JCTC Journal of Chemical Theory and Computation

Automatic Structure Analysis in High-Throughput Characterization of Porous Materials

Maciej Haranczyk*,[†] and James A. Sethian[‡]

Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States and Department of Mathematics, University of California, Berkeley, California 94720, United States

Received August 4, 2010

Abstract: Inspection of the structure and the void space of a porous material is a critical step in most computational studies involving guest molecules. Some sections of the void space, like inaccessible pockets, have to be identified and blocked in molecular simulations. These pockets are typically detected by visual analysis of the geometry, potential or free energy landscapes, or a histogram of an initial molecular simulation. Such visual analysis is time-consuming and inhibits characterization of large sets of materials required in studies focused on identification of the best materials for a given application. We present an automatic approach that bypasses manual visual analysis of this kind, thereby enabling execution of molecular simulations in an unsupervised, high-throughput manner. In our approach, we used a partial differential equations-based front propagation technique to segment out channels and inaccessible pockets of a periodic unit cell of a material. We cast the problem as a path planning problem in 3D space representing a periodic fragment of porous material, and solve the resulting Eikonal equation by using Fast Marching Methods. One attractive feature of this approach is that the to-be-analyzed data can be of varying types, including, for example, a 3D grid representing the distance to the material's surface, the potential or free energy of a molecule inside the material, or even a histogram (a set of snapshots) from a molecular simulation showing areas which were visited by the molecule during the simulation.

1. Introduction

1.1. Background. Porous materials contain complex networks of void channels and cages that are exploited in many different industrial applications. Zeolites, probably the most recognized class of crystalline porous materials, have found wide use in industry since the late 1950s. They are commonly used as chemical catalysts, in particular as cracking catalysts in oil refinement, membranes for separations and water softeners.[1−4] There is an increasing interest in utilizing zeolites as membranes or adsorbents for CO_2 capture applications. In addition to zeolites, other classes of nanoporous materials, such as metal organic frameworks (MOF)[5,6] and zeolitic imidazolate frameworks (ZIFS),[7] have enormous potential for gas separations or storage.[8,9] A key aspect for the success of any nanoporous material is that the chemical composition along with pore topology is optimal at the given conditions for a particular application. It should be noted that the number of possible pore topologies is extremely large. There are approximately 190 unique zeolite frameworks known to exist today[10] in more than 1400 zeolite crystals of various chemical composition and different geometrical parameters.[11] These experimentally known zeolites constitute only a very small fraction of more than 2.7 million structures that are feasible on theoretical grounds,[12,13] of which between 314k and 585k are predicted to be thermodynamically accessible as aluminosilicates, with the remainder potentially accessible via elemental substitution.[14] Databases of similar or greater magnitude can be developed for other nanoporous materials such as MOFs or ZIFs.

* To whom correspondence should be addressed E-mail: mharanczyk@lbl.gov.

[†] Lawrence Berkeley National Laboratory.

[‡] University of California, Berkeley.

10.1021/ct100433z © 2010 American Chemical Society
Published on Web 10/14/2010

Development of such databases brings great promise for discovery of new materials for many applications, particularly urgent ones like CO_2 capture. However, in order to make such discoveries possible, new computational and cheminformatics techniques have to be developed to characterize, categorize, and screen such large databases, bypassing manual and/or visual analysis in order to process and characterize a large number of structures in a high-throughput manner.

An important aspect of analysis of porous materials and their void space is the detection of inaccessible pockets, which can be occupied by guest molecules in computer calculations, even though such pockets are inaccessible in adsorption experiments. It is important to account for, and often artificially block or exclude, these pockets in the calculation of guest-accessible volumes, surface areas or prediction of guest-related properties using molecular simulation techniques. For example, in Monte Carlo (MC) simulations of adsorption, the blocking procedure can be a simple distance-check from the center of the small pockets and a rejection of all Monte Carlo trial moves that would place a molecule inside a certain radius. Alternatively, such pockets can be filled with dummy atoms. The importance of pore blocking in Monte Carlo simulations has been recently reemphasized by Krishna and van Baten.[15] (see for further examples and discussion.) Similar to MC simulations, molecular dynamics simulations have to account for such pockets in order to ensure that initial positions are chosen in the main channel system and not in such pockets.

Detection of inaccessible pockets is usually performed by visual analysis of so-called pore landscapes. The pore landscapes are isosurfaces corresponding to the maximum accessible free energy level (see Keffer et al.[16] for explanation on how isopotential energy surfaces are constructed.) Surfaces spreading across a periodic unit cell of a porous material encapsulate channels, whereas isolated surfaces correspond to inaccessible pockets. Similar approaches involve either visualization of histograms from molecular simulation which highlight all positions visited by a probe molecule during the time of simulation or simple visualization of isosurfaces corresponding to a distance from the material's surface equal to probe radii (Figure 1). Detection of inaccessible pockets in the last two cases is essentially the same as in the case of pore landscapes represented in terms of potential or free energy. An alternative approach involves analysis of abstract structure representations such as chemical hieroglyphs.[17] The latter presents structural building blocks such as cages and segments of channels, detailing their sizes and their connectivity. The notation used to represent connections between building blocks highlight the possibility of a guest molecule moving between the building blocks. Inaccessible pockets can be easily identified in such representation as building blocks without "valid" connections.

