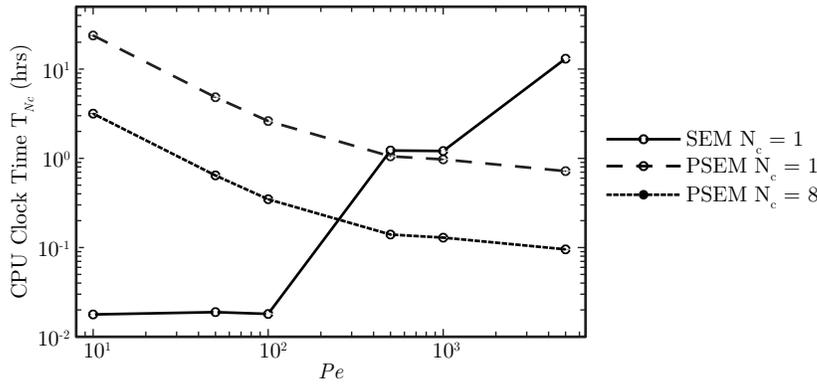


FIG. 4.1. The CPU clock time in hours required for the PSEM and SEM to compute the Taylor–Aris dispersion solution to $T = 100$ is plotted versus the Péclet number. We use the same type of processor in each case; N_c is the number of CPUs.

4.2. Effect of Re on microchannel flow field. To determine the variations in the flow field due to Reynolds number, we compute the SEM Navier–Stokes velocity solution for $Re = [0.01, 0.1, 1, 10, 50]$ and the SEM and BEM Stokes velocity solution ($Re = 0$) in the two geometries Ω_A and Ω_B shown in Figure 2.1. The variation in Re is assumed to be a change in the fluid property as the geometry and velocity remain constant. A parabolic velocity profile with maximum velocity $u_{\max} = 1$ is prescribed at each inlet. For the BEM, the boundary of domain Ω_A is discretized into 1076 3-node elements (node spacing = 0.025) and requires 46 seconds to compute the boundary Stokes solution using two processors. Each internal velocity point calculation (3.9) computed with 30 Gaussian points for this configuration requires $2.91e-3$ seconds on a single CPU; the calculation is nearly 100% efficient with increasing processors. The boundary of domain Ω_B is discretized into 1277 elements and requires 67 seconds to compute the boundary solution on two processors, and each internal velocity point calculation for this configuration requires $3.44e-3$ seconds on a single CPU. For the SEM computation, domains Ω_A and Ω_B are discretized into 9372 and 15121 triangles, respectively. Note that very small triangles are used along the curved parts of the boundary to make sure that the same domains as in the BEM are used. This results in the large number of triangles, a very small time step due to the CFL condition, and low polynomial degrees in the computations. Polynomial degree $N = 2$ is used for the velocity and $N = 1$ is used for the pressure. The SEM computation is performed using the same CPU as the BEM. For the $Re = 0$ case, the computation required 91 seconds for Ω_A and 213 seconds for Ω_B .

The physical time t required for the velocity to reach steady state and the total computation time T_{Nc} required are presented in Table 4.2. For $Re \leq 1$, the velocity solution reaches steady state within a single unit of time. The physical time and the computation time to reach steady state decrease with Re . We note that the $Re = 0.1$ solution reached steady state slightly faster than the $Re = 0.01$, resulting in fewer iterations and a shorter CPU time. The serpentine domain Ω_B requires both longer physical and computational time to reach steady state for $Re > 1$. The computational times of the BEM solution and the steady-state SEM solutions for $Re < 1$ are comparable; all are completed in about one minute.

In Figure 4.2 we plot the relative maximum difference Δu_{rel} between the velocity magnitude at $Re = 0$ and the steady-state velocity for $Re > 0$ in both domains,

TABLE 4.2

The physical time t and the CPU time T_{Nc} (hr:min:sec) for the SEM Navier–Stokes velocity solution to come to steady state in domains Ω_A and Ω_B for $Re = [0.01, 0.1, 1, 10, 50]$.

