Navigating molecular worms inside chemical labyrinths

M. Haranczyk[a] and J. A. Sethian[b,1]

[a]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720; and [b]Department of Mathematics, University of California, Berkeley, CA 94720

Edited by Alexandre J. Chorin, University of California, Berkeley, CA, and approved October 28, 2009 (received for review September 1, 2009)

Predicting whether a molecule can traverse chemical labyrinths of channels, tunnels, and buried cavities usually requires performing computationally intensive molecular dynamics simulations. Often one wants to screen molecules to identify ones that can pass through a given chemical labyrinth or screen chemical labyrinths to identify those that allow a given molecule to pass. Because it is impractical to test each molecule/labyrinth pair using computationally expensive methods, faster, approximate methods are used to prune possibilities, ''triaging'' the ability of a proposed molecule to pass through the given chemical labyrinth. Most pruning methods estimate chemical accessibility solely on geometry, treating atoms or groups of atoms as hard spheres with appropriate radii. Here, we explore geometric configurations for a moving ''molecular worm,'' which replaces spherical probes and is assembled from solid blocks connected by flexible links. The key is to extend the fast marching method, which is an ordered upwind one-pass Dijkstra-like method to compute optimal paths by efficiently solving an associated Eikonal equation for the cost function. First, we build a suitable cost function associated with each possible configuration, and second, we construct an algorithm that works in ensuing high-dimensional configuration space: at least seven dimensions are required to account for translational, rotational, and internal degrees of freedom. We demonstrate the algorithm to study shortest paths, compute accessible volume, and derive information on topology of the accessible part of a chemical labyrinth. As a model example, we consider an alkane molecule in a porous material, which is relevant to designing catalysts for oil processing.

fast marching methods | robotic navigation | accessible volume

Access of a molecule to a particular site or place in a chemical system as a process or accessibility of the molecule at this site as a property of either the molecule or the site are of considerable interest in chemistry, biochemistry, and material science. For example, a substrate has to have access to an active site of an enzyme (catalyst) before the enzymatic (catalytic) reaction can take place: if the active size is blocked by an inhibitor, it may be rendered inaccessible for the enzyme's (catalyst's) proper substrate. In general, many of the critical sites in proteins or synthetic catalysts are buried in clefts, pockets, and/or hidden cavities or represent channel systems that can be accessed or trespassed by substrates, solvent molecules, ions, or other molecules. The "accessible volume," which represents free volume that is available to a penetrating molecule, is useful in discussing physical properties such as diffusion, viscosity, and electrical conductivity in context of studies on glasses, polymers, and porous materials.

Predicting whether a molecule can traverse through chemical labyrinths of channels, tunnels, and buried cavities and possibly reach a destination is a difficult question to answer. It usually requires performing very long molecular dynamics simulations, and they may be a limiting factor in evaluating the suitability of a structure. Faster methods to estimate the possibility of a molecule trespassing through chemical labyrinths are desirable. Key features that one would like to be able to quickly determine include identifying the shortest transverse route, finding the largest probe that can transverse thought the system, and calculating accessible volume.

Most of the proposed approximation methods focus on solely geometrical factors (treating atoms as hard spheres with appropriate van der Waals radii) (1). However, even within such a simplified model, the problem has remained a challenge. The current state-of-the-art solutions can study paths of a spherical probe that mimics the molecule inside a convex hull constructed from atoms of a protein or material's framework. A tool for solving the shortest-path problem in this context was recently presented by Petrek et al. (2). They applied a form of Dijkstra's algorithm on a 3D grid with each point having assigned a cost related to the distance to the closest obstacle (closest protein atoms). Foster et al. (1) exploited the Delaunay triangulation technique to find the largest free sphere in a porous material. A Delaunay triangulation of a set of points defined by the centers of framework atoms generates a set of spheres (the Delaunay circumspheres) that occupy the voids and crevices within the framework. The empty Delaunay circumspheres frequently overlap to form a channel system, and the circular intersection defines the diameter of the restricting aperture connecting them. The diameter of these apertures places upper bounds on the diameter of the largest mobile probe sphere.

The biggest deficiency of these approaches is the restriction to a spherical probe to model the traversing molecule. In practice, most molecules of interest, even the simplest solvents or gases, rarely have a spherical shape, and treating molecules as such may lead to errors (see example in ref. 1). Fig. 1 illustrates the problem on an example of a chain of simple pentane hydrocarbon of relevance in the context of oil refining processes. Using the length of the molecule as a spherical probe diameter leads to a non-negligible overestimation of the required size of an opening (e.g., channel diameters) in a structure or material that could act as a fuel processing catalyst. It should be stressed that the precise matching between the molecule and an opening (pore, channel etc.) is usually important for efficient and selective catalysis; therefore, there is a need for improved models.