Although visual analysis is powerful, it can be error prone and its throughput is limited by the resources and abilities of a researcher, making it impractical for a high-throughput computational characterization pipeline. Instead, efficient computational techniques which automatically detect and characterize void space in porous materials are attractive alternatives. Detecting internal cavities has been explored in the context of proteins,[18] and the more general question of finding possible pathways through chemical system has been addressed in both proteins[19] and materials,[20−23] while detection of inaccessible pockets in porous materials seems not to have been pursued. Regardless, the vast majority of these studies have been based on geometrical considerations. All but one study attempted to study paths of a spherical probe representing the molecule inside a convex hull constructed from atoms of a protein or materials framework. However, most molecules of interest are rarely spherical and a spherical approximation significantly overestimates the required channel diameter for a molecular sieve.

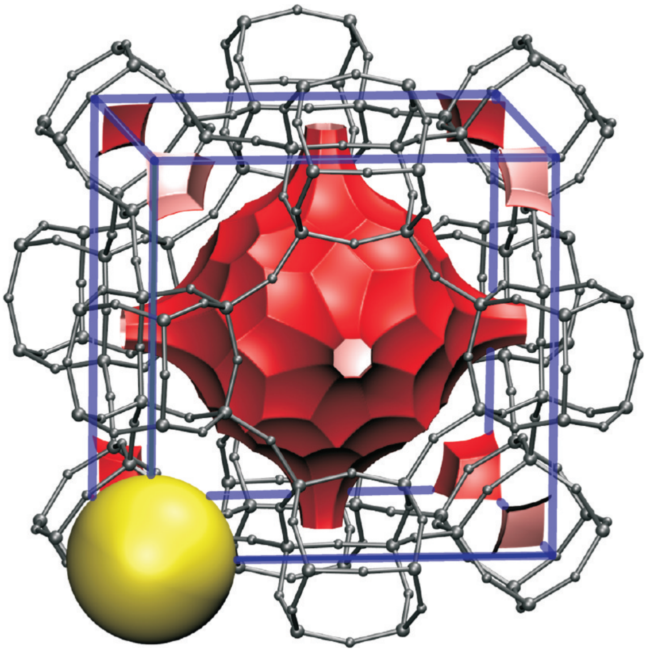

Figure 1. A periodic unit cell of LTA zeolite with highlighted probe accessible surface. Fragments of inaccessible pockets are visible in seven corners of the unit cell. One fragment is contained (blocked) by the yellow sphere.

1.2. A Different Approach: Molecular Worms, Path Planning, and Hamilton-Jacobi Equations. To address this issue, in previous work we built a more advanced approach, in which a spherical probe is replaced with one resembling the shape and flexibility of a "real" molecule.[24] In this approach, complex objects are built from solid blocks connected by flexible links, called "molecular worms". Such worms are able to change orientation and/or shape during the traversal of chemical structure, allowing them to reach areas not accessible to either a single large spherical probe or rigid real-shape probes. This converts the problem into one of path planning, which we characterize by an Hamilton-Jacobi-type Eikonal equation in configuration space; here, the cost of entering each point in configuration space corresponds to the local geometry. The equation is solved by using a variant of Fast Marching Methods,[25,26] which are Dijkstra[27]-like methods to solve the boundary value problems of the form of the Eikonal equation. Starting with an initial position for the front, the method systematically marches the front outward one grid point at a time, exploring all continuous pathways in the configuration space.

1.3. Current Work: A Comprehensive Approach to the Analysis of Porous Materials. In the current work, we present a comprehensive approach to analysis of porous materials, which allows detection and blocking of inaccessible pockets inside porous materials. Our approach does not require visualization and was developed with the goal to become part of a high-throughput pipeline for computational characterization of porous materials. In its essence, our approach mimics the visual analysis typically performed by researchers involving investigation of images representing pore landscapes. Unlike previously developed approaches to analyze accessibility, our framework is general. It allows analysis on the basis of geometry, potential or free energy profiles or histograms from molecular simulations. In particular, we use Fast Marching Methods to segment channels and inaccessible pockets from the 3D data representing any of the above. Additionally, we provide an algorithm to obtain blocking spheres for inaccessible pockets which can then be taken into account in molecular simulations, or excluded from calculations of accessible surfaces and volumes.

In this work, we present algorithms which examine accessibility issues based on data which can represent multiple characteristics, including the distance to the material's surface, the potential or free energy of a molecule inside a material, or the results of a molecular dynamics simulation showing areas which were visited by a molecule. For ease of exposition, in this work we only consider a three-dimensional configuration space, and hence our molecular worms can change position but not orientation or conformation. For this work, this translates into assuming spherical probes: it is important to stress that this is not a limitation of the method, and a full approach using molecular worms in higher dimensional configuration space is in progress and will be reported on elsewhere.

2. Methods

Our approach to automatic characterization of porous materials consist of two main tasks: (1) detection of channels and inaccessible pockets, and (2) generation of blocking spheres for inaccessible pockets. The detection of the latter relies on their segmentation from a guest molecule-dependent 3D data representing a repeating fragment of a porous material (e.g., 3D grid representing free energy of a molecule inside a periodic unit cell of a material). In the following subsections, we outline mathematical tools used for segmentation of space, their implementation to detect channels and inaccessible pockets, and the procedure of generating blocking spheres.

2.1. Mathematical Background. We cast the problem of segmentation of space in a periodic unit cell of a material as an Eikonal equation in domain representing this fragment and in which the cost of entering each point in the domain corresponds to its ability to be occupied:

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

Here, U is the minimal total cost and $C(x)$ is a cost function defined at each point x in the domain. Abstractly, this cost function is defined at the beginning of the problem, and the solution $U(x)$ to the above problem represents the total cost, which is the smallest obtainable integral of $C(x)$, considered over all possible trajectories throughout the computational domain from a start point to finish point. In practice, the above equation is solved for this total cost $U(x)$ first, and the actual cheapest path is obtained by starting at the finish and integrating a trajectory backward along the gradient field ∇U.

