

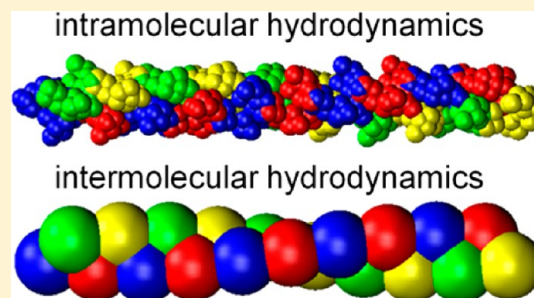
# Molecule-Centered Method for Accelerating the Calculation of Hydrodynamic Interactions in Brownian Dynamics Simulations Containing Many Flexible Biomolecules

Adrian H. Elcock\*

Department of Biochemistry, University of Iowa, Iowa City, Iowa 52242, United States

**S** Supporting Information

**ABSTRACT:** Inclusion of hydrodynamic interactions (HIs) is essential in simulations of biological macromolecules that treat the solvent implicitly if the macromolecules are to exhibit correct translational and rotational diffusion. The present work describes the development and testing of a simple approach aimed at allowing more rapid computation of HIs in coarse-grained Brownian dynamics simulations of systems that contain large numbers of flexible macromolecules. The method combines a complete treatment of intramolecular HIs with an approximate treatment of the intermolecular HIs, which assumes that the molecules are effectively spherical; all of the HIs are calculated at the Rotne–Prager–Yamakawa level of theory. When combined with Fixman’s Chebyshev polynomial method for calculating correlated random displacements, the proposed method provides an approach that is simple to program but sufficiently fast that it makes it computationally viable to include HIs in large-scale simulations. Test calculations performed on very coarse-grained models of the pyruvate dehydrogenase (PDH) E2 complex and on oligomers of ParM (ranging in size from 1 to 20 monomers) indicate that the method reproduces the translational diffusion behavior seen in more complete HI simulations surprisingly well; the method performs less well at capturing rotational diffusion, but its discrepancies diminish with increasing size of the simulated assembly. Simulations of residue-level models of two tetrameric protein models demonstrate that the method also works well when more structurally detailed models are used in the simulations. Finally, test simulations of systems containing up to 1024 coarse-grained PDH molecules indicate that the proposed method rapidly becomes more efficient than the conventional BD approach in which correlated random displacements are obtained via a Cholesky decomposition of the complete diffusion tensor.



## ■ INTRODUCTION

It is increasingly common for molecular simulation studies of biological macromolecules to employ coarse-grained representations in which the macromolecular structures and their solvent environment are replaced by structurally simplified models.<sup>1,2</sup> It is well-known that coarse-graining the solvent, or omitting it entirely, has consequences for how the energetic interactions are modeled, and steps must therefore be taken to ensure that water’s screening of electrostatic interactions and its central role in controlling the favorable interactions of hydrophobic groups are both correctly accounted for. It is also well-known that omitting the solvent completely means that the energy-redistributing effects that result from collisions with water molecules and the viscous drag effects that water causes are both neglected. Techniques that attempt to correct for these deficiencies by modeling them implicitly include Langevin dynamics (LD) and Brownian dynamics (BD);<sup>3,4</sup> both methods share the common feature that random forces or displacements are exerted on the solutes to mimic collisions with the missing water molecules.

There is, however, another complication that arises from omitting the solvent in molecular simulations: this is that the solvent-mediated hydrodynamic interactions (HIs) that occur

between solutes, and that cause their diffusive motions to be correlated even in the absence of an energetic interaction between them, are completely missing.<sup>5</sup> While this might sound a priori like a major drawback, it is important to note that it is not, in fact, a problem when one is interested only in the thermodynamics of molecular behavior: the inclusion of HIs does not affect the shape of a system’s free energy landscape, it only affects the speed at which the free energy landscape is explored.<sup>4,5</sup> However, it can be a major problem when attempting to accurately model kinetic or dynamic properties; in particular, the translational and rotational diffusive motion of macromolecules can be wildly inaccurate in simulations that neglect HIs.<sup>6</sup> This issue has long been known to be a problem in simulating the diffusive properties of highly flexible (unfolded) polymers,<sup>7</sup> but its implications for the modeling of folded biological macromolecules has been much less fully explored. In recent work,<sup>6</sup> we showed that in BD simulations of protein molecules modeled in their native states, the translational and rotational diffusion coefficients were all drastically underestimated when HIs were neglected, with errors of up to

**Received:** March 25, 2013

an order of magnitude for a protein of 145 residues. In the same work, we also showed that the  $\sim 1.5$ -fold difference between the translational diffusion coefficients of the folded and unfolded states of single-domain proteins could not be reproduced at all when HIs were omitted from BD simulations but was successfully captured when HIs were incorporated into the simulations at the Rotne–Prager–Yamakawa<sup>8,9</sup> (RPY) level of theory. In an interesting new direction for this kind of work, the group of García de la Torre has very recently shown that similar methods can be used to successfully model the hydrodynamic behavior of intrinsically disordered proteins.<sup>10</sup>

Because HIs affect the diffusive motions of macromolecules, they can also, in principle, affect the rates at which they fold and associate. A number of groups have explored the effects of including HIs in simulations of protein folding and unfolding rates.<sup>6,11–15</sup> The results of these studies appear more equivocal, perhaps in part because of differences in the way that the HIs, and the proteins themselves, have been modeled. Nevertheless, using simulation models similar to those employed here, the Cieplak group has shown that the folding rates of proteins appear to be accelerated by  $\sim 2$ -fold when HIs are included,<sup>13</sup> and we have obtained very similar findings;<sup>6</sup> the Cieplak group has also explored the effects of HIs in protein unfolding processes in externally applied solvent flows.<sup>14,15</sup> In the case of folding, most of the acceleration appears to be attributable to an increased rate of initial collapse of the polypeptide chain; again, the effects of HIs on the kinetics of polymer collapse have also been extensively studied in the field of polymer physics.<sup>12,16–18</sup>

In terms of effects of HIs on protein–protein association rates, theoretical studies from many years ago had suggested that the association rates of model proteins (i.e., spheres) should be decreased by  $\sim 10$ – $50\%$  depending on the model used.<sup>19–22</sup> These ideas were explored in BD simulation studies by the McCammon group, explicitly modeling the association kinetics of dumbbell models of enzymes and substrates,<sup>23</sup> and in our own recent work, examining the kinetics of the electrostatically accelerated barnase–barstar association process.<sup>24</sup> The Długosz and Trylska groups have recently examined the coupling of electrostatic and hydrodynamic interactions in determining association rates in a variety of model systems and have shown that, for associations of anisotropic molecules, the decrease in association rates due to HIs depends significantly on the strength of electrostatic interactions.<sup>25</sup> A significant role for HIs has also been demonstrated very recently in a study by the Skolnick group showing that they strongly influence the kinetics of lipid membrane formation in BD simulations.<sup>26</sup>

Finally, still another area of biomolecular interest where HIs are likely to play a role is in modulating the diffusion of macromolecules in the highly crowded environments encountered *in vivo*. In particular, the Skolnick group has used the sophisticated Stokesian dynamics (SD) simulation method to show that intermolecular HIs are likely to be a principal cause of the dramatically slowed translational diffusion coefficients observed for proteins in the bacterial cytoplasm.<sup>27</sup> When rigid body models of macromolecules are used, the slowed diffusion caused by intermolecular HIs can be approximately captured in simpler BD simulations by making the short-time translational and rotational diffusion coefficients (which are inputs to the Ermak–McCammon algorithm<sup>5</sup>) dependent upon the local volume fraction occupied by other macromolecules.<sup>28,29</sup> This approach avoids the need to explicitly calculate HIs between macromolecules and has been shown to work surprisingly well in simulations of highly concentrated solutions of myoglobin

and hemoglobin at concentrations as high as 300–400 mg/mL.<sup>30</sup> However, this approach will not be applicable to describing the diffusive behavior of large assemblies of associated macromolecules, such as actin polymers, since omitting explicit calculation of the HIs neglects the fact that the random displacements of the molecules comprising the assembly will be highly correlated with one another. This issue becomes even more important when, as here, the molecules are themselves allowed internal flexibility.

The purpose of this manuscript is as follows. The studies cited all suggest that HIs are likely to be important to include for correctly modeling the kinetics of biological assembly, polymerization, and aggregation processes and that this is especially true when the component molecules of the system are modeled as flexible molecules. However, a major obstacle to including HIs in such simulations is the very heavy computational expense incurred when HIs are modeled. The level of the expense, most of which is caused by the need to calculate correlated random displacements for hydrodynamically interacting particles, has prompted a number of attempts to accelerate the calculation of HIs (see the Methods section). The present work is concerned with proposing and testing another such approach with a particular view to eventually performing simulations of biological assembly processes using flexible macromolecular models. To this end, an approximate treatment of the diffusion tensor is introduced in which intermolecular HIs are treated in a more simplified way than the intramolecular HIs. The approximation is first shown to work very well for describing the translational diffusion of protein assemblies, and somewhat less well for describing rotational diffusion. It is then shown that the method has advantages in speed and memory usage that make it potentially attractive for modeling large-scale biomolecular systems that cannot be addressed with more conventional RPY + BD simulations. Finally, the advantages, disadvantages, and potential applications of the proposed approach are discussed in the context of recent advances made by others in the modeling of HIs in large systems.

## MATERIALS AND METHODS

The Brownian dynamics algorithm of Ermak and McCammon<sup>5</sup> serves as the basis for all of the work that follows here. For a molecular system composed of  $N$  particles, the algorithm reads as follows:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \sum_j \Delta t \partial \mathbf{D}_{ij} / \partial \mathbf{r}_j + \sum_j \Delta t \mathbf{D}_{ij} \cdot \mathbf{F}_j / k_B T + \mathbf{R}_i$$

where  $\mathbf{r}_i$  is the position of particle  $i$ ,  $\mathbf{F}_j$  is the net force acting on particle  $j$ ,  $\mathbf{D}_{ij}$  is a  $(3 \times 3)$  submatrix of the complete  $(3N \times 3N)$  diffusion tensor,  $\mathbf{D}$ , and  $\mathbf{R}_i$  is a random displacement applied to particle  $i$ ;  $\Delta t$  is the time step,  $k_B$  is Boltzmann's constant, and  $T$  is temperature. Importantly, in the presence of HIs, the random displacements applied to particles are correlated with one another and must satisfy the relationship  $\langle \mathbf{R}_i \cdot \mathbf{R}_j \rangle = 2\mathbf{D}_{ij}\Delta t$ . It is to be noted that, as written, the algorithm accounts only for translational motions of the particles and does not account for hydrodynamic coupling between their translational and rotational motions. This is sufficient for the types of systems considered here, in which each diffusing molecule is modeled as being composed of multiple, bonded 'pseudoatoms': from

previous work, we know that the rotational motion of such molecules can be adequately described by applying a purely translational algorithm to each of the molecule's constituent pseudoatoms.<sup>6</sup> It is also to be noted that, in the present study, the diffusion tensor,  $\mathbf{D}$ , is described at the (pairwise) Rotne–Prager–Yamakawa level of theory;<sup>8,9</sup> for this form of  $\mathbf{D}$ , the divergence term  $(\partial \mathbf{D}_{ij} / \partial \mathbf{r}_i)$  is zero and so plays no further part in the analysis.

**Separation of the Diffusion Tensor into Intramolecular and Intermolecular Contributions.** Most discussions of the costs of HIs focus on the expense of calculating the correlated random displacements assigned to the pseudoatoms (particles) of the system. For now, however, we will focus instead on the calculation of the force-induced displacement terms, given by  $\Delta t \mathbf{D}_{ij} \cdot \mathbf{F}_j / k_B T$ . The presence of this term in the Ermak–McCammon algorithm means that the displacement of each pseudoatom  $i$  is a function not only of the force that acts upon it but also of the forces that act upon all other pseudoatoms with which it hydrodynamically interacts (i.e., all pseudoatoms,  $j$ , for which elements of the diffusion tensor  $\mathbf{D}_{ij}$  are nonzero); given the very long-range nature of HIs, this will be all other pseudoatoms in the system. Calculation of the complete set of force-induced displacements of all pseudoatoms (i.e., the complete set of  $\mathbf{D}_{ij} \cdot \mathbf{F}_j$  terms) therefore involves a dense matrix–vector multiplication, the cost of which is proportional to  $(3N)^2$  if each term is calculated separately. This is not a negligible cost, and it is worthwhile to consider, therefore, whether it might be decreased in some way. At a practical level, one way to accelerate it is to make use of GPGPU-based routines for computing dense matrix–vector products.<sup>31</sup> An alternative is to use fast, approximate methods of computing the product  $\mathbf{D} \cdot \mathbf{F}$ : in very recent work, for example, use has been made of fast multipole methods (FMM),<sup>32,33</sup> originally developed to rapidly compute long-range electrostatic and gravitational interactions,<sup>34</sup> to approximate long-range HIs between clusters of pseudoatoms. Here, we consider an alternative approach, in which calculations are instead accelerated by replacing the full diffusion tensor with a more approximate form and performing *exact* calculations of the product  $\mathbf{D} \cdot \mathbf{F}$ ; we will see later that the same approximation that accelerates the calculation of this term can also accelerate the calculation of the correlated random displacements.