Domain		Re = 0.01	Re = 0.1	Re = 1	Re = 10	Re = 50
Ω_A	t	0.07	0.06	0.44	2.04	5.49
	T_{Nc}	00:00:32	00:00:20	00:01:37	00:05:22	00:14:04
Ω_B	t	0.07	0.07	0.44	2.37	11.71
	T_{Nc}	00:01:02	00:00:55	00:04:09	00:15:30	00:55:54

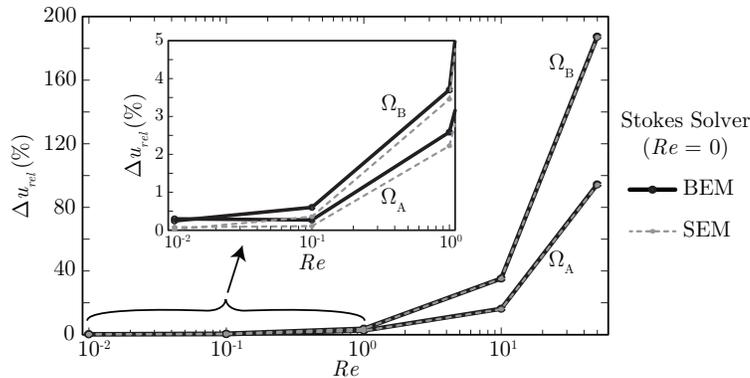


FIG. 4.2. Difference Δu_{rel} between the Navier–Stokes steady-state velocity solutions computed for $Re > 0$ with the SEM and the Stokes velocity solution ($Re = 0$) computed with the BEM and SEM in geometries Ω_A and Ω_B .

computed as

$$(4.8) \quad \Delta u_{\text{rel}} = \max_{\mathbf{x} \in \Omega} \left| \frac{|\mathbf{u}_{Re>0}(\mathbf{x})| - |\mathbf{u}_{Re=0}(\mathbf{x})|}{|\mathbf{u}_{Re=0}(\mathbf{x})|} \right|,$$

where $\mathbf{u}_{Re>0}$ and $\mathbf{u}_{Re=0}$ are the velocity profiles solved with the Navier–Stokes equation (2.1) and Stokes equation (2.3), respectively. We exclude the boundary points in the calculation of error where $\mathbf{u} = \mathbf{0}$, and internal velocity points are a minimum distance of 0.05 from the wall. The relative maximum difference Δu_{rel} between the SEM and BEM solutions for $Re = 0$ are $\Delta u_{\text{rel}} = 0.37\%$ for Ω_A and $\Delta u_{\text{rel}} = 0.25\%$ for Ω_B . As clearly shown in Figure 4.2, the difference between the two solutions decreases with Re and is less than 1% for $Re < 1$ in both the straight and serpentine geometries, indicating that the BEM solution may be used for such cases. In the straight domain Ω_A , the differences between the $Re = 0$ and $Re > 0$ velocity fields occur in the region where the three-inlets converge into the main channel. The largest difference occurs at $Re = 50$ in the serpentine geometry.

In Figure 4.3, we show the velocity vector field in domain Ω_B for $Re = 0$ and $Re = 50$. For the higher Reynolds number case, centrifugal forces cause the maximum velocity to move away from the channel center towards the walls. This shift may enhance the combination of two substances by forcing the lower streams into the vicinity of the upper streams. In three-dimensional channels, the wavy walls generate Dean vortices that promote cross-sectional mixing. This phenomenon has been applied in the development of microfluidic mixers [20, 19, 7, 43].