In this article, we pursue a more advanced approach, in which a spherical probe is replaced with one resembling the shape and flexibility of a "real" molecule. We model complex objects built from solid blocks connected by flexible links, which we call "molecular worms." As demonstrated in Fig. 1 *C* and *D*, such worms are able to change shape during the traversing of chemical structure, allowing them to reach areas not accessible to either a single large spherical probe (Fig. 1*A*) or rigid real-shape probes (Fig. 1*B*). This contribution significantly extends the range of probes and structures that can be efficiently examined. We cast the problem as an Eikonal equation in configuration space, in which the cost of entering each point in configuration space corresponds to the local geometry.

This equation is solved by using a variant of fast marching methods (3), which are Dijkstra-like methods to solve the Eikonal

Author contributions: M.H. and J.A.S. designed research, performed research, analyzed data, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1] To whom correspondence should be addressed. E-mail: sethian@math.berkeley.edu.

www.pnas.org/cgi/doi/10.1073/pnas.0910016106

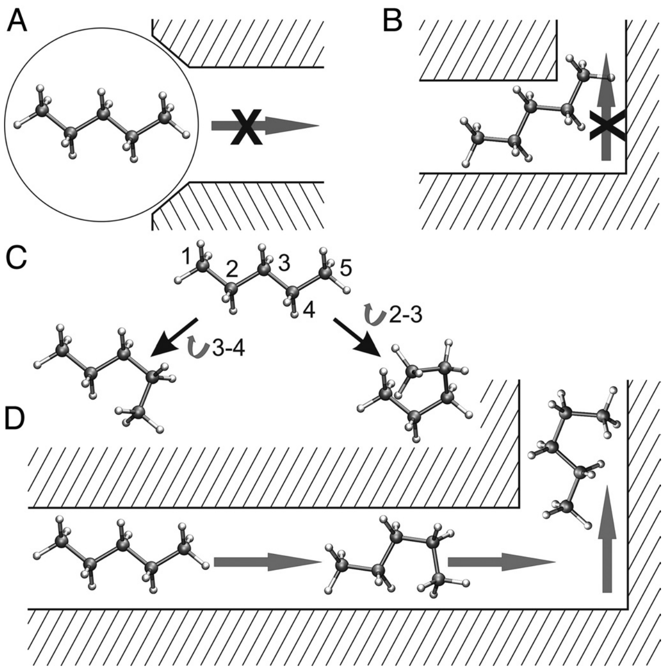

Fig. 1. Comparison of different approaches to study transport of a molecular worm in a hypothetical labyrinth. (*A*) Representing a molecule with a spherical probe leads to overestimation of its size and misjudgment of the possibility of the molecule to enter the structure. (*B*) A rigid molecule is stuck in a tight corner of the structure. (*C*) A flexible molecule with two internal degrees of freedom corresponding to rotations along the bonds connecting atoms 2–3 and 3–4 is shown. (*D*) A flexible molecule defined in *C* can trespass the structure and change its shape if required to maneuver in tight corners.

equation in $O(N \log N)$ time, where N is the total number of grid points in the computational domain. They have been successfully applied to problems in such topics as robotic navigation, fluid mechanics, and image analysis. Among other issues, the application of these methods to chemical pathways is challenging, because the path planning results in at least a seven-dimensional space to account for translational, rotational, and internal degrees of freedom.

Fast Marching Methods for Computing the Shortest Paths

Here, we review some work on fast marching methods to compute the shortest/cheapest path between points. Here, the cost is defined at every point in space, and for any path through space, the total cost is determined by integrating the cost function along that path. Our use of the word "shortest" is meant to mean that path that has the least total cost.

Dijkstra's Method and Optimal Paths. Consider a discrete optimal trajectory problem on a network. Given a network and a cost associated with each node, the global optimal trajectory is the most efficient path from a starting point to some exit set in the domain. Dijkstra's classic algorithm (4) computes the minimal cost of reaching any node on a network in $O(N \log N)$ operations. Because the cost can depend on both the particular node and the particular link, Dijkstra's method applies to both isotropic and anisotropic control problems. The distinction is minor for discrete problems, but significant for continuous problems. Dijkstra's method is a "one-pass" algorithm; each point on the network is updated a constant number of times to produce the solution. This efficiency comes from a careful use of the direction of information propagation and stems from the optimality principle.

We briefly summarize Dijkstra's method, because the flow of logic will be important in explaining fast marching methods. For simplicity, imagine a rectangular grid of size h in two space dimensions, where the cost $C_{ij} > 0$ is given for passing through each grid point $x_{ij} = (ih, jh)$. Given a starting point, the minimal total cost U_{ij} of arriving at the node x_{ij} can be written in terms of the minimal total cost of arriving at its neighbors:

$$U_{ij} = \min(U_{i-1,j},\ U_{i+1,j},\ U_{i,j-1},\ U_{i,j+1}) + C_{ij}. \qquad \textbf{[1]}$$

To find the minimal total cost, Dijkstra's method divides grid points into three classes: far (no information about the correct value of U is known), accepted (the correct value of U has been computed), and considered (adjacent to accepted). The algorithm proceeds by moving the smallest considered value into the accepted set, moving its far neighbors into the considered set, and recomputing all considered neighbors according to Eq. **1**. This algorithm has the computational complexity of $O(N \log(N))$; the factor of $\log(N)$ reflects the necessity of maintaining a sorted list of the considered values U_i to determine the next accepted grid point. Efficient implementation can be obtained by using heap-sort data structures.