Calculated at the Rotne–Prager–Yamakawa level of theory,<sup>8,9</sup> the complete diffusion tensor is constructed as follows. For all pairs of pseudoatoms  $i$  and  $j$  for which  $i \neq j$ , the elements of the  $(3 \times 3)$   $\mathbf{D}_{ij}$  submatrix are determined using the following relationships:

$$\mathbf{D}_{ij} = (k_B T / 8\pi\eta_s \mathbf{r}_{ij}) \times \{ [1 + 2\sigma^2 / 3r_{ij}^2] \mathbf{I} + [1 - 2\sigma^2 / r_{ij}^2] (\mathbf{r}_{ij} \mathbf{r}_{ij} / r_{ij}^2) \} \quad \text{for } r_{ij} \geq 2\sigma$$

$$\mathbf{D}_{ij} = (k_B T / 6\pi\eta_s \sigma) \times \{ [1 - 9r_{ij} / 32\sigma] \mathbf{I} + [3r_{ij} / 32\sigma] (\mathbf{r}_{ij} \mathbf{r}_{ij} / r_{ij}^2) \} \quad \text{for } r_{ij} < 2\sigma$$

where  $\mathbf{r}_{ij}$  is the vector connecting the centers of the pseudoatoms  $i$  and  $j$ ,  $r_{ij}$  is its magnitude,  $\sigma$  is the hydrodynamic radius of the pseudoatoms,  $\eta_s$  is the solvent viscosity, and  $\mathbf{I}$  is the  $(3 \times 3)$  identity matrix. For self-interactions (i.e., when pseudoatom  $i = j$ ) the off-diagonal elements of each  $(3 \times 3)$  submatrix are zero and the on-diagonal elements are determined from the Stokes–Einstein relationship:

$$\mathbf{D}_{ii} = (k_B T / 6\pi\eta_s \sigma) \mathbf{I}$$

Conceptually, the approximate form of the diffusion tensor proposed here can be thought of as being constructed in the following way. All intramolecular terms are calculated in exactly the same way as in the ‘complete’ approach described above: individual  $(3 \times 3)$   $\mathbf{D}_{ij}$  submatrices are computed for all pairs of pseudoatoms that are within the same molecule. All intermolecular terms, on the other hand, are treated at an approximate level, by (a) computing for each pair of molecules independently a single  $(3 \times 3)$  submatrix that provides an average description of their intermolecular hydrodynamic coupling and (b) assuming that this single  $(3 \times 3)$  submatrix can be used to describe the hydrodynamic coupling of all pairs of pseudoatoms shared between the two molecules. We compute the elements of the ‘average’  $(3 \times 3)$  submatrix by applying the RPY equations to the two molecules—assuming that both can be approximated as spheres—using each molecule's hydrodynamic radius and the distance between the centers of the two molecules, instead of the distance between individual pairs of pseudoatoms, to calculate the strength of the HI. In the present work, we deal with molecules that are not only roughly spherical but that are also of limited internal flexibility, so the effective hydrodynamic radius of each molecule can be predetermined prior to simulation by, for example, applying the hydrodynamics program HYDRO-PRO<sup>35,36</sup> to all-atom models of each molecule's structure. In other situations, where the molecules may change shape significantly during the simulation, it should be possible to calculate effective hydrodynamic radii of molecules ‘on the fly’ using Kirkwood's approximation.<sup>37–39</sup>

The above description outlines how an approximate diffusion tensor can be constructed in a form that is identical in size and shape to the complete (unapproximated) diffusion tensor. In practice, the approximation is implemented by replacing this single diffusion tensor with separate diffusion tensors that describe the intramolecular HIs of each molecule and one additional diffusion tensor that describes all of the intermolecular HIs; in the latter, each molecule is treated, as noted, as a single sphere. One significant advantage that follows from re-expressing the diffusion tensor in this way becomes apparent when we consider the computational cost of summing the  $\mathbf{D} \cdot \mathbf{F}$  terms in the Ermak–McCammon algorithm. Consider a system containing  $P$  copies of a molecule that contains  $Q$  pseudoatoms. Using the complete (unapproximated) diffusion tensor, the number of terms that one would need to compute in order to calculate all of the  $\mathbf{D}_{ij} \cdot \mathbf{F}_j$  terms is  $(3PQ)^2 = 9P^2Q^2$ . The same calculation using the approximation proposed here would require  $P(3Q)^2$  terms for the intramolecular contributions alone: this stems from the fact that the intramolecular  $\mathbf{D} \cdot \mathbf{F}$  terms for each of the  $P$  molecules would be calculated using separate diffusion tensors each containing  $3Q \times 3Q$  elements. To this cost would be added a total of  $3PQ + 9P(P - 1) + 3PQ$  operations necessary to compute the intermolecular contributions: the first term represents the cost of adding up the  $x$ ,  $y$ , and  $z$  components of the forces on  $Q$  pseudoatoms to obtain the total force acting on each of the  $P$  molecules; the second term reflects the fact that each of the  $P$  molecules interacts with  $(P - 1)$  other molecules and that calculating the contribution from each such interaction involves multiplying a  $3 \times 3$  diffusion tensor by the 3 components of the force acting on the other molecule; and the final term accounts for the fact that, for each of the  $P$  molecules, the combined force-induced



displacement in the  $x$ ,  $y$ , and  $z$  directions due to all of their intermolecular interactions must be added to each of its  $Q$  pseudoatoms. To illustrate the differences between the calculation of the  $\mathbf{D}\cdot\mathbf{F}$  terms using the complete and approximate diffusion tensors, some sample Fortran code is provided in the Supporting Information.

Since the total number of terms to calculate using the approximate diffusion tensor approach is  $9PQ^2 + 9P(P-1) + 6PQ$ , the saving associated with it depends nontrivially on both  $P$  and  $Q$ . When  $P$  is considerably less than  $Q^2$ , the cost of the approximate diffusion tensor calculations is dominated by the term  $9PQ^2$ , and the saving, relative to the complete diffusion tensor calculation, is  $\approx 9P^2Q^2/9PQ^2 = P$ . With  $P = 100$  and  $Q = 100$ , for example, the expected saving in operations is a factor of 99. When  $P$  begins to approach  $Q^2$ , however, the cost of the approximate diffusion tensor calculations becomes increasingly dominated by the term  $9P(P-1) \approx 9P^2$ , and the saving, relative to the complete diffusion tensor calculation, becomes instead  $\approx 9P^2Q^2/9P^2 = Q^2$ .

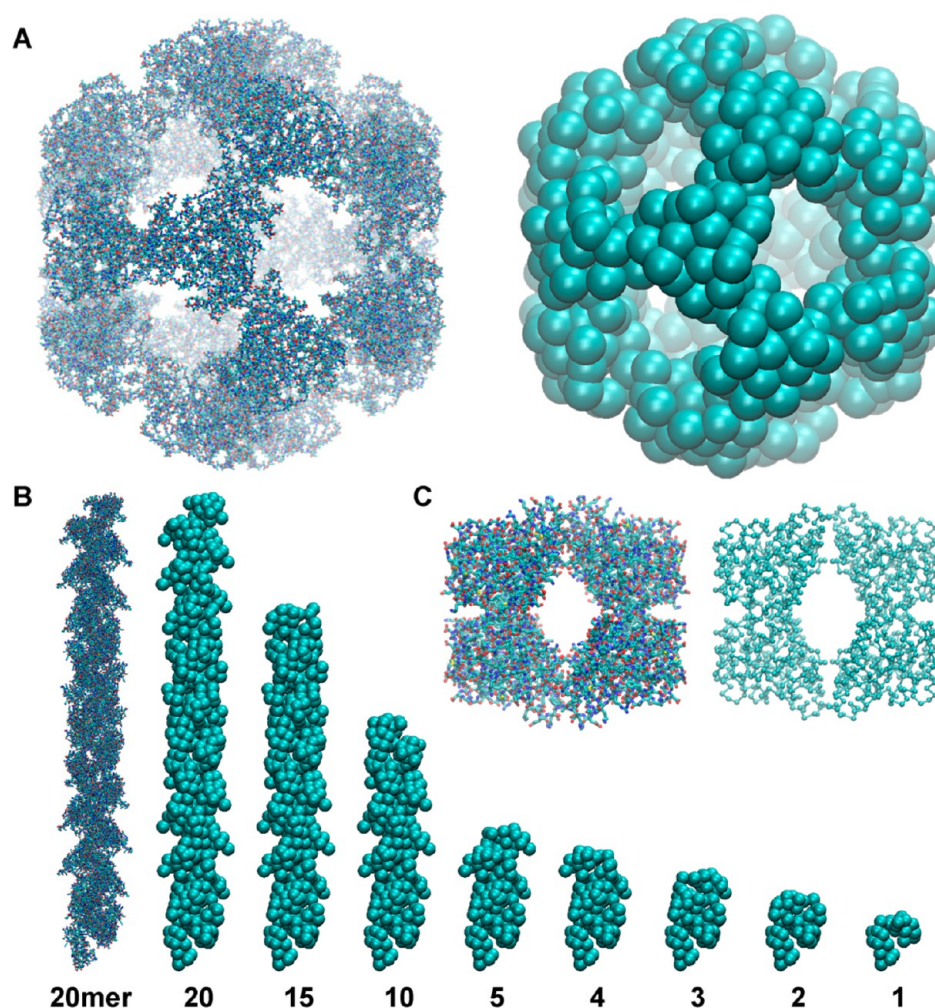
Importantly, the approximate diffusion tensor approach also has significant potential for accelerating the calculation of the correlated random displacements required by the Ermak–McCammon algorithm. To see this, we consider in turn the most commonly used methods for calculating these correlated random displacements. The most straightforward approach to calculating the random displacements, and crucially, one that involves no further approximation, is to perform a Cholesky decomposition of the diffusion tensor  $\mathbf{D}$  to obtain a lower-diagonal matrix  $\mathbf{S}$ , such that  $\mathbf{S}\cdot\mathbf{S}^T = \mathbf{D}$ . Once calculated,  $\mathbf{S}$  can be used to obtain a  $3N$  vector ( $\mathbf{R}_{\text{corr}}$ ) of suitably correlated random displacements simply by multiplying it by a  $3N$  vector ( $\mathbf{R}_{\text{unc}}$ ) of uncorrelated random displacements (of appropriate standard deviation and zero mean), that is, by using  $\mathbf{R}_{\text{corr}} = \mathbf{S}\cdot\mathbf{R}_{\text{unc}}$ . The Cholesky decomposition procedure used to generate  $\mathbf{S}$  has a computational cost that scales as  $(3N)^3$ , and although parallelized Cholesky decomposition methods have been developed,<sup>40</sup> allowing much larger systems to be dealt with than were previously possible, the  $N^3$  scaling remains a formidable obstacle to modeling very large-scale systems. Importantly, this remains the case even if one chooses to update the diffusion tensor (and its Cholesky decomposition) only periodically during the simulation.<sup>6,26,41–43</sup>

Although we do not consider the following issue in detail here, it is worth noting that BD simulations that use the Cholesky decomposition approach stand to benefit significantly from employing the approximate intermolecular diffusion tensor proposed here. From test calculations, we have seen that the Cholesky decomposition,  $\mathbf{S}$ , of the approximate diffusion tensor has a clearly simplified structure: in any given column of  $\mathbf{S}$ , the elements describing a given intermolecular interaction are all identical with one another. This simplified structure suggests two ways in which calculation of the correlated random displacements might be accelerated. First, and most obviously, the repeated occurrence of the same elements in  $\mathbf{S}$  can be exploited to accelerate the calculation of the  $\mathbf{S}\cdot\mathbf{R}_{\text{unc}}$  terms in exactly the same way that the repeated occurrence of identical elements in the approximated  $\mathbf{D}$  can be used to accelerate the calculation of the  $\mathbf{D}\cdot\mathbf{F}$  terms (see earlier). Second, the simplified structure of  $\mathbf{S}$  suggests that the Cholesky decomposition itself is likely also to be amenable to acceleration, since it ought to be possible to skip many steps of the algorithm. Exploiting this second aspect could have a big pay-off in terms of computational speed; however, since doing

so would require a detailed study of the algorithmic steps of the Cholesky decomposition, exploration of this issue is left for future work.