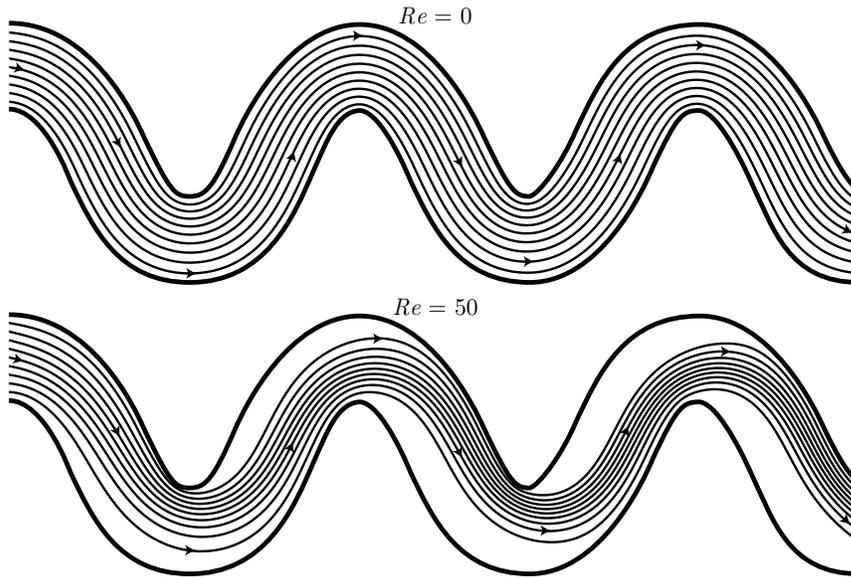


FIG. 4.3. The steady-state velocity streamlines for domain Ω_B with $Re = 0$ and $Re = 50$.

4.3. Advection-diffusion-reaction in microchannel domains.

4.3.1. Problem definition. We investigate the transport-reaction between two concentrations in the two domains Ω_A and Ω_B and compare the amount of product formation. The steady-state velocity profiles computed in section 4.2 are used to solve the advection-diffusion-reaction equation (2.4) for the concentration of three species: an antibody c_1 , an analyte c_2 , and antibody-analyte complex c_3 . A reversible reaction occurs between c_1 and c_2 that is described by the reaction terms

$$(4.9) \quad R_1 = R_2 = -r_f c_1 c_2 + r_d c_3, \quad R_3 = r_f c_1 c_2 - r_d c_3,$$

where $r_f = 5.2$ and $r_d = 0.1872$ are the dimensionless formation and dissociation parameters, respectively. These specific values describe the reaction between the analyte fluorescein and its antibody. The three species are assumed to have identical Péclet numbers for this calculation, and the initial concentration conditions are set as shown in Figure 2.1 with $C_0 = 1$. We compute the average c_3 relative to the total amount of c_2 in the outlet region A at time $t = 20$:

$$(4.10) \quad \bar{c}_3 = \frac{\int_A c_3 d\mathbf{x}}{\int_A (c_2 + c_3) d\mathbf{x}}, \quad A = \{\mathbf{x} \in \Omega : 15.2 \leq x_1 \leq 18\}.$$

While the path length the fluid travels in Ω_B is longer due to the sinuous geometry, the two domains are of equal geometric length in the x - y coordinate system.

4.3.2. Computational settings. We use the PSEM formulation for the case of $Pe = 5000$ and compare both methods at $Pe = 50$ and $Pe = 500$. We complete the simulation for $Re = [0, 0.01, 0.1, 1, 10, 50]$. We note that the changes in Re and Pe are based upon a change in a fluid property (density or viscosity) or concentration property (diffusivity) since the velocity and geometry length scale are held constant for all cases.

The steady-state velocity field for each Re case is obtained from the SEM or BEM and used in the transport calculation. For the PSEM computations, the velocities at the particle locations are interpolated from this field at every time step to advance the particles. We use $h = 0.05$ and a mirror particle distance of 0.2. For Ω_A , the particles are redistributed every two time units as determined by the interparticle distance criterion described in section 3.2.2. This time interval corresponds to the time when more than 7% of the particles are at a distance smaller than $h/2$ or greater than $2h$ to the nearest particle. For Ω_B , redistribution occurs at every one time unit, corresponding to the time when more than 17% of the particles are out of the specified distance. More frequent redistribution created an unacceptable smoothing error due to the interpolation in this geometry. The SEM is solved with polynomial degree $N = 2$ for $Pe = 50$ and $N = 4$ for $Pe = 500$.

4.3.3. Simulation results. The average time steps used for the PSEM and the time steps determined by the CFL condition for the SEM are shown in Table 4.3. The average time step is reported for the PSEM because it is determined adaptively by the ODE solver. The results for $Re < 0.1$ are not shown as they are similar to the results for $Re = 0.1$.

