Integrating Physics-based Modeling with PDE Solids for Geometric Design

Haixia Du **Hong Qin**
Department of Computer Science, State University of New York at Stony Brook
Stony Brook, NY 11794 – 4400, USA
{dhaixia, qin}@cs.sunysb.edu

Abstract

PDE techniques, which use Partial Differential Equations (PDEs) to model the shapes of various real-world objects, can unify their geometric attributes and functional constraints in geometric computing and graphics. This paper presents a unified dynamic approach that allows modelers to define the solid geometry of sculptured objects using the second-order or fourth-order elliptic PDEs subject to flexible boundary conditions. Founded upon the previous work on PDE solids by Bloor and Wilson, as well as our recent research on the interactive sculpting of physics-based PDE surfaces, our new formulation and its associated dynamic principle permit designers to directly deform PDE solids whose behaviors are natural and intuitive subject to imposed constraints. Users can easily model and interact with solids of complicated geometry and/or arbitrary topology from locally-defined PDE primitives through trimming operations. We employ the finite-difference discretization and the multi-grid subdivision to solve the PDEs numerically. Our PDE-based modeling software offers users various sculpting toolkits for solid design, allowing them to interactively modify the physical and geometric properties of arbitrary points, curve spans, regions of interest (either in the isoparametric or nonisoparametric form) on boundary surfaces, as well as any interior parts of modeled objects.

Keywords: *PDE Solids, Geometric Modeling, Physics-Based Modeling, Geometric Constraints.*

1. Introduction and Motivation

At present, solid modeling [8, 11] has quickly gained popularity as a convenient and natural paradigm for representing, manipulating and interacting with 3D objects in graphics, animation, and geometric design. This is primarily because a solid model offers engineers an unambiguous shape representation of a physical entity. In essence, the CAD-based solid representation of a real-world physical object is both geometrically unambiguous and topologically consistent. There is a wide array of solid modeling techniques including: Constructive Solid Geometry (CSG), Boundary representation (B-rep), cell decomposition, trivariate parametric *superpatches*, and subdivision solids.

The CSG approach exploits semi-algebraic sets and Boolean operations on the simple primitives such as cubes, spheres, cylinders, cones, and tori to construct more complex solid models. The B-rep technique typically defines a solid object via a set of its boundary surfaces along with extra topological information. The cell decomposition method usually uses 2D cross-sectional slices or cubical units (e.g., voxels) to approximate complicated solids with hierarchically structured octree schemes. Nonetheless, prior solid modeling representation schemes encounter difficulties in interactive sculpting of solid objects, solid geometry deformation, topology modification, and kinematic and dynamic analysis of physical solids.

The parametric solids can be constructed through a function mapping from the trivariate parametric domain of u, v, w to the physical domain of x, y, z. One typical example is free-form solid modeling which combines the benefits of free-form boundary surfaces and interior geometry within a unified framework and provides users more flexible design interface for modeling a much larger variety of objects than aforementioned approaches. It also facilitates cost-effective algorithms for evaluation and manipulation of solid geometry. Typical examples of sculptured solids include trivariate B-splines, Hermite solids and NURBS solids. However, free-form solid techniques are less natural and counter-intuitive, primarily because free-form splines are associated with indirect shape manipulation through time-consuming operations on a large number of control vertices, non-unity weights, and/or non-uniform knots. In general, there are more degrees of freedom than what users can handle. Furthermore, free-form solids are restrained to model regular shapes. It is difficult to extend their geometric coverage to the shape of arbitrary topology without resorting to various non-intuitive geometric constraints.

Subdivision solids generalize the trivariate B-spline solids to free-form solid models of arbitrary topology by applying subdivision rules on 3D control lattice. However, the shape sculpting of subdivision solids is related with indirect manipulation of control lattice and subdivision rules will become complicated for a subdivision model with complex features.

In contrast, PDE solids can effectively model objects through the use of certain elliptic partial differential equations of u, v, w subject to boundary conditions. In comparison with conventional modeling techniques discussed above, PDE approach offers many advantages:

- The behavior of PDE models is governed by differential equations. Solving the PDEs results in both boundary and the interior information simultaneously. In principle, PDE solids can be reconstructed from a small set of heterogeneous boundary/initial conditions, the interior information will be automatically obtained by solving the given PDEs, therefore, fewer parameters are required than that of the free-form or subdivision solids.