In attempts to circumvent the  $N^3$  scaling of the Cholesky decomposition, a number of alternative methods for the calculation of correlated random displacements have been devised;<sup>43–45</sup> an excellent comparison of a number of these methods as applied to simulations of flexible polymer chains has recently been reported.<sup>42</sup> The method that is used in this paper is due to Fixman and involves using a Chebyshev-polynomial approach to approximate the product of the diffusion tensor's square root ( $\sqrt{\mathbf{D}}$ ) and the  $3N$  vector of uncorrelated random displacements,  $\mathbf{R}_{\text{unc}}$ . It is to be noted that calculating the correlated random displacements using  $\mathbf{R}_{\text{corr}} = \sqrt{\mathbf{D}}\cdot\mathbf{R}_{\text{unc}}$  is just as valid as using  $\mathbf{R}_{\text{corr}} = \mathbf{S}\cdot\mathbf{R}_{\text{unc}}$ . Excellent descriptions of Fixman's method can be found in a number of papers,<sup>42,43,46</sup> and it is therefore not repeated in detail here. A key step in the method is to shift and scale the diffusion tensor  $\mathbf{D}$  in such a way that its minimum and maximum eigenvalues become  $-1$  and  $+1$  respectively; this requires that the minimum and maximum eigenvalues ( $\lambda_{\text{min}}$  and  $\lambda_{\text{max}}$  respectively) of  $\mathbf{D}$  itself first be known, at least to some degree of precision. Beyond this, the essential point to note about the method in the present context is that the final  $3N$  vector of correlated random displacements,  $\mathbf{R}_{\text{corr}}$ , is obtained by performing a series of matrix–vector product calculations in which the matrix is the scaled version of  $\mathbf{D}$ . Given that a scaled version of the approximate intermolecular diffusion tensor will share the crucial property of its parent, that is, that it will contain many elements that are identical to one another, it is straightforward to imagine that these matrix–vector multiplication steps might also be amenable to acceleration. Again, the practical implementation of this would make use of separate diffusion tensors for each molecule's intramolecular terms and a single diffusion tensor that accounts for all of the intermolecular terms; when implemented in this way, the expected speed-up for each matrix–vector multiplication required in the Chebyshev polynomial approximation would be the same as that outlined earlier for the  $\mathbf{D}\cdot\mathbf{F}$  calculations.

**Testing the Approximate Diffusion Tensor with Protein Models.** To test whether the proposed intermolecular diffusion tensor is sufficiently accurate to be practically useful, three different protein systems have been modeled in the present study. The first is the E2 catalytic core of the pyruvate dehydrogenase (PDH) multienzyme complex, which has been modeled here using the structure solved by Izard et al.<sup>47</sup> (pdb code: 1B5S). The complete E2 assembly is formed by the noncovalent association of 20 trimers of the E2 chain; the goal of the simulations reported here has been to determine whether the diffusional properties of the complete assembly can be accurately captured by simulations that replace the complete diffusion tensor with the approximate intermolecular diffusion tensor. In the simulations described here, it is the trimer that is the smallest unit simulated; since each trimeric unit of PDH contains 726 residues, the total number of residues in the final complex is 14520. The E2 complex is therefore sufficiently large that modeling it at even a residue-level of description with HIs included would be a major computational undertaking. To avoid this expense, a more simplified coarse-grained model has been constructed in which each trimer is modeled as 30 pseudoatoms whose positions are arranged in such a way that they capture reasonably well the overall shape of an E2 trimer (Figure 1A). As in our previous work,<sup>24</sup> the positions of the



**Figure 1.** Simulation models employed in this work. A. The E2 core of the PDH multienzyme complex; the crystal structure<sup>47</sup> is shown at left, and the very coarse-grained model used in BD simulations is shown at right. B. Oligomers of ParM; the crystal structure<sup>50</sup> is shown at left, and the various coarse-grained oligomers simulated are shown at right. C. The tetrameric enzyme DapA; the crystal structure<sup>51</sup> is shown at left, and residue-level BD simulation model is shown at right. See the text for explanations of the various approximate treatments of the diffusion tensor. This figure was produced with VMD.<sup>58</sup>

pseudoatoms within the envelope of the all-atom model of the trimer were determined using the ‘qpdb’ module of the Situs suite of programs.<sup>48</sup> Similar low-resolution models have been used by Geyer in BD studies of the effects of HIs on the diffusional properties of macromolecules.<sup>49</sup>

The second type of protein for which BD simulations have been carried out is the bacterial cytoskeletal protein ParM, which under appropriate experimental conditions forms long polymeric filaments; in this case, the atomic structure solved by Orlova et al.<sup>50</sup> was used (pdb code: 2QU4). The goal of the simulations in this case has been to model the diffusional properties of ParM oligomers containing 1, 2, 3, 4, 5, 10, 15, and 20 monomers and to ask whether the progressive slowdown in diffusion that accompanies lengthening of the filament can be captured by simulations that employ the approximate intermolecular diffusion tensor. Structures of each ParM oligomer were built using the symmetry transformations listed in the PDB file. As with PDH, since simulating the largest of these oligomers with a residue-level description of protein structure would be formidably expensive, use was again made of very coarse-grained structural models (see Figure 1B): in this case, 30 pseudoatoms were used to represent each ParM

monomer, with the qpdb utility again being used to optimize their relative positions.

The hydrodynamic radii assigned to the pseudoatoms of the PDH and ParM low-resolution models were determined by a series of calculations performed with the hydrodynamic modeling program HYDROPRO.<sup>35,36</sup> HYDROPRO was first used to compute the translational diffusion coefficient of the individual proteins from their all-atom models using the program’s standard settings. Hydrodynamic radii for the pseudoatoms of the very coarse-grained models were then optimized by trial and error with HYDROPRO calculations until the model’s predicted diffusional properties matched those predicted for the more structurally detailed models of the same units. The hydrodynamic radii parametrized in this way are 12.0 and 9.35 Å for PDH and ParM, respectively.

The third type of protein studied here is DapA, the tetrameric dihydrodipicolinate synthase from *Escherichia coli*. This protein is sufficiently small that it is much more straightforward to subject it to long BD simulations while retaining a reasonably high level of structural detail: DapA, therefore, was simulated with a residue-level (i.e.,  $\alpha$ -only) description using a modeling procedure identical to that which

we used in our previous studies of HI effects on protein diffusion.<sup>6</sup> The tetrameric form of the protein was taken from the high resolution crystal structure solved by Mirwaldt et al.<sup>51</sup> (pdb code: 1DHP). To provide an additional point of comparison, similar simulations were also carried out using a residue-level model of a ParM tetramer. In the case of these  $\alpha$ -only models, all hydrodynamic radii were set to 5.3 Å, as this value was found to produce translational and rotational diffusion coefficients in good agreement with HYDROPRO's predictions in our previous work.<sup>6</sup>

**Brownian Dynamics Simulation Details.** All BD simulations were performed using software written in-house. For the low-resolution models, structural integrity of the monomeric units was maintained by the imposition of harmonic bond stretch and bond angle deformation terms between pseudoatoms separated by less than 25 Å. The correct association of monomers within oligomeric assemblies was maintained by the use of 12–10 Lennard-Jones potentials set with an equilibrium distance equal to that measured in the initial model and with an energy well-depth equal to 1 kcal/mol; in effect, therefore, this represents a so-called Gō model of the type widely used to model protein folding events.<sup>6,52</sup> For the higher-resolution  $\alpha$ -only models, a Gō-type model essentially identical with that used in our previous work was employed: standard bond stretching, angle and dihedral deformation terms were included, interactions between pairs of residues in contact in the native state were treated with a 12–10 Lennard-Jones potential (well-depth of 1.0 kcal/mol), and interactions between pairs of residues not in contact in the native state were treated with purely repulsive  $1/r^{12}$  potentials. In practice, since the proteins modeled are of limited internal flexibility and thermodynamically stable under the conditions of the simulations, the non-native interaction terms are of little consequence.

Since the first goal of the present work was to establish the accuracy of the approximate intermolecular diffusion tensor, all BD simulations were performed under identical conditions with the only difference being the exact form of the diffusion tensor employed. For most systems studied, simulations were performed with (a) the complete, unapproximated diffusion tensor ('full HI'), (b) the approximate intermolecular diffusion tensor proposed here ('intermol HI'), (c) a diffusion tensor in which all intramolecular terms are included but all intermolecular terms are set to zero ('no intermol HI'), and finally, (d) a diffusion tensor in which all intramolecular and intermolecular HIs are set to zero ('no intramol HI'). For simulations performed with 'intermol HI', the hydrodynamic radii of individual molecules,  $\sigma_{\text{mol}}$ , were set to values of 36.496, 30.036, and 25.778 Å for PDH, ParM, and DapA, respectively; these values were determined from fitting the  $D_{\text{trans}}$  value computed by HYDROPRO using detailed structural models to the Stokes–Einstein relationship:  $D_{\text{trans}} = k_{\text{B}}T/6\pi\eta\sigma_{\text{mol}}$ .

In all simulations, correlated random displacements were computed using the Cholesky decomposition approach (see above). Although this is not the scenario in which we anticipate the approach developed here will be most practically useful—we think that it may be best used in conjunction with Fixman's method of calculating correlated random displacements—performing all simulations with the same protocol allows us to assess unambiguously the accuracy of the diffusional behavior obtained with the various approximate diffusion tensors; a similar approach has been followed recently by the Skolnick group.<sup>43</sup> In all simulations, the diffusion tensors and their

Cholesky decompositions were recalculated every 20 simulation steps; this is a conservative approach, as it has been shown in a number of works that updates of the diffusion tensor can be performed quite infrequently.<sup>6,43</sup> The calculation of the Cholesky decompositions was accelerated using the multi-threaded HSL\_MP54 routine developed by Dr. Jonathan Hogg,<sup>40</sup> who kindly guided us in its implementation within our code. Implementation of the Chebyshev approach within our code was heavily influenced by the implementation of Dr. Tihamer Geyer in his Brownmove package.<sup>49</sup>

Simulations of all proteins were performed for 10  $\mu$ s with coordinates saved at intervals of 100 ps; timesteps of 250 and 125 fs were used for the low-resolution and higher resolution  $\alpha$ -only models, respectively. Following the simulations, translational diffusion coefficients,  $D_{\text{trans}}$ , were computed by fitting the displacement of the center of geometry of the macromolecular assembly to the Einstein equation:

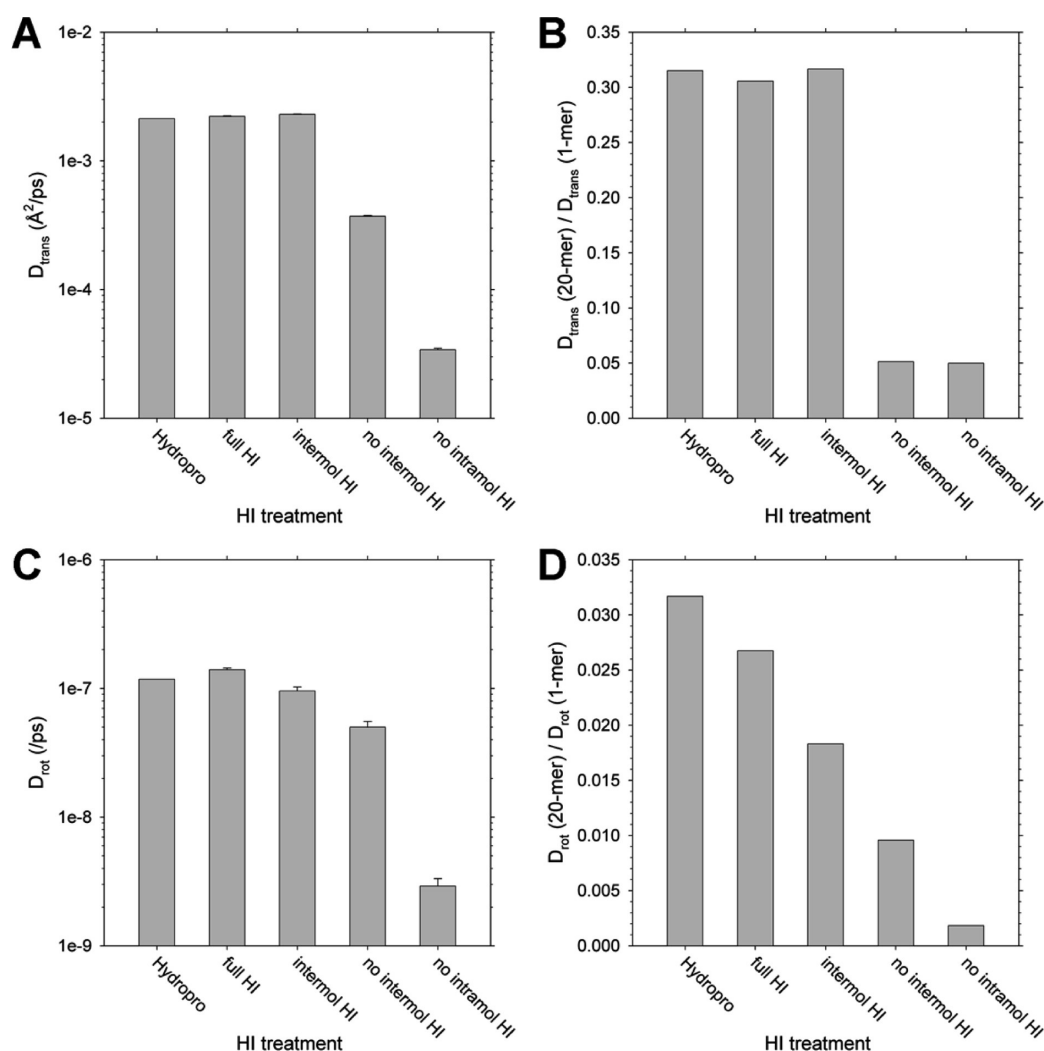
$$\langle \delta r^2 \rangle = 6D_{\text{trans}}\delta t$$

where  $\delta r$  is the distance moved in a time interval  $\delta t$  (here, 1 ns), and the brackets indicate an average over all simulation snapshots separated by  $\delta t$ . In all cases, error bars were obtained by breaking each 10  $\mu$ s trajectory into three equally sized blocks and calculating the standard deviation of the three independent estimates (for this purpose the first 100 ns of each simulation was discarded).

As in our previous work,<sup>6</sup> rotational diffusion coefficients were calculated from the time-dependent autocorrelation functions of three (effectively) mutually orthogonal vectors; these vectors were defined by aligning each initial model structure along its moments of inertia using the hydrodynamics program SOMO<sup>53</sup> and then finding pairs of pseudoatoms positioned close to the three axes but on opposite sides of the protein. Autocorrelation functions for the relative rotation of these three vectors were then computed from the coordinates of those pseudoatoms throughout the trajectory. The resulting correlation functions were fitted to single-exponentials of the form  $y = \exp(-t/\tau_{\text{corr}})$  over the first 100 ns of the decay, with  $\tau_{\text{corr}}$  being the fitted correlation time. Finally, the rotational diffusion coefficient,  $D_{\text{rot}}$ , was obtained from  $D_{\text{rot}} = 1/2\tau_{\text{corr}}$ . For the effectively spherical E2 core of PDH, the final reported  $D_{\text{rot}}$  was obtained by averaging the three independently computed rotational diffusion coefficients obtained from analysis of the entire production period of the simulation (0.1–10  $\mu$ s); a similar approach was used for the residue-level models of DapA and the ParM tetramer. For the more obviously nonspherical ParM oligomers, we focused on the rotational diffusion coefficient of the vector pointing along the longest axis of the protein; this is expected to be the one yielding the slowest estimate of the rotational diffusion coefficient. For all cases, error bars were obtained by repeating the analysis on 0.1–3.4  $\mu$ s, 3.4–6.7  $\mu$ s, and 6.7–10.0  $\mu$ s.

**Timing Estimates.** To assess the relative computational speeds of the various proposed methods, a series of short BD simulations were performed on systems containing large numbers of PDH trimers. In these simulations, which were each conducted for 10 ps, we compared the speed of simulations that used the Cholesky decomposition (performed every 20 simulation steps) with the speed of simulations that instead used Fixman's Chebyshev polynomial approximation method<sup>44</sup> for the calculation of correlated random displacements. As has been noted previously,<sup>43,44,46</sup> the latter method has the not insignificant complication that it requires that the





**Figure 2.** Diffusional characteristics of the E2 core of PDH. A. Absolute values of the translational diffusion coefficients obtained from BD simulations using a variety of HI treatments, compared with the value predicted by HYDROPRO.<sup>35,36</sup> B. Ratio of the computed translational diffusion coefficients of the full E2 core (20 trimers) to that of a single trimeric subunit. C. Same as A but showing results for rotational diffusion coefficients. D. Same as B but showing results for rotational diffusion coefficients.

minimum and maximum eigenvalues ( $\lambda_{\min}$  and  $\lambda_{\max}$  respectively) of the diffusion tensor be known prior to calculations. For reasons outlined in both the Results and the Discussion, we have chosen here to precompute  $\lambda_{\min}$  and  $\lambda_{\max}$  and to omit the time required for their calculation from the reported timings. Potential ways to minimize the time required to calculate the extreme eigenvalues, or to avoid it completely<sup>33,44</sup> are considered in the Discussion.

Since Fixman's Chebyshev polynomial method is an approximation, use was made of two different error estimates in order to determine when sufficient terms in the expansion had been added to achieve a desired level of accuracy. The simplest and most practical definition of the error in the resulting set of correlated random displacements is that proposed by Ando et al.:<sup>43</sup>

$$E_k = \|\check{\mathbf{R}}_{\text{corr}}(k) - \check{\mathbf{R}}_{\text{corr}}(k-1)\|_2 / \|\check{\mathbf{R}}_{\text{corr}}(k-1)\|_2$$

Here,  $E_k$  is the error estimate obtained after the addition of  $k$  terms in the expansion,  $\check{\mathbf{R}}_{\text{corr}}(k)$  is the approximate vector of correlated random displacements obtained after the addition of  $k$  terms, and  $\|\dots\|_2$  indicates the so-called  $l_2$ -norm of the vector (i.e., the square root of the dot product of the vector with

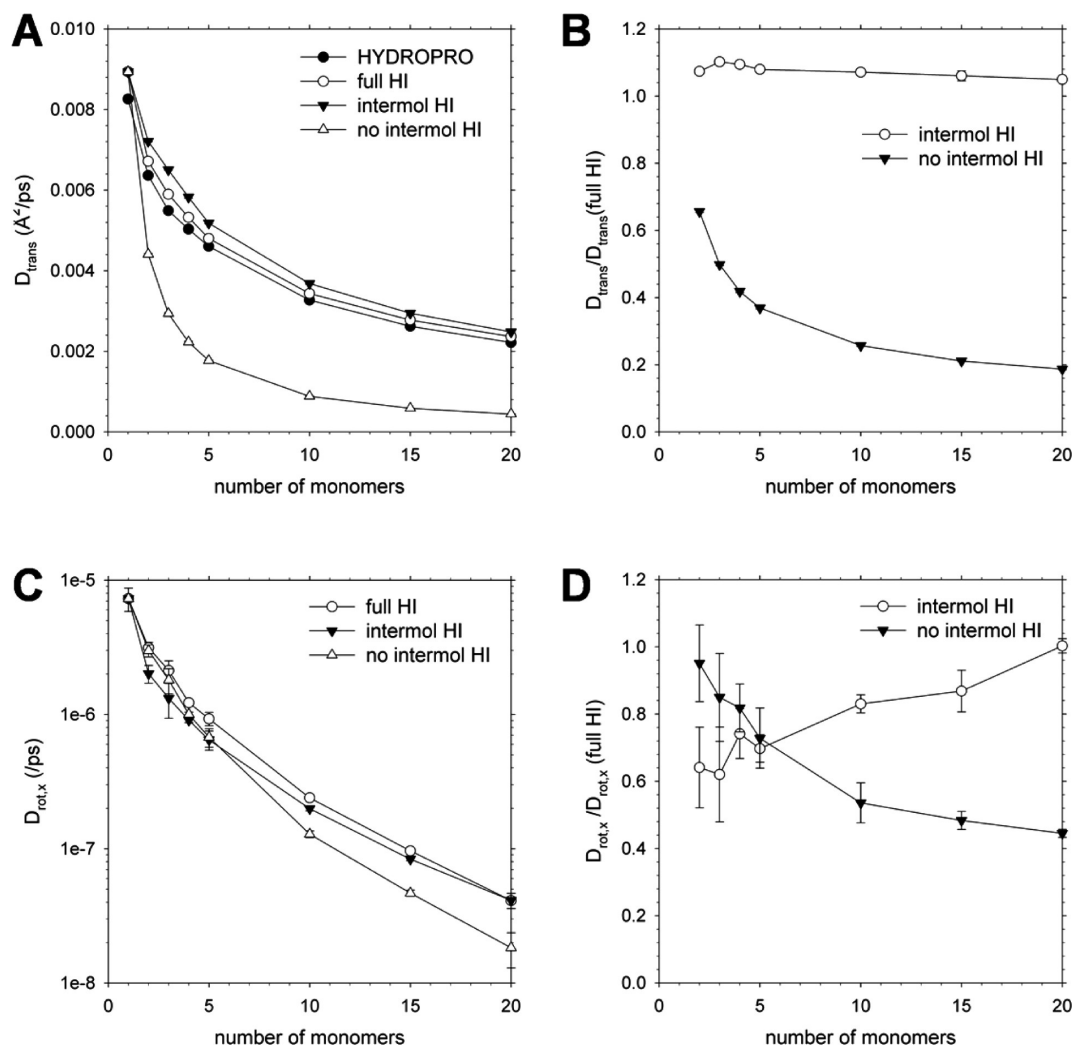
itself). An alternative, and apparently more stringent, error estimate is that devised by Jendřejek et al.:<sup>46</sup>

$$E_f = \sqrt{\{|\check{\mathbf{R}}_{\text{corr}}^T(k) \cdot \check{\mathbf{R}}_{\text{corr}}(k) - \mathbf{R}_{\text{unc}}^T \cdot \mathbf{D} \cdot \mathbf{R}_{\text{unc}}| / (\mathbf{R}_{\text{unc}}^T \cdot \mathbf{D} \cdot \mathbf{R}_{\text{unc}})\}}$$

Here again,  $E_f$  is the error estimate obtained after the addition of  $k$  terms in the expansion. Separate simulations were performed with  $E_k = 0.01$  and  $0.001$  and with  $E_f = 0.01$  and  $0.001$ . Each of these levels of accuracy has been employed in previous simulation studies; we note, however, that the Skolnick group has recently provided evidence that the level of accuracy provided by  $E_f = 0.001$  may be unnecessarily stringent, at least for the reproduction of translational diffusion behavior.<sup>43</sup>

## RESULTS

As outlined in the Methods, the present study proposes that one easily implemented way to accelerate the calculation of HIs in systems containing many flexible molecules is to replace the complete diffusion tensor that describes the interaction of all pseudoatoms in the system with a simplified diffusion tensor



**Figure 3.** Diffusional characteristics of ParM oligomers. A. Absolute values of the translational diffusion coefficients of ParM oligomers obtained from BD simulations using a variety of HI treatments, compared with the value predicted by HYDROPRO.<sup>35,36</sup> B. Ratio of the computed translational diffusion coefficients obtained with the approximate diffusion tensors to the values obtained with the complete RPY diffusion tensor. C. Same as A but showing results for rotational diffusion coefficients. D. Same as B but showing results for rotational diffusion coefficients.

that retains a full treatment of intramolecular HIs and combines it with an approximate treatment of intermolecular HIs. The relative computational efficiency of this approach is considered later in this article, but first, it is important to establish whether the approximation is sufficiently accurate to be useful. To investigate this, a number of different protein systems have been subjected to BD simulations using a variety of alternative treatments of the diffusion tensor; Figure 1 shows the model systems that have been studied. In all cases, the hydrodynamic radii of the simulation models have been adjusted in order to capture the diffusional properties of each assembly's smallest subunit. Importantly, the hydrodynamic radii have not been adjusted subsequently in an attempt to match the predicted diffusional properties of the higher-order assemblies; instead, it has been the purpose of the simulations to assess whether the diffusional properties of the larger assemblies can be accurately predicted.