TABLE 4.3

Time step Δt used for PSEM and SEM computations in Ω_A and Ω_B . The time step is constant for the SEM in each case; the average is given for the PSEM.

Δt		Pe = 50		Pe = 500		Pe = 5000
		SEM	PSEM	SEM	PSEM	PSEM
Ω_A	All Re	5e-3	2e-3	1e-3	1e-2	1e-2
	Re = 50	2e-3	3e-3	4e-4	4e-3	4e-3
Ω_A	Re = 10	5e-3	3e-3	5e-4	8e-3	1e-2
	Other Re	5e-3	3e-3	1e-3	8e-3	1e-2

The computational times for both methods are shown in Table 4.4. We see the same trend as discussed in section 4.1.4: the clock time required for N_c processors (T_{N_c}) increases with increasing Pe for the SEM, while the opposite is true for the PSEM. At Pe = 50 the SEM computation is completed in 1–2% of the time required by the PSEM. The SEM computes the solutions about two times faster than the PSEM on a single processor for Pe = 500. Increasing the number of processors reduces the computational time of the PSEM; the processor efficiency $E = T_{N_c=1}/(T_{N_c}N_c)$ of the computations is 94%. The PSEM computational time decreases for Pe = 5000. We did not compute this case with the SEM because of the fine spatial and time resolution, and thus long CPU time, required. For the PSEM times given, 25% of the computation time for Ω_A and 40% of the computation time for Ω_B is spent interpolating the particle velocities, while the remainder is spent computing particle concentrations. Once the Navier–Stokes solution reaches steady state, the SEM does not require this additional step because the steady-state velocity is always known at the grid points.

The averaged product concentration \bar{c}_3 (4.10) is shown in Table 4.5. First, comparing the PSEM and SEM methods in both geometries at Pe = 50 and Pe = 500, we find that the overall relative difference in \bar{c}_3 is 0.5% at Pe = 50 in Ω_A , 2.7–2.8% at Pe = 50 in Ω_B , 0.2%–1.0% at Pe = 500 in Ω_A , and 0.6–1.2% at Pe = 500 in Ω_B , with the SEM providing slightly larger values. There is little variation in \bar{c}_3 as a function of Re at Pe = 50 and Pe = 500 in both geometries indicating that diffusion

TABLE 4.4

We show the computational clock time T_{N_c} for each case, where N_c is the number of processors used. The parallel PSEM scales at 94% efficiency with $N_c = 8$.

Ω_A								
	Pe = 50			Pe = 500			Pe = 5000	
	SEM	PSEM		SEM	PSEM		PSEM	
Re	$T_{N_c=1}$	$T_{N_c=1}$	$T_{N_c=8}$	$T_{N_c=1}$	$T_{N_c=1}$	$T_{N_c=8}$	$T_{N_c=1}$	$T_{N_c=8}$
50	00:15:52	29:27:12	03:55:00	03:45:38	07:23:03	00:58:55	05:16:13	00:42:03
10	00:16:45	31:55:06	04:14:40	03:34:13	06:54:34	00:55:08	04:55:10	00:39:15
1	00:15:48	32:29:11	04:19:12	03:34:30	06:43:00	00:55:35	04:58:33	00:39:42
0.1	00:14:33	32:35:12	04:20:00	03:48:07	06:58:55	00:55:42	04:54:09	00:39:07
Ω_B								
	Pe = 50			Pe = 500			Pe = 5000	
	SEM	PSEM		SEM	PSEM		PSEM	
Re	$T_{N_c=1}$	$T_{N_c=1}$	$T_{N_c=8}$	$T_{N_c=1}$	$T_{N_c=1}$	$T_{N_c=8}$	$T_{N_c=1}$	$T_{N_c=8}$
50	01:34:50	44:06:17	05:51:54	12:44:42	20:38:48	02:44:44	19:32:52	02:35:58
10	00:48:44	49:59:59	06:38:56	10:33:10	12:19:43	01:38:22	10:02:29	01:20:07
1	00:49:03	53:07:59	07:03:56	05:59:05	11:43:07	01:33:30	09:38:48	01:16:58
0.1	00:50:18	53:03:43	07:03:22	05:54:26	11:50:31	01:34:29	09:43:48	01:17:38

TABLE 4.5

The average \bar{c}_3 relative to the total amount of c_2 in the outlet region A at time $t = 20$ (4.10) is computed for domains Ω_A and Ω_B at a range of Reynolds and Péclet numbers. Product formation decreases with increasing Pe, with little variation over Re.