The Eikonal equation is an example of the general static Hamilton-Jacobi equation, and applies in the case of a convex, non-negative isostropic cost function. In order to provide a numerical solution to the above equation, we first lay down a computational mesh in three-dimensional space: this grid need not be uniform in the coordinate directions. The next step is to define an approximation u_{ijk} defined at each mesh point, and approximate the gradient in the above equation is by an upwind approximant of the form:

$$\begin{bmatrix} \max(D^{-x}u_{ijk}, -D^{+x}u_{ijk}, 0)^2 + \\ \max(D^{-y}u_{ijk}, -D^{+y}u_{ijk}, 0)^2 + \\ \max(D^{-z}u_{ijk}, -D^{+z}u_{ijk}, 0)^2 \end{bmatrix}^{1/2} = c_{ijk} \qquad (1)$$

where we have used standard finite difference notation.

Our goal now is to efficiently compute the solution to this equation on a mesh, starting with given boundary values. This equation is solved by using a variant of Fast Marching Methods, which systematically marches the solution outward from the known boundary values one grid point at a time. The key is to exploit the essential idea of Dijkstra's method: the causal nature of Bellman's optimality principle means that the value u_{ijk} that satisfies the above approximation can be constructed using only smaller values of u_{ijk}. Thus, one can start with a single boundary value, solve the above approximation for trial values at each neighbor one mesh point away, and then be assured that the smallest such trial value must be correct. Repeating this process systematically computes the solution at all mesh points in the computational domain: using a heap to extract the smallest such trial values produces the solution in $O(N\log N)$ time, where N is the total number of mesh points in the computational domain.

In more detail, the Fast Marching Method operates 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 1, 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, since it depends only on accepted values which are themselves smaller. This "causality" relationship can be exploited to efficiently and systematically compute the solution as follows:

First, tag points in the initial conditions as *Accepted*. Then tag as *Considered* all points one grid point away and compute values at those points by solving eq 1. The latter points form a boundary (a front) around *Accepted* grid points, which will move outward as procedure continues. Finally, tag as *Far* all other grid points. Then the loop is as follows:

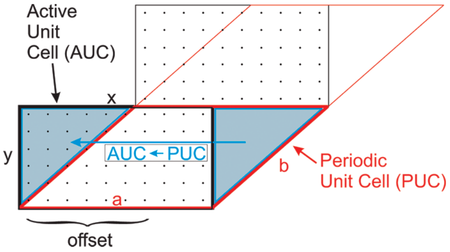

Figure 2. Geometrical relation between periodic unit cell and active unit cell. Faces of neighboring AUCs, unlike faces of neighboring PUCs, do not have to be fully aligned. Instead, they can be shifted by an offset value.

(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*, then 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 equation according to eq 1.

(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 25: the key to an efficient implementation of the above technique lies in a fast heap algorithm to locate the grid point in set of trial values with the smallest value for *u*. These methods have been successfully applied to problems in such topics as robotic navigation, fluid mechanics, and image analysis.[26] For more details on Fast Marching Methods we refer the Readers to ref 26.

In the following three sections, we describe components of Fast Marching Methods that have to be implemented and adopted to allow detection of inaccessible pockets, mainly: discretization of the space, definition of the cost function, and sequence of execution to segment these pockets.

2.2. Discretization of Space. The structure of crystalline porous material is typically stored as atomic coordinates of a repeating fragment of a structure (asymmetric unit), symmetry information and periodic box parameters. This information enables reconstruction of a periodic unit cell (PUC), which may or may not have rectangular shape; however, a rectangular shape is preferred here because of implementation issues.

At the first step, PUC is used to construct the active unit cell (AUC), which will be used to discretize space and build a grid on which Fast Marching Methods will be executed. AUC and the to-be-constructed grid for solving Fast Marching Methods is always rectangular (the cell vectors that define AUC are orthogonal) to facilitate implementation of the Fast Marching Methods procedure and ensure minimal memory overhead compared to nonrectangular PUC embedded inside a rectangular box.

For rectangular PUC, AUC is essentially the same as PUC. For remaining types of PUCs, AUC has to be constructed from PUC by translating sections of the PUC that are outside AUC. Two neighboring AUCs do not necessarily have common faces fully overlapped (see Figure 2). For example, faces aligned in the *x* direction on the *y* level have an offset compared to the $y + 1$ level, which is important in handling periodicity of the chemical system represented by AUC.

The 3D space encapsulated by AUC is then discretized using specified step sizes v_x, v_y, and v_z along each of orthogonal directions (x, y, z, respectively). These sizes have to be small enough to achieve the desired accuracy. Very small steps may, however, result in a very large computational grid. The maximum number of points in a grid is limited by the available resources, in particular the memory size of a computer. Each grid point can be associated with a bin, which surrounds it, and has volume equal to $v_x * v_y * v_z$. Bins are used to the divide system into smaller volumes and sample its properties (e.g., free energy) in each of these volume. Averages from such sampling for each bin are later assigned to the corresponding grid points.