**Diffusion of a Coarse-Grained Model of PDH.** The first system considered is a very coarse-grained model of the E2 core of the pyruvate dehydrogenase (PDH) multienzyme complex (Figure 1A). The complete structure of the E2 core contains 20 protein trimers, each of which has been modeled here as a

single, independent molecule. Figure 2A compares the translational diffusion coefficients,  $D_{\text{trans}}$ , of the complete E2 core computed from BD simulations that use four different forms of the diffusion tensor; it should be noted that the computed values are plotted on a logarithmic scale, which tends to suppress the quite large differences between the results of the various methods. The  $D_{\text{trans}}$  value obtained from BD simulations that use the complete, unapproximated RPY diffusion tensor (see the 'full HI' column) is in excellent agreement, as it should be, with the value predicted for the same model of the complete E2 assembly by HYDROPRO;<sup>35,36</sup> in the present study, the results obtained with the 'full HI' approach are to be considered the 'gold standard' against which the other methods should be judged. Encouragingly, a similar level of agreement is obtained from BD simulations that use the approximate treatment of the intermolecular diffusion tensor proposed in this work (see 'intermol HI' column): in this case,  $D_{\text{trans}}$  is overestimated by 8%. This good agreement is in contrast to what is obtained from BD simulations that include the intramolecular (i.e., intratrimer) HIs but neglect the intermolecular (intertrimer) HIs: in this case, the  $D_{\text{trans}}$  value computed from the BD simulations is a factor of 5.8 too low



compared with the HYDROPRO value. Finally, to further emphasize a point made in our previous study,<sup>6</sup> the results obtained from BD simulations that neglect HIs entirely ('no intramol HI') are abysmal. While such simulations are easily the fastest to perform, they also give easily the worst results: the  $D_{\text{trans}}$  value of the E2 assembly is, in this case, underestimated by a factor of 63.

Another way to consider the effects of intermolecular HIs on the  $D_{\text{trans}}$  values of a large complex such as the PDH E2 core is to plot the computed  $D_{\text{trans}}$  value of the complete assembly, containing 20 molecules, relative to the  $D_{\text{trans}}$  value computed from corresponding simulations of a single component molecule (in this case, a single trimer). The data plotted in this way are shown in Figure 2B. The three methods that explicitly account for intermolecular HIs, at least in some way, (HYDROPRO, 'full HI' and 'intermol HI') all predict that  $D_{\text{trans}}$  of the 20 × trimer should be reduced by a factor of ~3 relative to that of a single trimer. By contrast, the two methods that ignore intermolecular HIs entirely both predict that the  $D_{\text{trans}}$  of the 20 × trimer should be reduced by a factor of ~20 relative to that of a single trimer. Note that when expressed relative to the diffusion of a single trimer the 'no intramol HI' approach actually performs no worse than the 'no intermol HI' (Figure 2B): the very poor performance of the 'no intramol HI' approach in absolute terms seen in Figure 2A, therefore, is at least in part a consequence of its inability to capture the diffusional properties of the individual trimer.

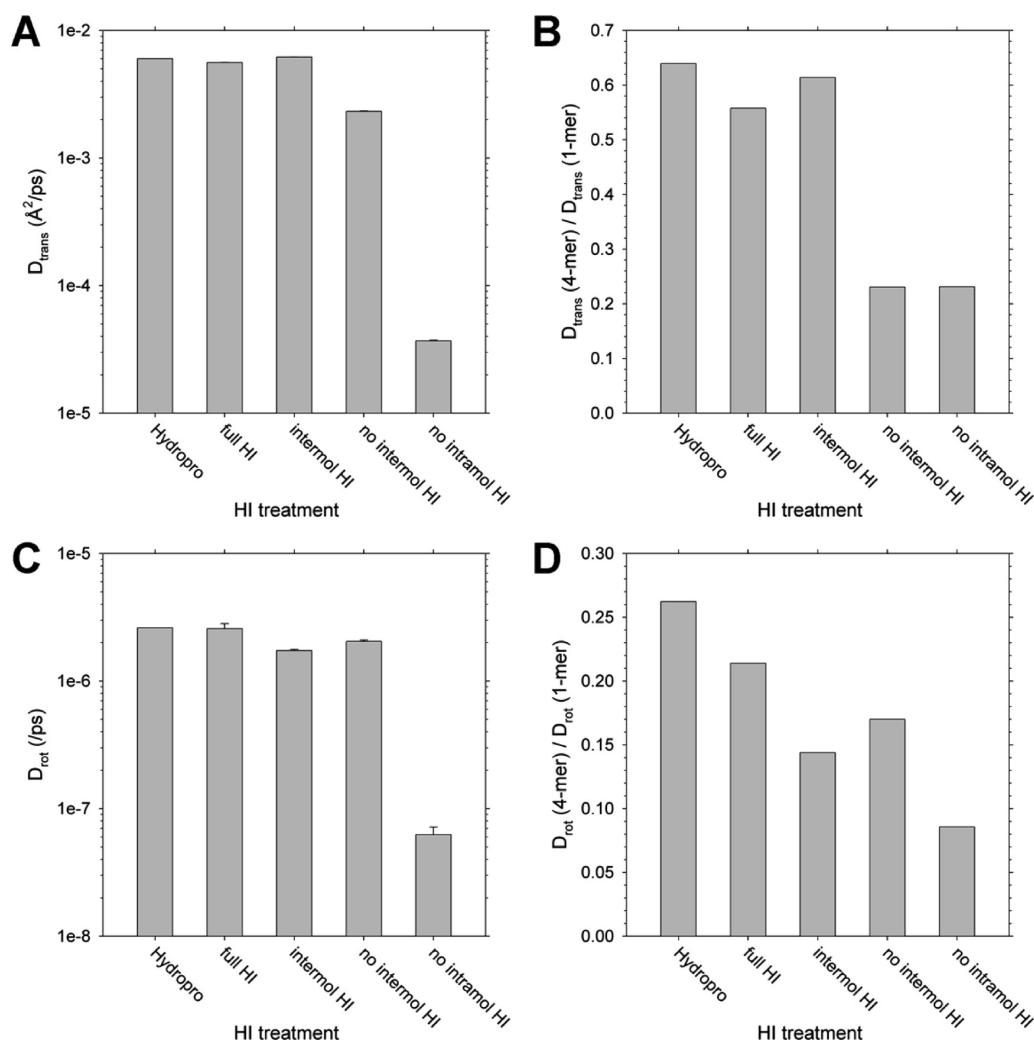
Figure 2C shows the corresponding comparison for rotational diffusion coefficients,  $D_{\text{rot}}$ , of the complete 20 × trimer assembly; as before, the  $D_{\text{rot}}$  values are plotted on a logarithmic scale. A similar story is obtained: the 'full HI' and 'intermol HI' results are close to those predicted by HYDROPRO, being 18% too high and 19% too low, respectively. The 'no intermol HI' results are poorer, a factor of 2.4 (i.e., 60%) too low, and the 'no intramol HI' results are again rubbish: the  $D_{\text{rot}}$  value is underestimated by a factor of 41. Figure 2D shows the  $D_{\text{rot}}$  values of the 20 × trimer assembly normalized to the corresponding values of single trimers. The first point to note is that the slowdown in rotational diffusion is much greater than that seen for translational diffusion (compare the y-axis of Figure 2B and D); this can be understood from the Stokes–Einstein relationships for isolated spheres, which predict that their  $D_{\text{trans}}$  and  $D_{\text{rot}}$  values should scale according to  $1/\sigma$  and  $1/\sigma^3$ , respectively, where  $\sigma$  is the sphere's radius. Plotted in this normalized way, the results for the 'intermol HI' approach proposed here are less good: while HYDROPRO predicts that the  $D_{\text{rot}}$  value of the 20 × trimer assembly should be a factor of ~32 lower than that of a single trimer, the 'intermol HI' method predicts that it should be decreased by a factor of ~55. While this is clearly not a wonderful result, it is still considerably better than that predicted by the 'no intermol HI' method that neglects intermolecular HI and includes only intramolecular HI, which predicts a decrease by a factor of ~104. Interestingly, in contrast to what was seen with translational diffusion (Figure 2B), when expressed relative to the diffusion of a single trimer the 'no intramol HI' approach does perform worse than the 'no intermol HI': it predicts that  $D_{\text{rot}}$  should be decreased by a factor of ~546 (Figure 2B). This suggests that the ability to correctly model the additional slowdown of rotational motion that occurs due to intermolecular interactions in larger assemblies does depend on how well the rotational motion of individual 'monomeric' units is captured. It is not obvious why this should be the case.

### Length Dependence of the Diffusion of ParM

**Oligomers.** The results shown above indicate that for the PDH 20 × trimer assembly the 'intermol HI' method proposed here does an excellent job of capturing the decrease in  $D_{\text{trans}}$  and an acceptable but clearly poorer job of capturing the decrease in  $D_{\text{rot}}$ , relative to that of a single trimer. To rule out the possibility that these results might be specific to effectively spherical protein assemblies, a similar set of BD simulations was performed on oligomers of ParM, a cytoskeletal protein from *E. coli* that forms long linear filaments (see Figure 1B); these simulations used protein models that are similarly coarse-grained to those used in the PDH simulations. Figure 3A compares the  $D_{\text{trans}}$  values computed from BD simulations of ParM oligomers of increasing length using three different treatments of HI: 'full HI', 'intermol HI', and 'no intermol HI'. For reference, the predictions obtained with HYDROPRO are also shown. Since all three of the BD methods model the intramolecular HI in exactly the same way, their predicted  $D_{\text{trans}}$  values all coincide with one another for the ParM monomer; they are all ~10% too high relative to the value predicted by HYDROPRO. For progressively longer ParM oligomers, the HYDROPRO-predicted  $D_{\text{trans}}$  values drop gradually, such that the predicted  $D_{\text{trans}}$  value for the 20-mer is a factor of 0.27 of that predicted for the monomer.

The two simulation models that include intermolecular HI (i.e., the 'full HI' and the approximate 'intermol HI' approach proposed here) both capture this length dependent decrease in  $D_{\text{trans}}$  very well: the predicted values of  $D_{\text{trans}}$  of the 20-mer relative to the monomer are factors of 0.26 and 0.27, respectively. In contrast, the  $D_{\text{trans}}$  value computed from the 'no intermol HI' BD simulations is a factor of 0.05 of that computed for the monomer. The different behaviors predicted by the two approximate methods of modeling HI can be more clearly illustrated by normalizing their predicted  $D_{\text{trans}}$  values by the  $D_{\text{trans}}$  values obtained from 'full HI' BD simulations (Figure 3B). Over the entire range of ParM oligomers studied here, the  $D_{\text{trans}}$  values computed from 'intermol HI' BD simulations are close to those predicted by the 'full HI' approach, exceeding those values by an average of ~10%. With the 'no intermol HI' approach, in contrast, the correspondence with the 'full HI' behavior starts badly and becomes progressively worse as the filament increases in length.

Figure 3C shows corresponding results for  $D_{\text{rot}}$  (note that HYDROPRO results are not shown for this case because we focus on rotation only of the long-axis of the molecule whereas HYDROPRO reports an isotropic  $D_{\text{rot}}$ ). As was the case with  $D_{\text{trans}}$ , all three of the HI models predict large decreases in the rotational diffusion of the long-axis of the ParM filament as the oligomer increases in length: with 'full HI', the  $D_{\text{rot}}$  of the 20-mer around its long axis is predicted to be a factor of ~269 lower than that of the monomer. The predicted  $D_{\text{rot}}$  values of the more approximate HI methods are normalized by the  $D_{\text{rot}}$  values obtained from 'full HI' BD simulations in Figure 3D. Interestingly, for short ParM oligomers the 'no intermol' HI  $D_{\text{rot}}$  values are clearly closer to the 'true' values (i.e., those predicted by the 'full HI' simulations) than are the 'intermol HI' values. This is something of an unpleasant surprise. However, as we move to longer ParM oligomers, the 'intermol HI' results quickly become superior and for 10-mers and above are within 20% of the 'full HI' values; in contrast, the 'no intermol'  $D_{\text{rot}}$  values get progressively worse. It appears, therefore, that the 'intermol HI' method's description of rotational diffusion, while clearly less than ideal for the smaller oligomers, is nevertheless



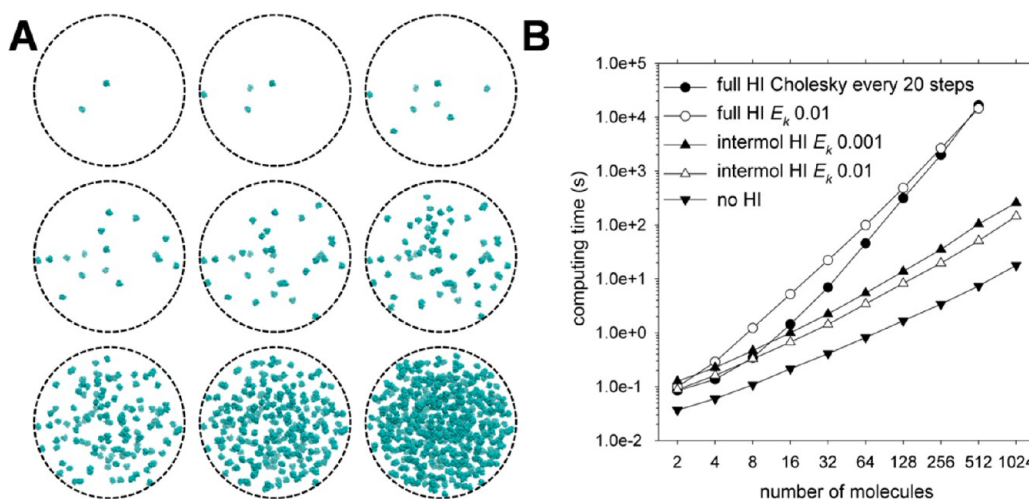
**Figure 4.** Diffusional characteristics of the residue-level model of DapA. A. Absolute values of the translational diffusion coefficients obtained from BD simulations using a variety of HI treatments, compared with the value predicted by HYDROPRO.<sup>35,36</sup> B. Ratio of the computed translational diffusion coefficients of the DapA tetramer to that of a single subunit. C. Same as A but showing results for rotational diffusion coefficients. D. Same as B but showing results for rotational diffusion coefficients.

likely to be increasingly accurate as the assembly (oligomer) increases in size.