Domain	Re	Pe = 50		Pe = 500		Pe = 5000
		SEM	PSEM	SEM	PSEM	PSEM
Ω_A	50	0.934	0.930	0.853	0.862	0.465
	10	0.934	0.930	0.858	0.860	0.475
	1	0.934	0.930	0.860	0.861	0.478
	0.1	0.934	0.930	0.860	0.862	0.479
Ω_B	50	0.928	0.902	0.904	0.894	0.754
	10	0.933	0.908	0.898	0.892	0.715
	1	0.933	0.908	0.898	0.893	0.714
	0.1	0.933	0.908	0.897	0.893	0.714

dominates over variation in the convection field as would be expected when $Pe \gg 1$. The differences are larger at $Pe = 5000$. We note a slight increase in product with *decreasing* Re in Ω_A at $Pe = 500, 5000$ and Ω_B at $Pe = 50$. Conversely, the product formation increases with *increasing* Re in Ω_B at $Pe = 500, 5000$. Decreasing Re enhances diffusive mixing of the two reacting species, while geometries such as the sinuous channel increase inertial effects that may also increase mixing at higher Re.

As Pe is increased, the reaction product \bar{c}_3 decreases in both geometries. However, the serpentine geometry Ω_B produces more \bar{c}_3 than the straight domain Ω_A , particularly when $Pe = 5000$. At this Pe, the averaged product concentration \bar{c}_3 is 62% greater in Ω_B than in Ω_A at $Re = 50$; the product formation is 50% greater in Ω_B than in Ω_A at lower Re. This indicates that at the high Péclet numbers augmenting the geometry of the microchannel domain to encourage varying flow patterns can enhance product formation.

4.4. Further comments. In the previous sections, we demonstrate the use of the BEM, PSEM, and SEM to simulate advection-diffusion-reaction processes in models of microchannels. We demonstrate the validity of the methods by comparing simulation results to the analytical solutions of the Taylor–Aris dispersion problem. The error between the analytical and computed solutions is small and decreases with increasing Pe . We show that at small Pe , the SEM computes the solution faster due to the increased diffusion, while the PSEM requires smaller (and hence more) time steps to solve the ODEs in this regime. Conversely, the PSEM is faster at large Pe because the method does not require the CFL condition. The BEM Stokes solution for the velocity field is appropriate for $Re < 1$, which is generally true in microfluidic devices. This method is suited to complex domains because an internal grid is not required to find the solution. The internal velocity at any point may be computed as a postprocessing step. In steady flow, we compute the internal velocity only once at the SEM mesh points or at a high density of internal points so that the velocity at individual particle positions can be interpolated for the PSEM. On the other hand, if it takes considerable time for the velocity to reach steady state because Re is large or the domain geometry is complicated, the SEM can compute the velocity field on the fly using the same triangulation as for the concentration field. This is advantageous because in this case there would be no need to store the velocity at each time step. Analysis of reaction product formation in straight and serpentine geometries at a range of Pe and Re shows that improvements in the geometry to promote mixing of two species encourage the reaction at the high Pe and low Re expected in microchannel flows.

While we have shown that the implementations of the methods described in this paper accurately compute the advection-diffusion-reaction equation in microchannel geometries, further improvements may be made. The SEM computations can be improved in several ways. First, in the computations in sections 4.2 and 4.3, very small triangular elements are used along the curved portions of the boundary. This ensures that the boundary is the same as the piecewise-linear one assumed for the BEM. This results in an increased number of triangular elements and very small time steps, which increase the computational time. In practice, one could use bigger triangular elements by allowing curved triangles along the boundary. Second, the SEM computations are not parallelized. We found that most of the CPU time was spent calling the matrix-vector multiplication function DGEMV in the BLAS library (Netlib Repository, <http://www.netlib.org>). We attempted a simple parallelization by using the multithread version of the BLAS library (Intel Math Kernel Library), but saw no improvement with eight threads compared to a single processor. To achieve an efficient parallelization of the SEM code, we could develop a parallel algorithm of the SEM by recomposing the existing code. Compared with the SEM, the PSEM algorithm is simple, and therefore it is easier to distribute the workload over multiple CPUs.