2.3. Constructing the Cost Function. Given a discretization of space, the most straightforward cost function $C(x)$, where *x* is a grid point, is defined as follows: $C = 1$ for each point *x* which can be occupied, $C = \infty$ otherwise. Here, an "occupiable" point corresponds to a position in which the probe can reside. Its exact definition depends on the source data used to represent the investigated chemical system (see the summary in Table 1). For example, if considering atoms as hard spheres (geometry consideration), the occupiable point means a position in which the probe is not colliding with any atom of material's framework: here, a collision occurs when the distance between centers of the probe and any of structure's atoms is smaller than the sum of their specified radii.

When analysis is performed using free energy profiles, the occupiable point means that the average free energy of the probe inside the bin corresponding to the point is equal-to or lower than a specified threshold corresponding to accessible energy levels. Finally, when analyzing histograms from molecular simulations, "occupiable" point means that a position pointed by a point (or associated with the corresponding bin) was visited *t* number of times (for the time *t*) by a molecule during the time of simulation, where *t* is a selected threshold value.

Besides the simple definition of the cost function discussed above, it is also possible to define a continuous cost function. A number of examples of such definitions are presented in Table 2. It must be noted that the definitions collected in Table 2 are not the only ones possible and the cost function can be tuned accordingly to the application. From a practical point of view, the cost function controls the direction and speed at which a front propagates. When considering grids with distance to atomic surface, one may define cost to be inverse proportional to this distance (Table 2). In such definition, the front will propagate faster through centers of cavities rather than areas close to surfaces. Similarly, one may define the cost proportional to the energy experienced by a probe. In such a manner, the front will explore low energy regions sooner than higher energy regions. From a practical point of view, it is worth setting the cost function to infinity for regions which are occupied by the framework atoms and the probability of visiting these regions is practically zero. The latter can be controlled by specifying threshold values (for example r_p, e_t, and t_t in Table 2). From

Table 1. Definition of Cost Function at Point *x* Depending on the Type of Guest-Molecule-Dependent 3D Data Representing a Porous Material

cost	distance-to-surface	potential (or free) energy	histogram
1 (occupiable)	Probe placed at *x* does not overlap with any atom of the framework	The value of averaged potential (or free) energy inside a bin associated with the point *x* is lower than the specified threshold corresponding to accessible potential (or free) energy level at a given temperature.	Bin associated with the point *x* was visited by a probe during the time of molecular simulation *t* number of times (or for time longer than *t*). (The probe can be also defined by any atom of any guest molecule included in the simulation.)
∞ (not-occupiable)	otherwise	otherwise	otherwise

Table 2. Example Definitions of a Continuous Cost Function at Point *x* Depending on the Type of Guest-Molecule-Dependent 3D Data Representing a Porous Material

cost	distance-to-surface	potential (or free) energy	histogram
Value associated with point *x*	The distance *d* from the center of probe with the radii of r_p placed at *x* to the surface of any atom of the framework	The value of averaged potential (or free) energy, *e*, inside a bin associated with the point *x*	Number of times, *t*, the bin associated with the point *x* was visited by a probe during the time of molecular simulation (*t* can also be defined as integrated time a probe was visiting the bin)
Definition of occupiable point	$d > r_p$	$e < e_t$, where e_t is the specified threshold corresponding to accessible potential (or free) energy level at a given temperature	$t > t_t$, where t_t is the specified threshold value
Cost at an occupiable point	$1/(d - r_p)$	e	$1/t$
Cost at a not-occupiable point	∞	∞	∞

the point of view of the work presented here, the threshold defining occupiable/not-occuptiable points is more important than the definition of cost function itself. This is because the threshold to determine the stop criteria for front propagation is directly correlated with the volume to be explored, which is essentially the idea behind this work. In turn, the definition of cost function has a significant effect on the shape of the moving front and in typical cases, like the one to be discussed here, has no effect on accessible volume. However, the ability to tune the cost function may be used to develop other approaches (for example, finding the shortest/lower energy diffusion paths), and is therefore highlighted here.

2.4. Segmentation of Topological Features. Having defined the value of the cost function at each point in the domain, we execute the Fast Marching Method to detect topological features on the void space. There are only two types of features that can be associated with occupiable points (ones with cost lower than infinity): channels and inaccessible pockets (Figure 3A). Channels are defined as features manifested on the face of a periodic unit cell that provide a accessible path throughout the unit cell from one face to the opposite face. Our procedure focuses on detection of channels: pockets are then defined as features consisting of occupiable points that are not identified as channels.

A porous material can have a number of channels in each of three principle directions. Detection along each direction is performed sequentially and consists of two steps illustrated in Figure 3B. In the first step, the periodic boundary conditions along the investigated direction are turned off and then all occupiable points on one face of the periodic unit cell are accepted. The Fast Marching Method is then executed until it finds occupiable points to accept. The important information gained from this algorithm is the identification of points on the opposite-to-initial face of the unit cell that belong to channels in the considered direction. These points are then used in Step 2 to initiate the front in the subsequent passes of the algorithm. All points accepted in the second execution are points corresponding to channels. The implementation of this procedure is fairly straightforward. However, one should note that in the case of nonrectangular unit cells, faces of PUC may not overlap with faces of AUC used to store the 3D data and therefore one need to identify points in AUC that correspond to faces of the original PUC.

After the algorithm finishes with detection of channels, it proceeds to the detection of inaccessible pockets. Any occupiable point that has not been assigned to a channel corresponds to a pocket. There may be, however, more than one pocket, and therefore the algorithm runs until all such points have been assigned to either channels or pockets. In a loop, the first not-yet-assigned point is taken as a starting point for a Fast Marching Methods procedure, and then the front propagates through the pocket until all occupiable points have been visited and assigned to the pocket. Then, it proceeds to detect the next pocket unless all points have been assigned.