Continuous Control: True Cheapest/Shortest Paths. Consider now the problem of finding the true cheapest path in a 2D domain: here, cost $C * h$ represents the cost of entering the subdomain of the region represented by the cell centered at grid point i,j. If one runs Dijkstra's method on this problem to find the optimal (cheapest/shortest) path from a starting position to an exit set, Dijkstra's method produces a solution to the PDE $\max(|u_x|,|u_y|) = C$.

Consider an easy version of this problem, in which one divides the unit square in cells, each of whose cost is 1, and the goal is to find the shortest path from (0, 0) to (1, 1). It is easy to see that as the grid becomes finer and finer, Dijkstra's method always produces a stairstep path with cost 2 and does not converge to the correct answer, which is a diagonal path with minimal cost $\sqrt{2}$ (see ref. 5). As h goes to 0, the true desired solution of this continuous Eikonal problem is given by the solution to $|u_x^2 + u_y^2| = C$.

Ordered Upwind Solvers for Continuous Isotropic Control. Nonetheless, algorithms that produce convergent approximations to the true shortest path for nonconstant metrics can be obtained by using Dijkstra's method as a building block: here, we follow the discussion in ref. 6. The first such algorithm was from Tsitsiklis (7), who obtains a control-theoretic discretization of the Eikonal equation, which then leads to a causality relationship based on the optimality criterion. Tsitsiklis' algorithm evolved from studying isotropic min-time optimal trajectory problems and involves solving a local minimization problem to update the solution. Tsitsiklis' algorithm uses Dijkstra-like ideas to order and sort the update procedure and has the same $O(N \log N)$ operation count.

A finite difference approach, based again on Dijkstra-like ordering and updating, was developed by Sethian (3, 8) for solving the Eikonal equation. Sethian's fast marching method evolved from studying isotropic front propagation problems and involves an upwind finite difference formulation to update the solution. Both Tsitskilis' method (7) and the fast marching method start with a particular (and different) coupled discretization and each shows that the resulting system can be decoupled through a causality property. In the particular case of a first-order scheme on a square mesh, the resulting quadratic update equation at each grid point is the same for both methods. We refer the reader to refs. 3, 7, and 8 for details on ordered upwind methods for Eikonal equations and ref. 9 for a detailed discussion about the similarities and differences between Tsitsiklis's method and the fast marching method. More recently, Sethian and Vladimirsky (9) have built a class of ordered upwind methods, based on Dijkstra-like methodology, for solving the more general class of optimal control problems in which the speed/cost function depends on both position and direction, which leads to a convex Hamilton–Jacobi equation. See ref. 9 for details.

We now briefly discuss the finite difference approximations behind fast marching methods.

Fast Marching Method Update Procedure. We approximate the Eikonal equation

$$|\nabla U| = C(x),$$

where $C(x)$ is the cost at point x in the domain. As a 2D example, we replace the gradient by an upwind approximant of the form:

$$\begin{bmatrix} \max(D_{ij}^{-x}u,\ -D_{ij}^{+x}u,\ 0)^2 + \\ \max(D_{ij}^{-y}u,\ -D_{ij}^{+y}u,\ 0)^2 \end{bmatrix}^{1/2} = c_{ij}. \qquad [2]$$

where we have used standard finite difference notation.

The fast marching method is as follows. Suppose at some time the Eikonal solution is known at a set of accepted points. For every not-yet accepted grid point with an accepted neighbor, we compute a trial solution to the above quadratic Eq. **2**, using the given values for u at accepted points, and values of ∞ at all other points. We now observe that the smallest of these trial solutions must be correct, because it depends only on accepted values that are themselves smaller. This "causality" relationship can be exploited to efficiently and systematically compute the solution as follows:

First, tag points in the boundary conditions as accepted. Then tag as considered all points one grid point away and compute values at those points by solving Eq. **2**. Finally, tag as far all other grid points. Then the loop is:

1. Begin loop: Let trial be the considered point with smallest value of u.
2. Tag as considered all neighbors of trial that are not accepted. If the neighbor is in far, remove it from that set and add it to the set considered.
3. Recompute the values of u at all considered neighbors of trial by solving the piecewise quadratic Eq. **2**.
4. Add point trial to accepted; remove from considered.
5. Return to top until the considered set is empty.

This is the fast marching method given in ref. 3: the key to efficient implementation lies in a fast heap algorithm to locate the grid point with the smallest value for u in set of considered grid points.

Application to Chemical Labyrinths

The above algorithm provides a fast and efficient technique to compute the minimal cost $U(x)$ satisfying the Eikonal equation, where x is a point in space, and boundary data are given on some set in the domain. Several issues are involved in extending this method to detecting pathways in chemical systems: (*i*) discretization of configuration space, (*ii*) constructing the cost function C on the basis of accessibility of a particular point in configuration space, (*iii*) accounting for more elastic boundaries and energy issues, (*iv*) computational issues, including initialization, execution, and analysis of fast marching runs and serial vs. parallel implementations, and (*v*) multiscale approaches for large systems. We address each in turn.