Finally, for convection-dominated applications, a large part of the domain has zero concentration at early time. One could use a smaller domain with zero boundary condition at the outlet at early time. In the PSEM computations, this strategy is already employed by putting particles only near the inlets initially. In the SEM computations, the strategy is employed in section 4.1 but not in section 4.3. In a general domain, this strategy would require more involved postprocessing procedures for the generated mesh.

We note that the BEM-PSEM implementation is formulated to compute advection-diffusion-reaction in complex domains with internal obstacles at high Pe , which

we did not show in this paper. The implementation may be extended to three dimensions and may be modified to include physico-chemical and/or fluid-structure interactions. We mentioned the drawbacks of the PSEM in section 1.2: particle distortion due to the flow map and enforcement of boundary conditions. The redistribution and interpolation scheme described in this paper caused a slight loss of mass at the boundaries. We could ameliorate this by using a different interpolation scheme or by using one-sided blob that covers only the space within the boundary in (3.19). The use of image particles as described in section 3.2.2 to deal with enforcing the no-flux condition on the channel walls is time-consuming and presents difficulties with internal obstacles. This method could be refined for efficiency, and we could incorporate one-sided blobs for the particles on or near the boundaries. We note, however, that to achieve an accuracy level similar to the results in section 4.1.3 without image particles the particle spacing h must be reduced to $h = 0.025$, requiring a large increase in the number of particles. Additionally, the code could be further optimized for faster processing time, such as by implementing a parallel ODE solver.

5. Conclusion. We have presented a combination of grid-based and grid-free methods that are useful for computing advection-diffusion-reaction in microchannel geometries for varying fluid and transport conditions. The grid-free methods are particularly advantageous for investigating transport in the convection-dominated, irregular geometry problems expected for microfluidic applications where the Reynolds number is near zero and the Péclet number is very high. Both the PSEM and the BEM may be adapted to problems with moving boundaries, which could be employed to model droplets or bubbles in microchannels. Together, these computational tools may be applied to the design and optimization of microscale channels for use in antibody-based sensors and other microfluidic applications, which would significantly accelerate the development of such devices.

Acknowledgments. The authors acknowledge Tulane's Center for Computational Science and the Louisiana Optical Network Initiative (LONI) for providing computational resources. The codes for the methods described in this paper are available upon request.

REFERENCES

- [1] R. ARIS, *On the dispersion of a solute in a fluid flowing through a tube*, Proc. R. Soc. Lond. Ser. A, 235 (1956), pp. 67–77.
- [2] M. BERGDORF, G.-H. COTTET, AND P. KOUMOUTSAKOS, *Multilevel adaptive particle methods for convection-diffusion equations*, Multiscale Model. Simul., 4 (2005), pp. 328–357.
- [3] K. BLACK, *Spectral element approximation of convection-diffusion type problems*, Appl. Numer. Math., 33 (2000), pp. 373–379.
- [4] L. S. BLACKFORD, J. CHOI, A. CLEARY, E. D'AZEVEDO, J. DEMMEL, I. DHILLON, J. DONGARRA, S. HAMMARLING, G. HENRY, A. PETITET, K. STANLEY, D. WALKER, AND R. C. WHALEY, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, PA, 1997.
- [5] C. A. BREBBIA AND J. DOMINGUEZ, *Boundary Elements: An Introductory Course*, Computational Mechanics, Southampton, England, 1989.
- [6] P. CHATELAIN, A. CURIONI, M. BERGDORF, D. ROSSINELLI, W. ANDREONI, AND P. KOUMOUTSAKOS, *Billion vortex particle direct numerical simulations of aircraft wakes*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 1296–1304.
- [7] J. J. CHEN, C. H. CHEN, AND S. R. SHIE, *Optimal designs of staggered Dean vortex micromixers*, Int. J. Mol. Sci., 12 (2011), pp. 3500–3524.
- [8] A. J. CHORIN, *Numerical study of slightly viscous flow*, J. Fluid Mech., 57 (1973), pp. 785–796.
- [9] R. CORTEZ, *Convergence of high-order deterministic particle methods for the convection-diffusion equation*, Comm. Pure Appl. Math., 50 (1997), pp. 1235–1260.