2.5. Inaccessible Pore Blocking. Execution of the previous steps of our approach segments the occupiable space into two types of features: channels and inaccessible pockets, where each of these features is represented as a set of grid points. Typically, molecular simulation packages have the option to exclude particular areas of the system (e.g., inaccessible pockets) by placing "blocking spheres'". Therefore, the final part of our approach is focused on calculation of these blocking spheres from the points corresponding to

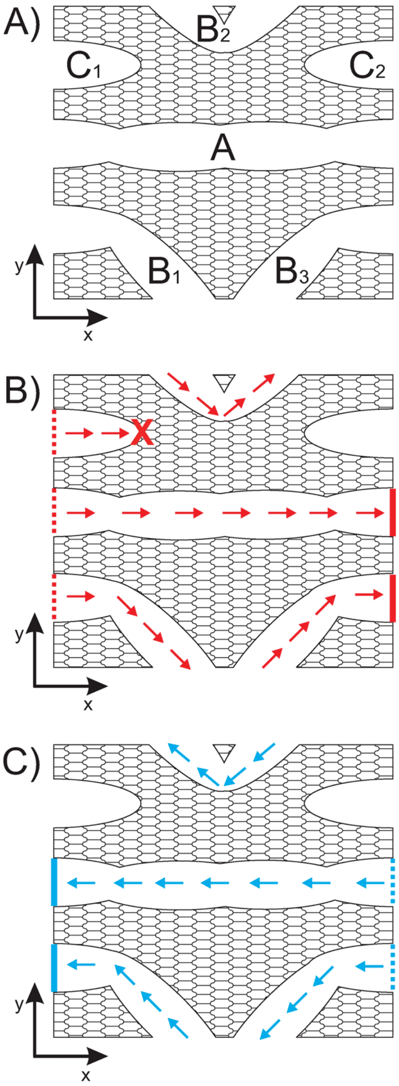

Figure 3. A sequence of front propagations to segment void space. (A) Three features of the void space in a periodic unit cell: straight channel A, topsy turvy channel B build of fragments B_1 − B_3 and inaccessible pocket C build of fragments C_1 and C_2; (B) "Forward" front propagation; (C) "Back" propagation. In B−C, dashed colored lines mark points on a face used to initiate the front, solid lines mark points on the opposite face that were reached by a front and X mark the front that has not reached the opposite face (inside a pocket).

identified features of this kind. It is important that all pockets are fully covered by the blocking spheres but at the same time, channels cannot overlap with any of such spheres.

Our procedure for blocking a pocket with spheres is executed in a loop over all not-yet-blocked pockets. Each pocket has an assigned number of grid points *n*. The procedure highlighted in Figure 4 is as follows:

(1) For each grid point assigned to the inaccessible pocket in consideration, *n*, calculate its density given by a Gaussian filter: $d_a = \sum_{k=1}^{n}\exp\ ((-(d_{ak})^2)/(<d>))$, where d_{ak} is the distance between points a and k, and $<d>$ is the averaged distance between the points forming a pocket;

(2) Select the most dense point of 1 as pocket center, P_c;

(3) Find the distance from P_c to the farthest point in the pocket, the radius of the pocket, r_p;

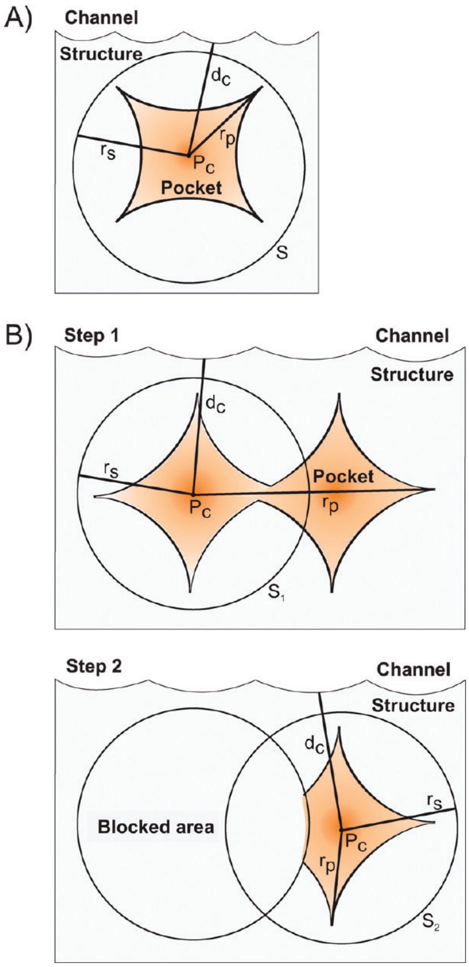

Figure 4. Blocking of inaccessible pockets. (A) Simple case with one blocking sphere. (B) Complex case with iterative blocking (Example with two iterations shown).

(4) Find the distance from P_c to the nearest grid point assigned to any channel, the distance to a channel, d_c;

(5) Check $r_p < d_c$.

- If *true*, then the pocket is blocked with one sphere placed at P_c and with a radius of $(1/2)(r_p + d_c)$;
- If *false*, then the pocket has to be blocked iteratively with a number of spheres. The first sphere is placed at P_c and its radius is set to $d_c - \gamma$, where γ a is small fraction of grid step. All points that belong to the current pocket and that have not been blocked with the newly defined blocking sphere are now defined as a new pocket, which is added to the end of the list of pockets.