Discretization of Configuration Space. We imagine a given structure representing a chemical labyrinth. This structure is built from a set of atoms (spheres) and an associated molecular probe, defined as a set of atoms (spheres): additionally, we are given information on how they are connected. The first task is to build an appropriate discretization of configuration space.

To accurately describe a unique configuration of a set of N atoms forming a molecule, one needs $3N$ coordinates, corresponding to a three-tuple for each molecule. They are typically provided as a list of Cartesian coordinates of atoms. However, the structure may be transformed into internal coordinates, where three coordinates correspond to translation of the center of a molecule, three to rotation of the whole molecule (or two rotational axes for linear molecules), and the remaining $3N - 6$ coordinates define internal molecular parameters ($3N - 5$ for linear molecules). These internal parameters correspond to chemical bond lengths, angles between neighboring pairs of bonds (valence angles), and dihedral angles defined by sets of four atoms. Of all internal degrees of freedom, changes of dihedral angles have usually both the largest influence on overall shape and size of the molecule and the lowest energy cost of doing so, which justifies the need to consider these changes when studying molecules at room and elevated temperatures.

To model navigation of molecular worms in chemical labyrinth, we use a further approximation that limits configuration space to "essential" degrees of freedom, namely, three translational, three rotational, and one internal degrees of freedom that correspond to rotations of chemical bonds leading to the largest changes of overall shape and size of the molecule. These three sets of coordinates define the position in the real (3D) space, orientation, and conformation, respectively: we shall refer to configuration space as the set of all $6{+}l$ coordinates.

The resulting ($6{+}l$)-dimensional configuration space is then discretized by using a specified step sizes (length unit for position, degrees for orientation and conformation). The points along each of the $6{+}l$ dimensions is set as follows. For three translational coordinates, the resulting grid has to span over the desired part of chemical system and number of points has to be large enough to achieve requested accuracy. For the $3{+}l$ rotational degrees of freedom, the number of points has to reflect the required accuracy (e.g., long molecules would require dense sampling as small change in angular coordinate lead to big changes in molecular orientation). The maximum number of points in the discretized configuration space is limited by the available resources, in particular the memory size of a computer.

Constructing the Cost Function. Given a discretization of configuration space, the most straightforward cost function $C(x)$, where x is a point in configuration space, is defined as follows: $C = 1$ for each configuration point x that can be occupied, $C = \infty$ otherwise. Here, an occupiable point corresponds to a configuration in which the probes's atoms are neither colliding with themselves nor the atoms of the chemical labyrinth: here, a collision occurs when the distance between centers of any two atoms is smaller than the sum of their specified radii. We will say that a point in the real space is occupiable if there exist at least one corresponding configuration x for which $C(x) < \infty$.

To calculate C at each point x, the molecule in the proper conformation is built, then rotated to the required orientation and translated to position defined in x. Then, in a loop, distance between each atom of the molecule and each atom of the chemical labyrinth is calculated. The loop is broken if a collision is detected.

We note that one could choose to use a direct application of Dijsktra's method at this point rather than the more accurate fast marching method. It would reduce the compute time. However, we want to compute the true shortest path, as best we can, and suspect that the accounting for additional properties such as elasticity and temperature will require a fairly detailed and accurate method.

Elastic Boundaries and Energy Issues. In reality, probes can push against internal structures and sometimes squeeze through, because chemical structure is often not rigid and becomes more flexible and distorted with increasing temperature. Similarly, the probe's energy may be sufficient to induce local structure reorganization that leads to widening of narrow gorges.

The presented framework can be extended to account for such dynamical effects, by building a mathematical model in which the cost function at each point depends on the resistance/energy experienced. For example, a point might be deemed nonoccupiable if the energy is higher than the thermal limit for some given

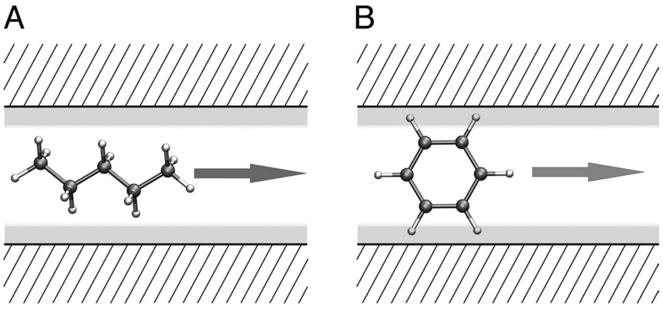

Fig. 2. The structure with ''rubber-covered'' walls. (*A*) Small molecules pass the structure without passing through the rubber part. (*B*) For molecules larger than a narrow gorge using rubber walls is the only option to pass.

conditions. Although this approach can be computationally very expensive, it could allow studying not only shape dependence, but also temperature dependence of the accessible void space of a chemical structure for a real molecular probe.