- [10] G.-H. COTTET AND S. MAS-GALLIC, *A particle method to solve the Navier–Stokes system*, Numer. Math., 57 (1990), pp. 805–827.
- [11] G.-H. COTTET, M.-L. OULD SALIHI, AND M. EL HAMRAOUI, *Multi-purpose regriding in vortex methods*, in ESAIM Proceedings Third International Workshop on Vortex Flows and Related Numerical Methods, ESAIM Proc. 7, 1999, pp. 94–103.
- [12] P. DEGOND AND S. MAS-GALLIC, *The weighted particle method for convection-diffusion equations. Part 1: The case of an isotropic viscosity*, Math. Comp., 53 (1989), pp. 485–507.
- [13] M. O. DEVILLE, P. F. FISCHER, AND E. H. MUND, *High-Order Methods for Incompressible Fluid Flows*, Cambridge University Press, Cambridge, England, 2002.
- [14] D. P. GAVER III, D. HALPERN, O. E. JENSEN, AND J. B. GROTBORG, *The steady motion of a semi-infinite bubble through a flexible-walled channel*, J. Fluid Mech., 319 (1996), pp. 25–65.
- [15] D. P. GAVER III AND S. M. KUTE, *A theoretical model study of the influence of fluid stresses on a cell adhering to a microchannel wall*, Biophys. J., 75 (1998), pp. 721–733.
- [16] D. HALPERN AND D. P. GAVER III, *Boundary element analysis of the time-dependent motion of a semi-infinite bubble in a channel*, J. Comput. Phys., 115 (1994), pp. 366–375.
- [17] S. E. HIEBER AND P. KOUMOUTSAKOS, *A Lagrangian particle level set method*, J. Comput. Phys., 210 (2005), pp. 342–367.
- [18] J. J. L. HIGDON, *Stokes flow in arbitrary two-dimensional domains: Shear flow over ridges and cavities*, J. Fluid Mech., 159 (1985), pp. 195–226.
- [19] P. B. HOWELL, JR., D. R. MOTT, J. P. GOLDEN, AND F. S. LIGLER, *Design and evaluation of a Dean vortex-based micromixer*, Lab Chip, 4 (2004), pp. 663–669.
- [20] F. JIANG, K. S. DRESE, S. HARDT, M. KÜPPER, AND F. SCHÖNFELD, *Helical flows and chaotic mixing in curved micro channels*, AIChE J., 50 (2004), pp. 2297–2305.
- [21] V. JOHN AND E. SCHMEYER, *Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion*, Comput. Methods Appl. Mech. Engrg., 198 (2008), pp. 475–494.
- [22] M. A. KELMANSON, *An integral equation method for the solution of singular slow flow problems*, J. Comput. Phys., 51 (1983), pp. 139–158.
- [23] A. KLAWONN, L. F. PAVARINO, AND O. RHEINBACH, *Spectral element FETI-dp and BDDC preconditioners with multi-element subdomains*, Comput. Methods Appl. Mech. Engrg., 198 (2008), pp. 511–523.
- [24] S. D. KOLEV AND W. E. VAN DER LINDEN, *Laminar dispersion in parallel plate sections of flow systems used in analytical chemistry and chemical engineering*, Anal. Chim. Acta, 247 (1991), pp. 51–60.
- [25] P. KOUMOUTSAKOS, *Inviscid axisymmetrization of an elliptical vortex*, J. Comput. Phys., 138 (1997), pp. 821–857.
- [26] P. KOUMOUTSAKOS, *Multiscale flow simulations using particles*, Annu. Rev. Fluid Mech., 37 (2005), pp. 457–487.
- [27] Y.-Y. KWAN, *Additive Schwarz preconditioners with minimal overlap for triangular spectral elements*, Commun. Comput. Phys., 13 (2013), pp. 411–427.
- [28] O. A. LADYZHENSKAYA, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York, 1963.
- [29] S. LI AND W. K. LIU, *Meshfree and particle methods and their applications*, Appl. Mech. Rev., 55 (2002), pp. 1–34.
- [30] C.-C. LIN, J.-H. WANG, H.-W. WU, AND G.-B. LEE, *Microfluidic immunoassays*, J. Assoc. Lab. Autom., 15 (2010), pp. 253–274.
- [31] P. A. LOTT AND H. ELMAN, *Fast iterative solver for convection-diffusion systems with spectral elements*, Numer. Methods Partial Differential Equations, 27 (2011), pp. 231–254.
- [32] J. W. LOTTES AND P. F. FISCHER, *Hybrid multigrid/Schwarz algorithms for the spectral element method*, J. Sci. Comput., 24 (2005), pp. 613–646.
- [33] J. J. MONAGHAN, *Extrapolating b splines for interpolation*, J. Comput. Phys., 60 (1985), pp. 253–262.
- [34] H. O. NORDMARK, *Deterministic high order vortex methods for the 2d Navier-Stokes equation with rezoning*, J. Comput. Phys., 129 (1996), pp. 41–56.
- [35] A. T. PATERA, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phys., 54 (1984), pp. 468–488.
- [36] L. F. PAVARINO, *Overlapping Schwarz preconditioners for spectral element discretizations of convection-diffusion problems*, Internat. J. Numer. Methods Engrg., 53 (2002), pp. 1005–1023.
- [37] R. F. PROBSTEIN, *Taylor dispersion in a capillary tube*, in Physicochemical Hydrodynamics: An Introduction, 2nd ed., Wiley, New York, 1994, pp. 82–96.