The algorithm presented here deserves comment: The advantage of using density calculation in Pt. (1) as a way to detect the center of a pocket over, for example, taking a geometrical center of a pocket is that it always return a point within a pocket, which is desirable in the case of complex-shaped pockets. The obvious drawback is the cost of density calculations, which scales as the square of number of points forming a pocket. For large pockets, we can speed up this calculation by using an approximation to the Gaussian density filter. For example, we can apply an MC sampling of points used in the density calculation of Pt. 1.[28] In our implementation, we use such an approximation for pockets larger than 10000 grid points.

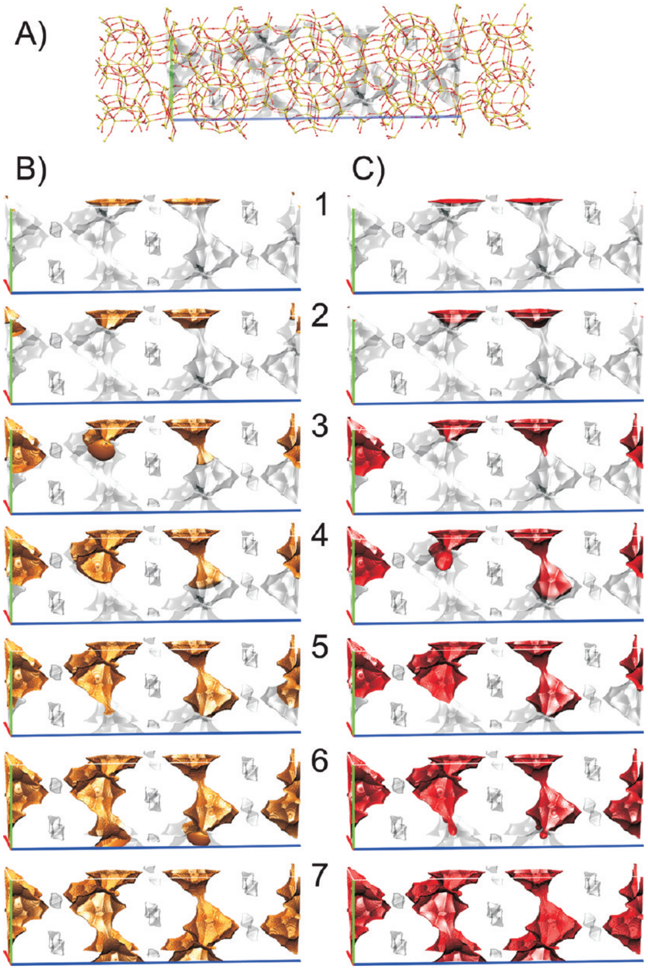

Figure 5. Example of DDR zeolite. (A) 1.5 × 1.5 × 1.5 supercell of DDR zeolite with probe accessible surface in a periodic unit cell. (B) A number of snapshots presenting a propagating front inside a porous material using binary cost function. (C) A number of snapshots presenting a propagating front inside a porous material using continuous cost function.

3. Results and Discussion

We demonstrate an application of our approach to automatic analysis of porous materials using the framework of DDR zeolite as an example. DDR was selected because of its common characteristics. As stored in the IZA database, it has a medium size and nonrectangular periodic unit cell of 6716 Å^3 volume. Figure 5A presents a fragment of the DDR zeolite and highlights an isosurface corresponding to 1.6 Å distance from the atomic surface. This surface corresponds to the boundary value of how close the center of a nitrogen molecule can approach the surface of the zeolite. Our analysis starts with the detection of channels followed by the identification of inaccessible pockets. Figure 5B demonstrates a number of snapshots from the front propagation procedure (a simple binary cost function is used). The front initiated on the face of the unit cell marches its way through the channels. The final snapshot highlights all fragments of space corresponding to channels. It can be noticed that these channels are manifested on two pairs of opposite faces and therefore they form a 2D channel system. The channel detection procedure is repeated for two other directions but, in the case of DDR, no additional channels are detected. The algorithm moves on to detection of inaccessible pockets. Nine pockets are identified. The relatively simple geometry of the material enable the blocking of pockets with nine blocking spheres with radii from 3.6 to 3.9 Å. The presented procedure was repeated for other probe molecules. For example, for a probe of the radius of 1.8 Å, which would roughly correspond to methane, we did not detect any channel in the DDR zeolite. We note that geometry-based considerations based on treating guest molecules and framework atoms as hard spheres may not always be valid. For example, the kinetic energy of guest molecules may allow them to diffuse through narrow passages of higher potential energy leading to a quantitatively different picture of accessible pore topology. Analysis of histograms from molecular simulations do provide much more confidence in this regard.

The time required to perform analysis with our algorithm depends most strongly on the number of grid points that represent the system being investigated. The size of the grid depends on the size of the periodic unit cell and the requested accuracy. In our benchmark case of the DDR zeolite, it took about 5 s to perform analysis of a grid of 0.85 M points. This grid was obtained for steps of 0.2 Å. Similar analysis involving grids with smaller spacing of 0.1 and 0.05 Å required 40 s and 5 min 20 s, respectively. We note that the reported analysis times represent only a small fraction of the time required to generate the to-be-analyzed grids. For example, when using distances to the material's surface, the last two grids required, respectively, 4 and 30 min to compute. These times are orders of magnitude larger when calculating averaged properties such as average free energy in a bin corresponding to each grid point. Nevertheless, the times briefly discussed here were obtained using our prototype implementation, and they possibly can be improved by code optimization.