Because our goal in this work is to build efficient methods that allow pruning large sets of possible probes and chemical labyrinths, we amend the above idea through a simplified version of elastic boundaries, namely "rubber walls." Here, the cost function *C*, as defined previously, is modified as follows. Each configuration point x is defined by $6+l$ coordinates, $\{t_1,t_2,t_3,r_1,r_2,r_3,i_1.,.,i_l\}$, in the discretized configuration space and in which coordinates t_1–t_3 define position of the probe, coordinates r_1–r_3 define orientation, and internal coordinates i_1–i_l define conformation. For each nonoccupiable point x_{ia} ($C(x_{ia}) = \infty$):

```
For j,m,k from −p to +p such that $j^2+m^2+k^2 \leq p^2$
    Consider a point $x_a=\{t_1+\mathrm{j},t_2+\mathrm{m},t_3+\mathrm{k},r_1,r_2,r_3,i_1...,i_l\}$
    if $C(x_a) = 1$, then set $\mathrm{C}(x_{ia})=v$
endif
```

Here, v is set to some arbitrary value $\gg 1$ (e.g., 100). This introduces rubber walls with thickness of $p * \Delta x$, where Δx is length of step in the real physical space, and probes will be able to access this additional volume although at higher cost. For probes smaller than openings in their paths, the introduction of rubber walls will not effect the resulting path (in typical cases); however, a larger molecule will be given the ability to pass through what was previously inaccessible regions. See Fig. 2 for a clarifying graphical example. We note that more sophisticated definitions of rubber walls, where the cost of occupying the rubber section depends on the amount of overlap, are also possible but may be pointless if the rubber wall is one or two grid points thick (as in examples presented below).

Computational Issues. We now turn to implementation issues. To begin, the main bulk of the computational work for the fast marching method applied to probe molecules in chemical systems can be divided into two categories: (*i*) calculation of cost function, and (*ii*) execution of fast marching calculation and analysis of results. The latter step has different variations depending on the task (e.g., finding the shortest path or calculating the accessible volume).

The cost calculation is straightforward but computationally intensive. Therefore, we note two approaches that lead to improvement of run times. First, the cost calculation at each configuration point x_1 is independent from the cost calculation for any other point, x_2, and therefore this task is easily parallelized. Second, a heuristic approach can be introduced, motivated by adaptive mesh refinement approaches. Here, although the grid spacing is kept uniform, the cost function is not calculated at each point. Instead it is calculated every nth point in each dimension. Then, if the cost function at point x_k and x_{k+n} has the same value, the cost function at points $x_{k\ +\ 1}...x_{k+n-1}$ is assigned the same value. If the cost function value at point x_k and x_{k+n} are not of the same value, then the cost function at $x_{k\ +1}...x_{k+n-1}$ is calculated directly according to previously described procedure.

Having the cost function calculated in all grid points of discretized configuration space, one can proceed to the execution step. Here, we focus on several distinct and important goals: (*i*) traversal of a chemical labyrinth using any path, (*ii*) finding the shortest path through a space, (*iii*) finding all possible distinct paths through a space, and (*iv*) computing the total accessible volume.

Suppose we are searching for the shortest path connecting two points in configuration space corresponding to positions in the opposite sides of the chemical labyrinth. In most applications, the path's start-point and end-point orientation and conformation is of no concern. Hence, we can just search for a shortest path connecting any two points whose positions are on the opposite real-space faces (faces in 3D) of the grid. To do so, the fast marching method is initiated ($t = 0$) on all points which positions lay on the selected real-space face of the grid. Depending on the case, periodic boundary conditions are imposed on selected dimensions; typically, these are dimensions corresponding to rotational degrees of freedom (orientation, conformation). The front propagation is continued until the position of the last accepted configuration point corresponds to a point on the opposite face of the real-space grid. The actual shortest path can be then obtained by steepest descent walk on the front arrival time grid.

This framework may be further extended to predict all molecular transport pathways. The chemical labyrinth may have independent channel systems connecting different real-space faces of the grid, each corresponding to different accessible volume. We detect these sequentially in separate fast marching runs using the same grid. Each of these runs is initiated from a configuration point x_{i1} positioned on the real-space face of grid for which $C(x_{i1}) < \infty$. The fast marching method is stopped after all accessible configurations within the channel system were explored (until C of the accepted points is $< \infty$). All points visited are marked as belonging to this particular channel system. The procedure is repeated until there are no points laying on the real-space faces that $C < \infty$ and that have not been assigned to any channel system in the previous runs.

The real-space faces of the grid can be analyzed to find out whether they are connected with each other through a channel system, e.g., if two faces have at least one configuration point each, both corresponding to the same channel system, they are connected. The more detailed analysis of the real-space grid can be performed to identify topology and topography of the accessible channel system but it will not be discussed here.