**Diffusion of Residue-Level Protein Models: DapA and a ParM Tetramer.** The above results indicate that the approximate intermolecular HI method works quite well in the context of very coarse-grained models of proteins. To determine whether this remains true when more structurally detailed models are used two final systems were simulated using residue-level models that we have used in previous work.<sup>6</sup> Figure 4A shows the  $D_{trans}$  values computed from BD simulations of a residue-level model of the DapA tetramer, each chain of which is modeled with 290 pseudoatoms. Qualitatively, the results obtained are very similar to those obtained with the much more coarse-grained model of the PDH core (compare Figure 4 with Figure 2): the ‘full HI’ and ‘intermol HI’ results are in very good agreement with the predictions of HYDROPRO (7% too slow and 3% too fast, respectively), the ‘no intermol HI’ results are significantly underestimated (a factor of 2.6 too slow), and the ‘no intramol HI’ results are drastically underestimated (a factor of ~163 too slow). Normalizing the tetramer  $D_{trans}$  values by the monomer  $D_{trans}$  values (Figure 4B) also produces similar behavior to that

seen with PDH: both of the simulation methods that neglect the intermolecular HI overestimate the extent to which the tetramer’s  $D_{trans}$  value is slowed relative to that of the monomer.

For the rotational diffusion behavior (Figure 4C and D), the observed trends appear to be a hybrid of those seen with PDH (Figure 2) and with ParM (Figure 3). On the one hand, as was seen with PDH, the  $D_{rot}$  values obtained from the approximate ‘intermol HI’ and ‘no intermol HI’ are both in reasonable accord with the HYDROPRO prediction, while the ‘no intramol HI’ results are in catastrophically poor agreement (a factor of ~42 too slow). On the other hand, as was seen with the shorter ParM oligomers examined above (Figure 3D), the  $D_{rot}$  value obtained with the ‘no intermol HI’ approach is actually slightly better than that obtained with the ‘intermol HI’ approach. For smaller oligomers (e.g., tetramers), therefore, both the ParM and DapA results suggest that the ‘no intermol HI’ approach works slightly better at capturing rotational diffusion than does the ‘intermol HI’ method. A qualitatively identical set of results is obtained with a residue-level model of the ParM tetramer, which contains a similar number of pseudoatoms to the DapA tetramer model but which has a



**Figure 5.** Timing results for BD simulations of large-scale PDH systems. A. Pictures of the systems containing from left to right, top to bottom, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024 PDH trimers. B. Timing results obtained with various approaches for treating the HIs.

significantly more extended shape (see Figure S1, Supporting Information).

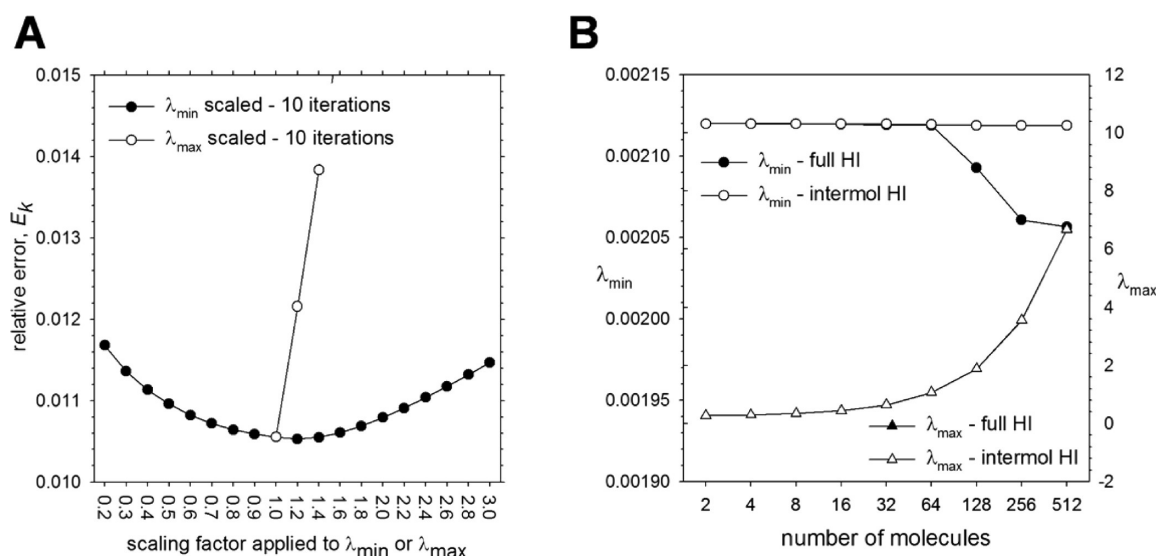
**Timings.** Taken together, the above results indicate that the ‘intermol HI’ approximation performs well in terms of reproducing the translational diffusion behavior of macromolecular assemblies and performs somewhat less well in terms of reproducing rotational diffusion behavior. For it to be truly worthwhile as an alternative to the ‘full HI’ treatment, however, it must be significantly faster in circumstances that matter to the simulator. Since one possible use of the approximate HI approach is in modeling assembly, polymerization, and aggregation processes, a series of test simulations have been conducted of the very coarse-grained PDH system described above, containing increasing numbers of molecules in a spherical simulation system (Figure 5A). Figure 5B shows how the computing time required to perform 10 ps of simulation on a single CPU increases with increasing numbers of molecules in the simulation from 2 to 4 to 8 to 16, etc. all the way up 1024 molecules. Results are shown for five simulation protocols: (a) ‘full HI’ using the Cholesky decomposition, (b) ‘full HI’ using Fixman’s method<sup>44</sup> for computing the random displacements with an error tolerance,  $E_k$ <sup>43</sup> of 0.01 (see the Methods), (c) ‘intermol HI’, again using Fixman’s method for computing the random displacements with an error tolerance,  $E_k$  of 0.01, (d) the same as (c) but with an error tolerance of 0.001, and (e) ‘no intramol HI’. It is to be noted that, in order to set up a relatively realistic scenario, the diffusion tensor and the costly Cholesky decomposition required for simulation condition (a) were performed only every 20 simulation steps.

As would be expected, the approach that ignores HI completely (‘no intramol HI’) is easily the fastest of the various methods tested (Figure 5B); for molecule numbers ( $N_{\text{mol}}$ ) of 64 and greater the cost of these simulations scales as  $N_{\text{mol}}^{1.10}$ , with the nonquadratic dependence being a result of the use of a nonbonded cutoff for computation of energetic interactions. For small numbers of molecules, the ‘full HI’ method in conjunction with the Cholesky decomposition is the fastest of the methods that attempt to model HI; for larger molecules, however, the cost of this approach increases rapidly such that the overall cost scales as  $N_{\text{mol}}^{2.82}$ , reflecting the expected cubic cost of the Cholesky decomposition. When the ‘full HI’ method is used in conjunction with Fixman’s

Chebyshev polynomial method the performance is clearly much worse at low molecule numbers and only becomes competitive with the Cholesky decomposition when the number of molecules reaches 512; in this case, the cost of the calculations scales as  $N_{\text{mol}}^{2.40}$ . When, however, the ‘intermol HI’ method is used instead, the situation changes significantly: now, the Chebyshev approximation out-performs the Cholesky decomposition for all molecule numbers above 8 when  $E_k = 0.01$  and above 16 molecules when  $E_k = 0.001$ . Increasing the stringency of the approximation by an order of magnitude from  $E_k = 0.01$  to  $E_k = 0.001$  results in the simulation time roughly doubling; in both cases, however, the scaling is surprisingly good:  $N_{\text{mol}}^{1.34}$  and  $N_{\text{mol}}^{1.45}$  for  $E_k = 0.01$  and  $E_k = 0.001$ , respectively. Simulations that use the alternative error estimate,  $E_f$ , proposed by Jendreck et al.<sup>46</sup> (see the Methods), in general, take longer (see Figure S2A, Supporting Information) since more iterations are typically required in order to satisfying the convergence criterion. In addition, the number of iterations required at each simulation step varies much more when  $E_f$  is used than when  $E_k$  is used (see error bars in Figure S2B, Supporting Information); this appears to be related to the fact that  $E_f$  does not always decay monotonically with increasing iteration number, in contrast to what is seen with  $E_k$ .<sup>43</sup>

It will be noted that neither of the plots shown in Figure 5B that use the ‘full HI’ approach have an entry for 1024 molecules. This is because the memory requirements of the complete diffusion tensor become sufficiently large that it cannot easily be accommodated on a typical server. For the 1024-molecule case, the complete diffusion tensor is a square matrix containing 1024 molecules  $\times$  3 dimensions  $\times$  30 pseudoatoms per molecule = 92160 entries on each line; the memory requirement for the entire matrix in single-precision, therefore, is 92160  $\times$  92160  $\times$  4 bytes = 34 GB. Of course, since the matrix is symmetric, it is possible in principle to decrease the memory footprint by at least a factor of 2; nevertheless, it should be clear that a point can quickly be reached at which the complete diffusion tensor will simply not fit into RAM. In contrast, the corresponding memory requirements of the approximate intermolecular HI method, implemented as described in the Methods, are much smaller: the 1024 intramolecular diffusion tensors together require 1024  $\times$  (90  $\times$  90)  $\times$  4 bytes = 33.2 MB, and the intermolecular





**Figure 6.** Issues relating to implementation of Fixman's Chebyshev polynomial method. A. Dependence of the error in computed random displacements,  $E_k$ , obtained for the 64 PDH trimer system after 10 iterations of the method as  $\lambda_{\min}$  and  $\lambda_{\max}$  are scaled away from their true values. Note that values of  $E_k$  for  $\lambda_{\max}$  scaled by factors below 1.0 are not shown because the values explode. B. Dependence of the true values of  $\lambda_{\min}$  and  $\lambda_{\max}$  for the complete RPY tensor and the approximate intermolecular tensor on the size of the system;  $\lambda_{\min}$  is plotted against the left-hand axis,  $\lambda_{\max}$  is plotted against the right-hand axis. Values are shown for the PDH systems illustrated in Figure 5A.

diffusion tensor requires  $1024 \times 1024 \times (3 \times 3) \times 4$  bytes = 37.7 MB (again, the potential factor of 2 saving has been omitted). With a total of 70.9 MB, therefore, the memory requirement of the 'intermol HI' approach is a factor of 480 less than that of the 'full HI' approach.

#### Note on the Use of Chebyshev Polynomial Method.

One factor that has been omitted from the above comparisons is the time necessary for determining the minimum and maximum eigenvalues of the diffusion tensor. While alternative methods have recently been described that enable rapid approximate calculation of correlated Brownian displacements to be performed *without* requiring extreme eigenvalues (e.g., using Krylov subspace methods described by Ando et al.<sup>43</sup> and by Liang et al.<sup>32</sup>), the Chebyshev method is sufficiently widely used that it is worth considering the extreme eigenvalue issue a little further. In principle, again as noted by others,<sup>46</sup> the extreme eigenvalues could be obtained relatively rapidly using established software packages such as Arpack (<http://www.caam.rice.edu/software/ARPACK/>), or using iterative Lanczos methods.<sup>33</sup> As outlined below, however, there appear to be two very simple additional approaches to estimating the limiting eigenvalues that are likely to be of practical use in settings similar to those explored here.