As mentioned in the Section 2.3, it is also possible to construct a continuous cost function. In comparison to the previously presented binary cost function, such a continuous cost function provides more flexibility on controlling the behavior of the propagating front. Figure 5C illustrates a propagating front in the DDR zeolite with the cost function defined as inverse of the distance to atomic surface (as discussed in Table 2). Comparing Figure 5, parts B and C, side by side exposes a different shape of the front: essentially in the continuous case, the front propagates faster in open areas of the structure (away from the atomic surfaces) creating tongues rather than spherical cups like in the case of binary cost function. The bottom line, however, is that all occupiable points of the grid are explored in both cases, and therefore for the application discussed in this article, binary definition of the cost function is sufficient.

Finally, we demonstrate the effect of selecting a threshold value used to define an occupiable point. In this case, we perform analysis using the free energy grid of IRMOF-10 metal organic framework. The free energy grid of 343 × 343 × 343 size was prepared following the procedure discussed in ref 16 and the used simulation parameters and forcefield followed ref 29. Figure 6 presents the IRMOF-10 structure with two isosurfaces corresponding to accessed

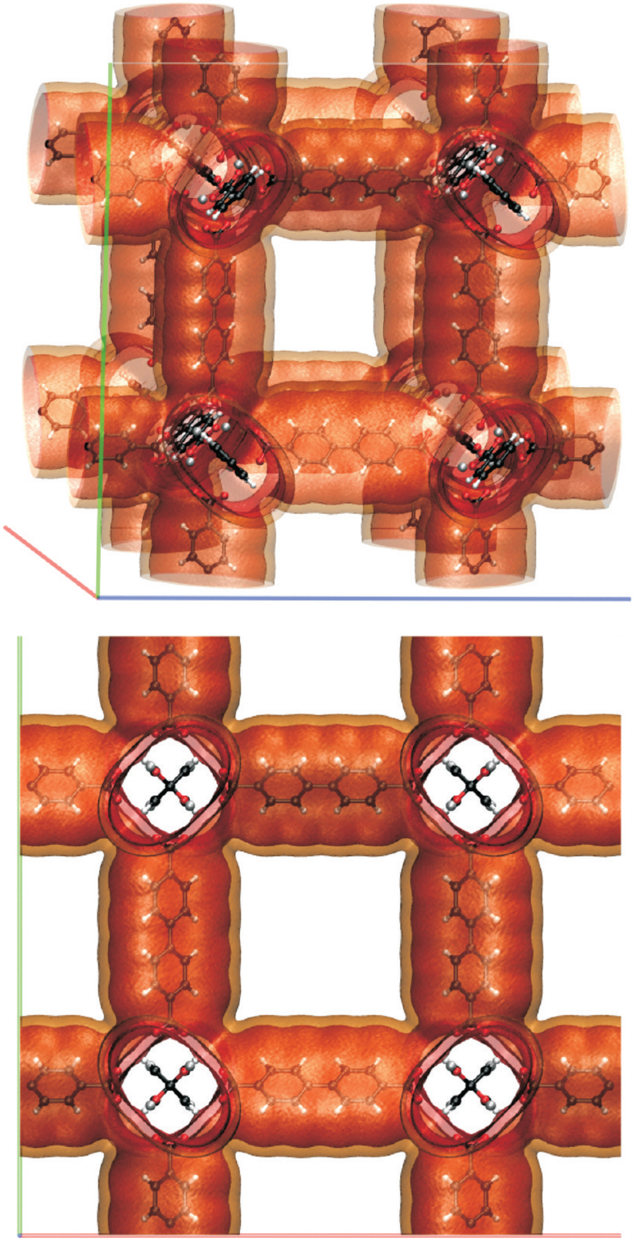

Figure 6. Example of IRMOF-10 metal organic framework. Surfaces highlight accessible volume corresponding to free energy threshold of $10k_BT$ (orange) and $150k_BT$ (red).

volume at free energy level of $10k_BT$ and $150k_BT$. As seen in the Figure 6, the isosurface corresponding to higher energy levels is closer to atoms, and therefore corresponding to larger accessible volume. Continuous cost function allows one not to treat guest molecules as hard spheres but rather allow them to interpenetrate the framework atoms at a higher cost (in our initial formulation we called the accessible interpenetration region "rubber walls"[24]). In general, in case of structures with inaccessible pockets, it may be sometimes possible that setting a higher accessible energy level will lead to exploration of new cavities separated by tight windows with high energy barriers.

4. Conclusions

Inspection of the structure and the void space of a porous material is a critical step in most computational studies involving guest molecules. For example, detection of channels and inaccessible pockets is performed, depending on application, before or after performing a molecular simulation.

We presented an automatic approach to perform such structure inspection. In our approach, we used a partial differential equations-based front propagation technique to segment out features of the void space present in a periodic unit cell of a material. We cast the problem as a path planning problem in 3D space representing a periodic fragment of porous material, and solve the resulting Eikonal equation by using Fast Marching Methods. One attractive feature of this approach is that the to-be-analyzed data can be of varying types, including, for example, a 3D grid representing the distance to the material's surface, the potential or free energy of a molecule inside the material, or even a histogram (a set of snapshots) from a molecular simulation showing areas which were visited by the molecule during the simulation.

The ultimate worth of the algorithm proposed here rests in the ability to perform automatic unsupervised structure analysis and thereby bypass the time-consuming manual analysis. The time savings and reliability are substantial, and further efficiency will come from parallel implementations, leading to an automated high-throughput characterization environment.