- [38] K. RADHAKRISHNAN AND A. C. HINDMARSH, *Description and Use of lsode, the Livermore Solver for Ordinary Differential Equations*, LLNL Report UCRL-ID-113855, Lawrence Livermore National Laboratory, Livermore, CA, 1993.
- [39] J. SCHÖBERL, J. M. MELENK, C. PECHSTEIN, AND S. ZAGLMAYR, *Additive Schwarz preconditioning for p -version triangular and tetrahedral finite elements*, IMA J. Numer. Anal., 28 (2008), pp. 1–24.
- [40] O. SHIPILOVA, H. HAARIO, AND A. SMOLIANSKI, *Particle transport method for convection problems with reaction and diffusion*, Internat. J. Numer. Methods Fluids, 54 (2007), pp. 1215–1238.
- [41] B. J. SMITH AND D. P. GAVER III, *The pulsatile propagation of a finger of air within a fluid-occluded cylindrical tube*, J. Fluid Mech., 601 (2008), pp. 1–23.
- [42] H. A. STONE, A. D. STROOCK, AND A. AJDARI, *Engineering flows in small devices: Microfluidics toward lab-on-a-chip*, Annu. Rev. Fluid Mech., 36 (2004), pp. 381–411.
- [43] A. P. SUDARSAN AND V. M. UGAZ, *Multivortex micromixing*, Proc. Natl. Acad. Sci. USA, 103 (2006), pp. 7228–7233.
- [44] G. TAYLOR, *Dispersion of soluble matter in solvent flowing slowly through a tube*, Proc. R. Soc. Lond. Ser. A, 219 (1953), pp. 186–203.
- [45] L. J. P. TIMMERMANS, P. D. MINEV, AND F. N. VAN DE VOSSE, *An approximate projection scheme for incompressible flow using spectral elements*, Internat. J. Numer. Methods Fluids, 22 (1996), pp. 673–688.
- [46] C. W. WININGER AND J. J. HEYS, *Particle transport modeling in pulmonary airways with high-order elements*, Math. Biosci., 232 (2011), pp. 11–19.
- [47] L.-A. YING, *Vortex method for two dimensional Euler equations in bounded domains with boundary correction*, Math. Comp., 67 (1998), pp. 1383–1400.