Finally, we note that the accessible volume corresponding to a particular channel system is calculated from the number of real-space points belonging to that system. The real-space point belongs to the channel system if at least one configuration corresponding to that real-space position was visited during the corresponding fast marching run. The "real" volume can be estimated by further multiplication by a volume element in physical space, taking care to not duplicate the calculation for other configurations. The total accessible volume is given as a sum of accessible volumes corresponding to all channel systems.

Multiscale Approaches for Large Systems. The space of possible configurations of a typical molecule in a typical system of interest is immense. Even after the introduced reduction to "essential" $6+l$ degrees of freedom, the solution space is still quite large. An example nonrigid molecule presented in the Fig. 1*C* would require five coordinates to describe orientation and conformation. A coarse sampling of 30° steps leads to 12^5 ($\approx$250 k) configurations for each position in 3D space. Translation over fragments of the real space at the nanoscale level requires sampling over millions of points (e.g., 1 nm^3 with $\Delta x = 0.01$-nm steps along each direction). A rough estimate indicates 10^{11} to 10^{14} configurations to correctly under-

APPLIED MATHEMATICS

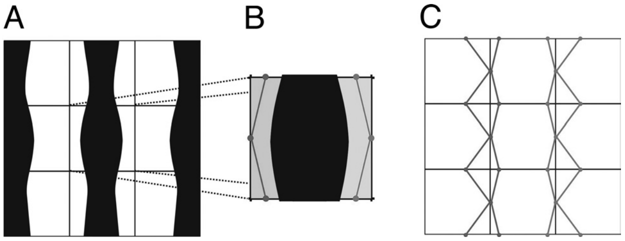

Fig. 3. A scheme presenting our multiscale approach. (*A*) Large system (void space is white) is divided into smaller patches. (*B*) Each patch is studied with fast marching to find existing channel systems (filled with two tones of gray) that are manifested on faces of a patch. Information about the connections of patch's faces via channel system is retrieved (marked as graphs in two tones of gray). (*C*) Large-scale graph representing connections of channels systems of each patch is built and studied.

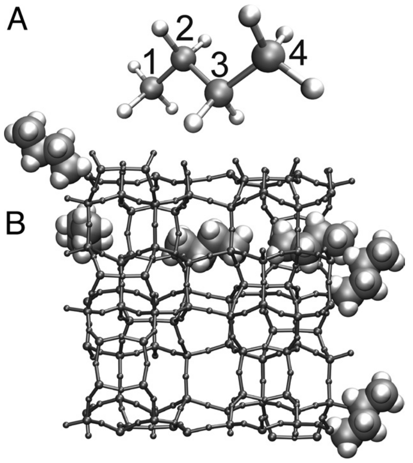

Fig. 4. A molecular worm inside chemical labyrinth. (*A*) Butane molecule and atom numbering. (*B*) The shortest path across one unit cell of periodic MFI zeolite structure is shown (larger radii of butane's atoms were used to improve clarity).

stand potential pathways and estimate accessible volume in typical systems. The larger numbers lead to significant memory and CPU time requirements that make it impossible to calculate by using the basic scheme described so far.

A multiscale approach can be used in cases in which the requested accuracy and/or the size of the system prohibits using the basic approach. Start by defining a large-scale system to span over the chemical system of interest. Next, divide this large-scale chemical system (in the real space) into a number of smaller rectangular cells, which are called patches. Each patch is an independent system (at the small scale) to be studied at the required accuracy. Usually the size of patches is at least one order of magnitude larger than step size chosen for the required accuracy (e.g., if step in translational degree of freedom is 0.01 nm, then size of a patch is 0.1 nm). Although the description of the system at the patch level is basically the same as on the coarse level, each patch itself describes structure that is much more uniform, because patches are of the size comparable with the size of atoms. In other words, there are patches that describe a fragment of the real space entirely contained within an atom, or, on the contrary, patches that are located in the void space and have all points accessible.

We now describe both a serial/sequential and a parallel approach to this multiscale representation. The basic difference is that information in the serial approach is calculated only when necessary, whereas in the parallel approach, all patches are calculated before execution of large-scale calculation. Thus, the parallel approach spends time characterizing patches that are never visited by the propagating front in the large scale; however, overall calculation time may be shorter as all patches are characterized in parallel at the initial stage.

To begin, we note that each real-space patch has eight physical space corners. If for all orientation/conformation pairs of all eight corner positions, the value of C is the same, the patch does not have to be fully characterized with our fast marching approach. Thus, at the patch level we are making the assumption/approximation that there are no internal obstacles wholly within the patch.

Sequential Multiscale Approach: Heuristic Technique for Determining Paths and Accessible Volumes. Begin with the system in 3D real physical space, with a particular starting and ending point, each with the same configuration (rotation and internal degrees in freedom). Call this configuration the "base state." First, at the boundaries of each patch, build a coarse-grain cost function by evaluating if each of the eight corners is an occupiable point for the base state: note that we need not check all possible configurations for each corner, only the base state. Set this coarse-grain cost function to one if all eight corners are occupiable points and infinity otherwise. Next, run the 3D fast marching method on the coarse-grain mesh from the starting point, until it stops. This characterizes the accessible volume on this coarse scale. Then, switch to the fine-scale representations for the fast marching method. This approach maintains an accurate calculation of the accessible region and detection of pathways: the compromise is that it may not find the shortest possible path.