We are currently using our approach to characterize all 191 zeolites in the IZA database. Our initial results confirm the need for the automatic structure analysis tools, especially the presented application for detection and blocking of inaccessible pockets in molecular simulations. For example, when considering a probe with radii of 1.4 Å that mimics the He atom, we found out that 22 zeolites have no valid channels, 66 have just channel systems, and 103 (more than 50%!) have both channel systems and inaccessible pockets that have to be blocked in molecular simulations. Similar calculations conducted for the methane molecule ($r = 1.8$ Å) indicated that 60 out of 191 zeolites have no valid channels, 79 have just channel systems, and 52 have both channel systems and inaccessible pockets.

Acknowledgment. M.H. is a 2008 Glenn T. Seaborg Fellow at Lawrence Berkeley National Laboratory. M.H. is supported by the U.S. Department of Energy under Contract DE-AC02-05CH11231. M.H. is also supported jointly by DOE Office of Basic Energy Sciences and the Office of Advanced Scientific Computing Research through SciDAC project CSNEW918 entitled "Knowledge guided screening tools for identification of porous materials for CO_2 separations". J.A.S. is 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. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

(1) Auerbach, S. M.; Carrado, K. A.; Dutta, P. K. *Handbook of Zeolite Science and Technology*; Marcel Dekker: New York, USA, 2004.

(2) Smit, B.; Maesen, T. L. M. *Nature* **2008**, *457*, 671–677.

(3) Smit, B.; Maesen, T. L. M. *Chem. Rev.* **2008**, *108*, 4125–4184.

(4) Krishna, R.; van Baten, J. M. *Chem. Eng. J.* **2007**, *133*, 121–131.

(5) Millward, A. R.; Yaghi, O. M. *J. Am. Chem. Soc.* **2005**, *127*, 17998–17999.

(6) Walton, K. S.; Millward, A. R.; Dubbeldam, D.; Frost, H.; Low, J. J.; Yaghi, O. M.; Snurr, R. Q. *J. Am. Chem. Soc.* **2008**, *130*, 406–407.

(7) Banerjee, R.; Phan, A.; Wang, B.; Knobler, C.; Furukawa, H.; O'Keeffe, M.; Yaghi, O. M. *Science* **2008**, *319*, 939–94.

(8) Sumida, K.; Hill, M. R.; Horike, S.; Dailly, A.; Long, J. R. *J. Am. Chem. Soc.* **2009**, *131*, 15120–15121.

(9) Choi, H. J.; Dinca, M.; Long, J. R. *J. Am. Chem. Soc.* **2008**, *130*, 7848–7850.

(10) (a) Baerlocher, C.; Meier, W. M.; Olson, D. H. *Atlas of Zeolite Framework Types*, 7th ed.; Elsevier: Amsterdam, NL, 2007. (b) http://www.iza-online.org/ (accessed Jan 1, 2010).

(11) Zimmermann, N. E. R.; Haranczyk, M.; Sharma, M.; Liu, B.; Smit, B.; Keil, F. J. *Mol. Sim.*, submitted for publication.

(12) Foster, M. D.; Treacy, M. M. J. http://www.hypotheticalzeolites.net (accessedNov 13, 2009).

(13) Earl, D. J.; Deem, M. W. *Ind. Eng. Chem. Res.* **2006**, *45*, 5449–5454.

(14) Deem, M. W.; Pophale, R.; Cheeseman, P. A.; Earl, D. J. *J. Phys. Chem. C* **2009**, *113*, 21353–21360.

(15) Krishna, R.; van Baten, J. M. *Langmuir* **2010**, *26*, 2975–2978.

(16) Keffer, D.; Gupta, V.; Kim, D.; Lenz, E.; Davis, H. T.; McCormick, A. V. *J. Mol. Graph.* **1996**, *14*, 108–116.

(17) Theisen, K.; Smit, B.; Haranczyk, M. *J. Chem. Inf. Model.* **2010**, *50*, 461–469.

(18) Till, M. S.; Ullmann, G. M. *J. Mol. Model.* **2010**, *16*, 419–429.

(19) Petrek, M.; Otyepka, M.; Banas, P.; Kosinova, P.; Koca, J.; Damborsky, J. *BMC Bioinformatics* **2006**, *7*, 316–325.

(20) Foster, M. D.; Rivin, I.; Treacy, M. M. J.; Friedrichs, O. D. *Micropor. Mesopor. Mater.* **2006**, *90*, 32–38.

(21) Blatov, V. A.; Ilyushin, G. D.; Blatova, O. A.; Anurova, N. A.; Ivanov-Schits, A. K.; Dem'yanets, L. N. *Acta Crystallogr.* **2006**, *B62*, 1010–1018.

(22) Haldoupis, E.; Nair, S.; Sholl, D. S. *J. Am. Chem. Soc.* **2010**, *132*, 7528–7539.

(23) Haranczyk, M.; Rycroft, C. H.; Willems, T. F. *Mol. Inform.*, submitted for publication.

(24) Haranczyk, M.; Sethian, J. A. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 21472–21477.

(25) Sethian, J. A. *Proc. Nat. Acad. Sci.* **1996**, *93*, 1591–1595.

(26) Sethian, J. A. *Level Set Methods and Fast Marching Methods*, 2nd ed.; Cambridge University Press: New York, 1999, 86−99.

(27) Dijkstra, E. W. *Numer. Math.* **1959**, *1*, 269–271.

(28) de Silva, V.; Carlsson, G. *Topological Estimation Using Witness Complexes*; Alexa, M., Rusinkiewicz, S., Eds.; Eurographics Symposium on Point-Based Graphics, 2004.

(29) Liu, B.; Smit, B. *Langmuir* **2009**, *25*, 5918–5926.

CT100433Z