The first approach involves exploiting the error estimate,  $E_k$ , proposed by Ando et al.<sup>43</sup> Since it has been noted by others<sup>42,46</sup> that achieving a reasonable rate of convergence with the Chebyshev polynomial method can depend critically on having good estimates of  $\lambda_{\min}$  and  $\lambda_{\max}$ , this suggests that the rate of convergence of  $E_k$  might itself be used as a measure to rapidly obtain estimates for  $\lambda_{\min}$  and  $\lambda_{\max}$ . In principle, it should be possible to start with some initial guessed values of  $\lambda_{\min}$  and  $\lambda_{\max}$ , calculate the value of  $E_k$  obtained with a small, given number of Chebyshev terms (e.g., 10), and then iteratively adjust  $\lambda_{\min}$  and  $\lambda_{\max}$  in an attempt to find lower values of  $E_k$  using the same number of Chebyshev terms. To show that this is a practical possibility, Figure 6A shows how  $E_k$  obtained with 10 Chebyshev terms changes as either  $\lambda_{\min}$  (closed symbols) or  $\lambda_{\max}$  (open symbols) are scaled from their true values in a 64-

molecule PDH system. When  $\lambda_{\min}$  is scaled in either direction from its true value,  $E_k$  obtained after the addition of 10 Chebyshev terms (closed symbols) gradually increases. When  $\lambda_{\max}$  is increased from its true value,  $E_k$  increases much more rapidly; when decreased from its true value, the Chebyshev method blows up (i.e., does not converge). The same behavior is seen when more terms are used in the expansion (Figure S3A, Supporting Information), and similar behavior is also observed when the error estimate  $E_k$  advocated by Jendreck et al.<sup>46</sup> is used instead, although in that case the behavior is nonmonotonic and is therefore much less robust (Figure S3B, Supporting Information). Importantly, a two-dimensional surface representation of  $E_k$  computed following the addition of 10 Chebyshev terms shows that it should be possible to arrive at reasonable values of  $\lambda_{\min}$  and  $\lambda_{\max}$  even when neither of them is known in advance: the  $E_k$  surface is smooth, devoid of local minima, and it drops to a clear global minimum when  $\lambda_{\min}$  and  $\lambda_{\max}$  are at their true values (see Figure S3C, Supporting Information).

For the types of systems that are of interest here, an alternative way that  $\lambda_{\min}$  and  $\lambda_{\max}$  might be estimated is by simple extrapolation from the true values of  $\lambda_{\min}$  and  $\lambda_{\max}$  obtained for smaller systems. Figure 6B shows how the true values of  $\lambda_{\min}$  and  $\lambda_{\max}$  change with increasing number of PDH molecules (up to a maximum of 512) in a sphere of radius 750 Å. In this case,  $\lambda_{\min}$  is essentially independent of molecule number, although it shows a drop for the two largest systems due to molecules being (randomly) placed quite close to one another.  $\lambda_{\max}$ , on the other hand, shows a straightforward and highly predictable linear dependence on the number of molecules. Figure S4, Supporting Information, shows that the same basic behavior is obtained when the overall concentration of molecules is held constant. In both situations, therefore, it should be possible to obtain useful estimates of  $\lambda_{\min}$  and  $\lambda_{\max}$  from true eigenvalue calculations on smaller copies of the system.

## ■ DISCUSSION

The present work has described a method intended to allow more rapid modeling of HIs in systems that contain many interacting, flexible macromolecules. The test calculations on oligomeric assemblies indicate that the approximate description of intermolecular HIs advocated here yields a very good reproduction of the translational diffusion of macromolecular assemblies and a somewhat less good, but probably still acceptable, description of their rotational diffusion; importantly, the description of rotational diffusion appears to improve as the number of molecules in the assembly increases (Figure 3D). The timing calculations on the many-molecule PDH systems (Figure 5B) indicate that the method, when combined with Fixman's Chebyshev polynomial approach for computing random displacements, quickly outperforms the conventional 'full HI' tensor + Cholesky decomposition approach, and for very large systems has the additional advantage that it does not require storage of the complete  $3N \times 3N$  diffusion tensor in memory.

The model systems explored here also illustrate, once again, the importance of including HIs in situations where correctly capturing the translational and rotational diffusion of flexible molecules is important. Omitting HIs entirely (i.e., for intramolecular and intermolecular HIs) clearly produces atrocious results. Omitting the intermolecular HIs while retaining the intramolecular HIs on the other hand has, depending on the system, considerably less drastic consequences. At least for modeling *rotational* diffusion, the 'no intermol HI' approach can work surprisingly well for relatively small oligomers: in fact, for the smaller ParM oligomers (Figure 3D) and the DapA tetramer (Figure 4D), it produces somewhat better estimates of  $D_{\text{rot}}$  than does the approximate intermolecular HI approach proposed here. For larger oligomers, and for translational diffusion, however, omitting intermolecular HIs entirely is clearly much less acceptable. Nevertheless, the 'no intermol HI' approach may have some practical applications due to its very significant computational advantages. In particular, neglecting the intermolecular HIs entirely means that the entire diffusion tensor can be replaced by a set of independent, intramolecular diffusion tensors; this, in turn, means that the Cholesky decomposition of the entire diffusion tensor can be replaced by a set of independent Cholesky decompositions. For a system containing  $P$  copies of a molecule, each copy of which contains  $Q$  pseudoatoms, the cost of the Cholesky decomposition of the entire system would be  $(3PQ)^3$ ; the combined cost of the  $P$  independent Cholesky decompositions, on the other hand, would be only  $P(3Q)^3$ , which is smaller by a factor of  $P^2$  (i.e., 10 000 in the case of 100 molecules). In addition, it is to be noted that this cost would scale linearly with the number of molecules in the system. Depending on the system to be simulated, therefore, a combination of a 'no intermol HI' diffusion tensor with the Cholesky decomposition approach may well be a useful compromise.

For applications in which accurately reproducing the translational and rotational diffusion of macromolecular assemblies is critical, however, the molecule-based intermolecular HI method proposed here is a better alternative. In particular, the method combines an ability to reproduce diffusion coefficients that is superior to that of the 'no intermol HI' approach, with a computational efficiency for large systems that is much better than that which can be achieved when the

'full HI' approach is used. Here, we have shown that it performs well with the Chebyshev polynomial approach and that a major stumbling block for this approach, the need to know the extreme eigenvalues of the diffusion tensor, can be reduced at a practical level either by seeking values of  $\lambda_{\text{min}}$  and  $\lambda_{\text{max}}$  that lead to the most rapid convergence of the error criteria introduced by Ando et al.<sup>43</sup> or by Jendreck et al.,<sup>46</sup> or by estimating their values by extrapolating from smaller, but similar systems. Importantly, however, the same approximate intermolecular HI approach advocated here should work equally well with newer methods that have been developed to determine correlated random displacements and that do not require the calculation of  $\lambda_{\text{min}}$  and  $\lambda_{\text{max}}$ . In very recent work, the Skolnick group has shown that an alternative polynomial approach, based on Krylov subspace methods, is likely to be superior to the Chebyshev approach in typical applications of BD simulations,<sup>43</sup> and a very similar approach, based on earlier work by others,<sup>54</sup> has been used by Liang et al.<sup>32</sup> to solve the same problem. In common with the Chebyshev polynomial approach, these methods make repeated use of matrix–vector products in which the matrix is the diffusion tensor; in principle, therefore, both methods stand to gain computationally from using the present work's approximate intermolecular diffusion tensor. While not directly investigated in this work, it also seems worth reiterating that in addition to being well suited to use with methods for calculating approximate correlated random displacements, the simplified structure of the intermolecular diffusion tensor proposed here could also be exploited to accelerate the calculation of exact random displacements via Cholesky decomposition. At least one very nicely parallelized Cholesky routine is available for multicore architectures;<sup>40</sup> it might be a complicated matter, however, to develop a parallelized Cholesky method capable of exploiting the redundancy of information in the approximate intermolecular diffusion tensor.

Finally, it is important to consider this work in the context of a very recent advance in the computational modeling of HIs. As this work was being completed, two closely related implementations of the fast multipole method (FMM) to accelerate the computation of HIs at the RPY level of theory were reported. In one of these papers,<sup>33</sup> the authors used a kernel-independent FMM method to accelerate both the calculation of the  $\mathbf{D} \cdot \mathbf{F}$  terms and the calculation of the correlated random displacements with the Chebyshev approximation. In a second paper,<sup>32</sup> the same authors creatively recast the calculation of the RPY terms so that they could be calculated using an FMM code that had originally been used to compute electrostatic interactions; in addition, as noted, the authors used the Spectral Lanczos decomposition method in place of the Chebyshev approximation.

The advantages of these FMM approaches over the method reported here are clear: in principle, FMMs scale linearly with the number of particles, and they naturally subdivide a problem hierarchically such that the interactions of particles that are close in space are treated directly while those of particles that are more distantly separated are approximated using multipole expansions. Importantly, in both of the recent papers the  $O(N)$  scaling of the FMM methods for calculations of HIs been explicitly demonstrated with particle numbers up to 1 000 000.<sup>32,33</sup> However, one aspect that remains to be fully established is how well the methods will perform in conditions that are more typical of those encountered in biomolecular simulations. Based on the calculation details provided in those

papers, it appears that the timing results are for systems of randomly distributed particles at volume fractions of particles that range from  $1.33 \times 10^{-6}$  to  $1.33 \times 10^{-12}$ . In typical biomolecular conditions, the volume fraction may be in the range 0.1–0.3, and particles that are part of the same molecule will clearly not be randomly arranged. Given that the results obtained with the FMM approaches are extremely encouraging, however, it will be of considerable interest to see how the same methods perform in conditions that are more typical of those used in biomolecular simulations (e.g., using coarse-grained molecular models such as those employed here).

Interestingly, in a number of respects, the method presented here can be considered a “poor man’s FMM”. First, in both the FMM and the present approach, the clustering of particles is a key step in accelerating the calculations. In the FMM, clustering is based solely on the extent to which particles are in close spatial proximity to one another; in the present approach, on the other hand, clustering is based instead on whether particles share the same parent macromolecule. Second, while the FMM involves calculation of particle–particle, particle–cluster, and cluster–cluster interactions, the present method involves only particle–particle interactions (for pseudoatoms that are part of the same molecule) and cluster–cluster interactions (if they are in different molecules). Particle–cluster interactions could, in principle, be introduced to the present method in order, for example, to improve the modeling of the hydrodynamic behavior of an unfolded protein surrounded by many folded proteins. Third, while the FMM makes use of multipole expansions to describe cluster–cluster interactions, with the number of terms in the expansion being under the user’s control, the present approach, by assuming that each molecule is spherical, effectively truncates the description of cluster–cluster (i.e., molecule–molecule) interactions at the monopole level. The principal advantage of the current approach, therefore, relates to the ease of its implementation: in particular, by clustering pseudoatoms on the basis of their parent molecule; for instance, it eliminates the need to build the hierarchical tree structure that would otherwise be needed to cluster them spatially. It is to be noted that taking this approach is only reasonable because of the fact that many folded proteins are globular in shape and move as pseudorigid bodies; the spatial clustering approaches that are central to FMM and treecode methods,<sup>55</sup> on the other hand, are ‘blind’ to these molecular realities and are therefore unable to take advantage of this ‘shortcut’ to clustering. Of course, molecule-based clustering will probably not work well for highly flexible or elongated molecules, but it is quite possible to imagine extending the basic method presented here by introducing an intermediate level of clustering, for example, to better model proteins that consist of multiple domains connected by flexible linkers.<sup>10</sup>

Finally, in addition to being easy to implement, the results shown in Figure 5B for the PDH system indicate that the method presented can be surprisingly fast. The method’s speed in other situations will depend on how much work is required to compute the intramolecular HIs and how much is required to compute the intermolecular HIs: for systems dominated by the former, the scaling will be linear, while for systems dominated by the latter, the scaling will be quadratic. This means that, again depending on the system, it is quite possible that the present method may be computationally competitive with the FMM approach, as it avoids entirely the overhead associated with tree construction and computation of the

various multipole moments. However, determining whether this is indeed the case will require a true head-to-head comparison of the various methods in biomolecular applications. Given that there are now a number of very nice simulation packages that implement the Brownian dynamics technique,<sup>49,56,57</sup> establishing the most practical approach to including HIs for specific biomolecular problems will likely continue to be an important goal.