Parallel Multiscale Approach. In the parallel multiscale approach each patch is characterized according to the already described procedure before the large-scale system is studied. For each patch, we obtain information about channel systems that are present in the patch, and manifested on a real space faces of it. Two channel systems of neighboring patches are connected if they share a common configuration point on the patches' shared face. All shared faces of all pairs of neighboring patches are analyzed to construct a graph that represents 3D topography of all channel systems at the

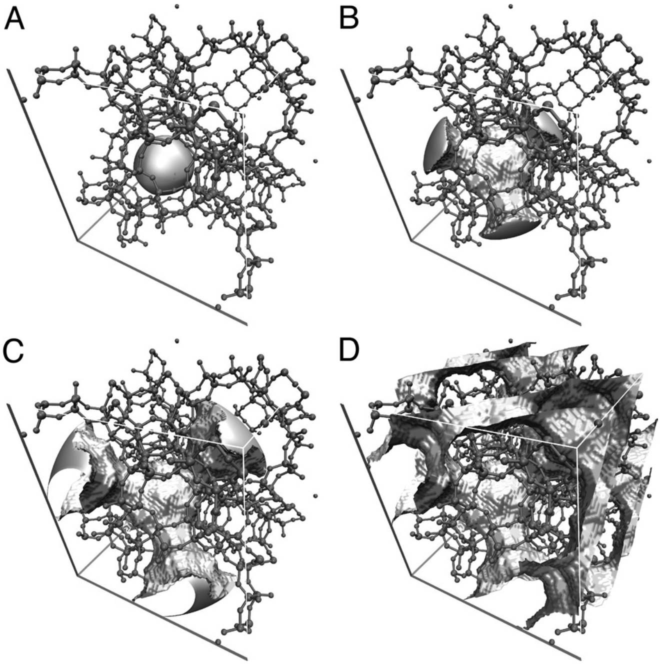

Fig. 5. An example fast marching front evolution in the 3D structure of one periodic unit cell of FAU zeolite. Shown are the isosurfaces corresponding to a number of arrival times 16 (*A*), 37 (*B*), 57 (*C*), and 113 (*D*). To improve clarity, small radii of FAU framework atoms were used.

large scale. To obtain the transport pathways and accessible volume, we then study propagation in the resulting graph. Analogously to the small-scale case, the answer to the question if a molecule can traverse the system can be obtained by finding a path (the shortest path in particular) connecting two real-space faces of the large-scale system. This can be achieved by executing the Dijkstra algorithm on the obtained graph. Accessible volume can be obtained as it is known which verticies of the graph have been visited during the appropriate front propagation, and the volume corresponding to the visited channel systems of the visited patched is known. The basic idea is illustrated in Fig. 3.

Demonstration: Test Cases

Here, we focus on a challenging application, namely the characterization of a class of porous materials, zeolites, in the context of oil refining. Here, the traversing molecules are hydrocarbons, long, flexible chain molecules very suggestive of molecular worms.

Zeolites are microporous mineral materials that have found wide use in industry since the late 1950s, with common applications as chemical catalysts, membranes for separations, and water softeners. They are particularly important as cracking catalysts in oil refinement (10, 11). There are $\approx$190 zeolite structures known to exist today. They constitute only a very small fraction of the $>$2.5 million structures that are feasible on theoretical grounds (12, 13). The development of a database of hypothetical zeolite structures has long been regarded as an important step toward "designer catalysts" (14) as it could, in principle, be screened for zeolites of any property. Brute-force screening of all possible structures involving molecular dynamics characterization is computationally infeasible: hence the need for rapid triaging based on initial analysis of various properties. One pruning criterion is the ability to traverse the zeolite structure or the accessible volume: ideally, one would apply a two-stage pruning, where in the first run all structures that have too small openings to even accommodate the building blocks of the molecular worms, followed by the high configuration space fast marching method.

Here, we verify if an alkane molecule is able to traverse chemical labyrinth of a zeolite. The answer is given by calculating shortest path between configurations corresponding to positions on the opposite sides of the system. Fig. 4 presents the resulting shortest paths of butane across one unit cell of periodic structure of MFI zeolite. This system is well documented in the literature (15–17) and can be used for verification purposes. In the case of butane being the traversing molecule, we consider one internal degree of freedom ($l = 1$) $-$ dihedral angle between atoms 1 and 4 (Fig. 4). The problem is solved by using our fast marching method in seven dimensions. For clarity of presentation, only a few snapshots along the shortest path are presented in Fig. 4. Using a limited memory machine, in this example we use a rather coarse grid when sampling configuration space. To determine position we used 0.02-nm steps leading to real space grid of $100 \times 100 \times 67$ points. Molecular orientations were sampled every 45° and the coordinate corresponding to internal degree of freedom was sampled at four points, every 90° starting at 0. The resulting grid has $100 \times 100 \times 67 \times 8 \times 8 \times 8 \times 4$, $\approx$1,372 M points.

The application of our algorithm to obtain accessible volume and accessible channel topology is illustrated in Fig. 5, which presents snapshots of front propagation in the real space within the atomic framework of FAU zeolite. The grid has $100 \times 100 \times 100 = 1$ M points, and the space was sampled by using a step of 0.025 nm. All positions visited by the moving front can be integrated to obtain accessible volume. Further analysis of the visited points gives information about the topography of the accessible channel system.

Our algorithm for detecting pathways and calculating accessible volumes can be used to investigate chemical systems and design new materials. It can be used in complex applications, to be discussed in detail elsewhere, to verify whether a given molecule is able to pass a channel of a transport protein located in a membrane or check whether a drug can reach an active site of an enzyme. We are currently using it to examine suggested designer materials, under consideration as molecular sieves for applications like carbon dioxide (CO_2) sequestration.

Summary and Additional Comments

The ultimate worth of the sort of algorithms proposed here rests in the ability to successfully triage between purely geometric hard sphere-based tests and full molecular dynamics simulations. Indeed, the key issue is to balance the accuracy of our molecular worm tests against the computing time of a full molecular dynamics simulation. However, it is important to note that the efficacy of our tests varies from one molecular structure to another: even within zeolites, every test is different.

The bulk of the computing time for the algorithms, as presently configured, lies in the computing the cost function for each possible configuration. As an example, computing the shortest path and available volume in a seven-dimensional example takes 10 min on a single Pentium 4 computer. However, the precomputation of the cost function for various configurations requires 60–80 min for this problem. This immediately suggests taking advantage of computational methodologies to decrease the compute time; one example is to spread the determination of the cost function over many different processors, because this is a highly parallel and disjoint operation. Additionally, one can explore computing the cost function "on-the-fly," only finding configurations when the expanding front reaches a new spot.

ACKNOWLEDGMENTS. We thank Berend Smit and Jenny Suckale for many valuable conversations. M.H. is a 2008 Glenn T. Seaborg Fellow at Lawrence Berkeley National Laboratory. M.H. was supported by the U.S. Department of Energy under Contract DE-AC02-05CH11231. J.A.S. was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract DE-AC03-76SF00098, and the Division of Mathematical Sciences of the National Science Foundation.

1. Foster MD, Rivin I, Treacy MMJ, Friedrichs OD (2006) A geometric solution to the largest-free-sphere problem in zeolite frameworks. *Microporous Mesoporous Mater* 90:32–38.
2. Petrek M, et al. (2006) CAVER: A new tool to explore routes from protein clefts, pockets, and cavities. *BMC Bioinformatics* 7:316–325.
3. Sethian JA (1996) A fast marching level set method for monotonically advancing fronts. *Proc Natl Acad Sci USA* 93:1591–1595.
4. Dijkstra EW (1959) A note on two problems in connection with graphs. *Numerische Math* 1:269–271.
5. Sethian JA (1999) *Level Set Methods and Fast Marching Methods* (Cambridge Univ Press, Cambridge, UK), 2nd Ed.
6. Andrews J, Sethian JA (2007) Fast marching methods for the continuous traveling salesman problem. *Proc Natl Acad Sci USA* 104:1118–1123.
7. Tsitsiklis JN (1995) Efficient algorithms for globally optimal trajectories. *IEEE Tran Automatic Control* 40:1528–1538.
8. Sethian JA (1999) Fast marching methods. *SIAM Rev* 41:199–235.
9. Sethian JA, Vladimirsky AV (2003) Ordered upwind methods for static Hamilton-Jacobi equations: Theory and algorithms. *SIAM J Numer Anal* 41:325–363.
10. Smit B, Maesen TLM (2008) Toward a molecular understanding of shape selectivity. *Nature* 457:671–677.
11. Smit B, Maesen TLM (2008) Mol simuls of zeolites: Adsorption, diffusion, and shape selectivity. *Chem Rev* 108:4125–4184.
12. Foster MD, Treacy MMJ (2009) *Atlas of Prospective Zeolite Frameworks*. Available at www.hypotheticalzeolites.net. Accessed August 31, 2009.
13. Earl DJ, Deem MW (2006) Toward a database of hypothetical zeolite structures. *Ind Eng Chem Res* 45:5449–5454.
14. Thomas JM, Klinowski J (2007) Systematic enumeration of microporous solids: Toward designer catalysts. *Angew Chem Int Ed* 46:7160–7163.
15. Narbeshuber TF, Vinek H, Lercher JA (1995) Monomolecular conversion of light alkanes over H-ZSM-5. *J Catal* 157:388–395.
16. van Bokhoven JA, et al. (2004) Observation of a compensation relation for monomolecular alkane cracking by zeolites: The dominant role of reactant sorption. *J Catal* 224:50–59.
17. Kotrel S, Knözinger H, Gates BC (2000) The Haag–Dessau mechanism of protolytic cracking of alkanes. *Microporous Mesoporous Mater 35*- 36:11–20.