## ■ ASSOCIATED CONTENT

### 📄 Supporting Information

Sample Fortran code; diffusional characteristics of a residue-level model of a ParM tetramer; timing results; dependence of the error in the computed random displacements on the estimates of the minimum and maximum eigenvalues; dependence of eigenvalues of large-scale PDH systems in which the molecule density is kept constant. This information is available free of charge via the Internet at <http://pubs.acs.org>.

## ■ AUTHOR INFORMATION

### Corresponding Author

\*E-mail: [adrian-elcock@uiowa.edu](mailto:adrian-elcock@uiowa.edu).

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

This work was made possible by the support of NIH R01 GM087920 to A.H.E. The author is very grateful to Dr. Jonathan Hogg (Science & Technology Facilities Council, U.K.) for his help and patience in merging his multithread Cholesky decomposition code with the author’s own BD code. He is also thankful to Dr. Tihamer Geyer (University of Saarland, Germany) for providing access to his C++ code implementing the Fixman Chebyshev polynomial method: this was of invaluable help in guiding the author’s own rudimentary Fortran implementation.

## ■ REFERENCES

- (1) Tozzini, V. Minimalist models for proteins: A comparative analysis. *Q. Rev. Biophys.* **2010**, *43*, 333–371.
- (2) Saunders, M. G.; Voth, G. A. Coarse-graining of multiprotein assemblies. *Curr. Opin. Struct. Biol.* **2012**, *22*, 144–150.
- (3) McCammon, J. A.; Harvey, S. C. *Dynamics of Proteins and Nucleic Acids*; Cambridge University Press: Cambridge, U.K., 1987.
- (4) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, U.K., 1987.
- (5) Ermak, D. L.; McCammon, J. A. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.* **1978**, *69*, 1352–1360.
- (6) Frembgen-Kesner, T.; Elcock, A. H. Striking effects of hydrodynamic interactions on the simulated diffusion and folding of proteins. *J. Chem. Theor. Comput.* **2009**, *5*, 242–256.
- (7) Zimm, B. H. Dynamics of polymer molecules in dilute solution: viscoelasticity, flow birefringence, and dielectric loss. *J. Chem. Phys.* **1956**, *24*, 269–278.
- (8) Rotne, J.; Prager, S. Variational treatment of hydrodynamic interaction in polymers. *J. Chem. Phys.* **1969**, *50*, 4831–4837.
- (9) Yamakawa, H. Transport properties of polymer chains in dilute solution—hydrodynamic interaction. *J. Chem. Phys.* **1970**, *53*, 436–443.
- (10) Amorós, D.; Ortega, A.; García de la Torre, J. Prediction of hydrodynamic and other solution properties of partially disordered proteins with a simple, coarse-grained model. *J. Chem. Theor. Comput.* **2013**, *9*, 1678–1685.



- (11) Baumketner, A.; Hiwatari, Y. Influence of the hydrodynamic interaction on kinetics and thermodynamics of minimal protein models. *J. Phys. Soc. Jpn.* **2002**, *71*, 3069–3079.
- (12) Kikuchi, N.; Ryder, J. F.; Pooley, C. M.; Yeomans, J. M. Kinetics of the polymer collapse transition: The role of hydrodynamics. *Phys. Rev. E* **2005**, *71*, 061804.
- (13) Cieplak, M.; Niewieczerzał, S. Hydrodynamic interactions in protein folding. *J. Chem. Phys.* **2009**, *130*, 124906.
- (14) Szymczak, P.; Cieplak, M. Proteins in a shear flow. *J. Chem. Phys.* **2007**, *127*, 155106.
- (15) Szymczak, P.; Cieplak, M. Hydrodynamic effects in proteins. *J. Phys.: Condens. Matter* **2011**, *23*, 033102.
- (16) Pitard, E. Influence of hydrodynamics on the dynamics of a homopolymer. *Eur. Phys. J. B* **1999**, *7*, 665–673.
- (17) Kikuchi, N.; Gent, A.; Yeomans, J. M. Polymer collapse in the presence of hydrodynamic interactions. *Eur. Phys. J. E* **2002**, *9*, 63–66.
- (18) Pham, T. T.; Dünweg, B.; Prakash, J. R. Collapse dynamics of copolymers in a poor solvent: influence of hydrodynamic interactions and chain sequence. *Macromolecules* **2010**, *43*, 10084–10095.
- (19) Friedman, H. L. A hydrodynamic effect in the rates of diffusion-controlled reactions. *J. Phys. Chem.* **1966**, *70*, 3931–3933.
- (20) Deutch, J. M.; Felderhof, B. U. Hydrodynamic effect in diffusion-controlled reactions. *J. Chem. Phys.* **1973**, *59*, 1669–1671.
- (21) Wolynes, P. G.; Deutch, J. M. Slip boundary conditions and the hydrodynamic effect on diffusion-controlled reactions. *J. Chem. Phys.* **1976**, *65*, 450–454.
- (22) Shushin, A. I. Influence of hydrodynamic interaction on the diffusion-controlled reaction kinetics of molecules with highly anisotropic reactivity. *J. Chem. Phys.* **2003**, *118*, 1301–1311.
- (23) Antosiewicz, J.; McCammon, J. A. Electrostatic and hydrodynamic orientational steering effects in enzyme-substrate association. *Biophys. J.* **1995**, *69*, 57–65.
- (24) Frembgen-Kesner, T.; Elcock, A. H. Absolute protein–protein association rate constants from flexible, coarse-grained Brownian dynamics simulations: The role of intermolecular hydrodynamic interactions in barnase–barstar association. *Biophys. J.* **2010**, *99*, L75–L77.
- (25) Długosz, M.; Antosiewicz, J. M.; Zieliński, P.; Trylska, J. Contributions of far-field hydrodynamic interactions to the kinetics of electrostatically driven molecular association. *J. Phys. Chem. B* **2012**, *116*, 5437–5447.
- (26) Ando, T.; Skolnick, J. On the importance of hydrodynamic interactions in lipid membrane formation. *Biophys. J.* **2013**, *104*, 96–105.
- (27) Ando, T.; Skolnick, J. Crowding and hydrodynamic interactions likely dominate *in vivo* macromolecular motion. *Proc. Natl. Acad. Sci. U.S.A.* **2010**, *107*, 18457–18462.
- (28) Heyes, D. M. Mean-field-hydrodynamics Brownian dynamics simulations of viscosity and self-diffusion of near-hard-sphere colloidal liquids. *J. Phys.: Condens. Matter* **1995**, *7*, 8857–8865.
- (29) Sun, J.; Weinstein, H. Toward realistic modeling of dynamic processes in cell signaling: Quantification of macromolecular crowding effects. *J. Chem. Phys.* **2007**, *127*, 155105.
- (30) Mereghetti, P.; Wade, R. C. Atomic detail Brownian dynamics simulations of concentrated protein solutions with a mean field treatment of hydrodynamic interactions. *J. Phys. Chem. B* **2012**, *116*, 8523–8533.
- (31) Fujimoto, N. Faster matrix–vector multiplication on GeForce 8800GTX. IPDPS. *IEEE International Symposium on Parallel and Distributed Processing*, 2008; pp 1–8.
- (32) Liang, Z.; Gimbutas, Z.; Greengard, L.; Huang, J.; Jiang, S. A fast multipole method for the Rotne–Prager–Yamakawa tensor and its applications. *J. Comput. Phys.* **2013**, *234*, 133–139.
- (33) Jiang, S.; Liang, Z.; Huang, Z. A fast algorithm for Brownian dynamics simulation with hydrodynamic interactions. *Math. Comput.* **2013**, *82*, 1631–1645.
- (34) Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348.
- (35) Carrasco, B.; García de la Torre, J.; Zipper, P. Calculation of hydrodynamic properties of macromolecular bead models with overlapping spheres. *Eur. Biophys. J.* **1999**, *28*, 510–515.
- (36) Ortega, A.; Amorós, D.; García de la Torre, J. Prediction of hydrodynamic and other solution properties of rigid proteins from atomic- and residue-level models. *Biophys. J.* **2011**, *101*, 892–898.
- (37) Kirkwood, J. G. The general theory of irreversible processes in solutions of macromolecules. *J. Polym. Sci.* **1954**, *12*, 1–14.
- (38) Bloomfield, C.; Dalton, W. O.; van Holde, K. E. Frictional coefficients of multi-subunit structures. 1. Theory. *Biopolymers* **1967**, *5*, 135–148.
- (39) Robert, C. H. Estimating friction coefficients of mixed globular/chain molecules, such as protein/DNA complexes. *Biophys. J.* **1995**, *69*, 840–848.
- (40) Hogg, J. D. *A DAG-Based Parallel Cholesky Factorization for Multicore Systems*, Technical Report RF-RAL-2008-029; Rutherford-Appleton Laboratory: U.K., 2008.
- (41) Huang, J.; Schlick, T. Macroscopic modeling and simulations of supercoiled DNA with bound proteins. *J. Chem. Phys.* **2002**, *117*, 8573–8586.
- (42) Rodríguez Schmidt, R.; Hernández Cifre, J. G.; García de la Torre, J. Comparison of Brownian dynamics algorithms with hydrodynamic interaction. *J. Chem. Phys.* **2011**, *135*, 084116.
- (43) Ando, T.; Chow, E.; Saad, Y.; Skolnick, J. Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations. *J. Chem. Phys.* **2012**, *137*, 064106.
- (44) Fixman, M. Construction of Langevin forces in the simulation of hydrodynamic interaction. *Macromolecules* **1986**, *19*, 1204–1207.
- (45) Geyer, T.; Winter, U. An  $O(N^2)$  approximation for hydrodynamic interactions in Brownian dynamics simulations. *J. Phys. Chem.* **2009**, *130*, 114905.
- (46) Jendřejack, R. M.; Graham, M. D.; de Pablo, J. J. Hydrodynamic interactions in long chain polymers: Application of the Chebyshev polynomial approximation in stochastic simulations. *J. Chem. Phys.* **2000**, *113*, 2894–2900.
- (47) Izard, T.; Aeversson, A.; Allen, M. D.; Westphal, A. H.; Perham, R. N.; de Kok, A.; Hol, W. G. Principles of quasi-equivalence and Euclidean geometry govern the assembly of cubic and dodecahedral cores of pyruvate dehydrogenase complexes. *Proc. Natl. Acad. Sci. U.S.A.* **1999**, *96*, 1240–1245.
- (48) Wriggers, W.; Milligan, R. A.; McCammon, J. A. SITUS: A package for docking crystal structures into low-resolution maps from electron microscopy. *J. Struct. Biol.* **1999**, *125*, 185–195.
- (49) Geyer, T. Many-particle Brownian and Langevin dynamics simulations with the Brownmove package. *BMC Biophys.* **2011**, *4*, 7.
- (50) Orlova, A.; Garner, E. C.; Galkin, V. E.; Heuser, J.; Mullins, R. D.; Egelman, E. H. The structure of bacterial ParM filaments. *Nature Struct. Mol. Biol.* **2007**, *14*, 921–926.
- (51) Mirwaldt, C.; Korndorfer, L.; Huber, R. The crystal structure of dihydridipicolinate synthase from *Escherichia coli* at 2.5 Å resolution. *J. Mol. Biol.* **1995**, *246*, 227–239.
- (52) Hills, R. D., Jr.; Brooks, C. L., III Insights from coarse-grained Gō models for protein folding and dynamics. *Int. J. Mol. Sci.* **2009**, *10*, 889–905.
- (53) Brookes, E.; Demeler, B.; Rosano, C.; Rocco, M. Developments in the US-SOMO bead modeling suite: New features in the direct residue-to-bead method, improved grid routines and influence of accessible surface area screening. *Macromol. Biosci.* **2010**, *10*, 746–753.
- (54) Druskin, V.; Knizhnerman, L. Extended Krylov subspaces: Approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 755–771.
- (55) Barnes, J.; Hut, P. A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature* **1986**, *324*, 446–449.
- (56) Huber, G. A.; McCammon, J. A. BrownDye: A software package for Brownian dynamics. *Comput. Phys. Commun.* **2010**, *181*, 1896–1905.
- (57) Długosz, M.; Zieliński, P.; Trylska, J. Brownian dynamics simulation on CPU and GPU with BD\_BOX. *J. Comput. Chem.* **2011**, *32*, 2734–2744.

(58) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graph.* **1996**, *14*, 33–38.