- Natural physical processes are frequently characterized by PDEs. Hence, PDE models can be controlled by physical laws, so they are natural and much closer to the real world. PDEs are potentially ideal candidates for both design and analysis tasks.

- The formulation of differential equations is well-conditioned and technically sound. Smooth objects with high-order continuity requirements can be readily defined through a wide spectrum of PDEs.

- Smooth objects that minimize certain energy functionals oftentimes are associated with differential equations, so optimization techniques can be unified with PDE models.

- PDE solids offer the combined advantages of conventional modeling techniques, such as boundary surfaces and underlying parameterization for (generalized) cell decomposition in the interior. The PDE solids have potential to integrate CSG, B-rep and cell decomposition into a single framework.

- PDE solids can unify both geometric and physical aspects for real-world models. They are invaluable throughout the entire modeling, design, analysis, and manufacturing processes. Various heterogeneous requirements can be enforced and satisfied simultaneously.

Although a lot of novel interactive techniques are developed for PDE surfaces in order to realize their full potential, there is a lack of natural interfaces and toolkits for the direct sculpting of PDE solids. Typical modeling difficulties associated with PDE solids include:

- The prior work on PDE techniques primarily concentrates on elliptic PDE surfaces. Interactive techniques for solid modeling are under-explored.

- Besides simple geometric conditions enforced over PDE solid boundaries, there is no formal mechanism for the direct editing of PDE solids anywhere across their domain.

- Traditional elliptic PDE solids are only computed from a set of regular boundary surfaces. More flexible and general boundary constraints are yet to be addressed.

- Conventional PDE techniques are unable to support localized geometric operations for solid models. Global control is less intuitive to manipulate.

- Despite the great potential to integrate different techniques such as CSG, B-rep, cell decomposition, and free-form solids, current PDE techniques only make use of boundary information and many interior properties and features have not yet been considered.

Recently, we [5, 6] proposed an interactive methodology and developed novel modeling techniques that can facilitate the direct and interactive sculpting of physics-based PDE surfaces. These techniques allow PDE surfaces of diverse types of topology to be defined through general, flexible boundary constraints and operations such as trimming, merging, manipulating of isoparametric curves and/or arbitrary curve networks, editing user-specified sub-surface, etc.

To further broaden the accessibility of PDE techniques in geometric modeling and visual computing, we shall forge ahead towards the realization of the full modeling potential associated with dynamic PDE models. In particular, this paper extends PDE techniques for interactive manipulation of physics-based PDE solids, so that both boundary and interior information of PDE solids can be easily edited. The system we develop offers users a powerful solid modeling method with more freedom and flexibility. In our framework, the PDE solids can be defined by boundary surfaces or a set of boundary curve network. We introduce the boundary surface manipulation, local control and trimming operations using simple CSG tools and user-specified datasets on PDE solids to obtain arbitrary topological shapes. The integration of PDE solids with physics-based modeling techniques offers users intuitive editing toolkits for direct sculpting and manipulation of solid models.

2. Prior Work

Bloor and Wilson [1] pioneered a modeling technique—PDE method—that defines a smooth surface as a solution of *elliptic* PDEs. Since its initial application on surface blending, PDE approach has broadened its uses in free-form surface design, solid modeling, and interactive surface sculpting in recent years. In principle, the PDE-based method has the advantage that most of the information defining an object comes from its boundaries. This permits an object to be generated and controlled by very few parameters such as boundary-value conditions and global coefficients associated with an elliptic PDE. This PDE technique can be used to generate piecewise free-form surfaces [2]. Lowe, Bloor and Wilson [9] presented a method with which certain engineering design criteria such as functional constraints can be incorporated into the geometric design of PDE surfaces.

In 1993, Bloor and Wilson formulated PDE solids in terms of parametric boundary surfaces [3], which allow to model the interior of solid objects with boundary surfaces, and further expand the geometric coverage of PDE methodology. Nonetheless, such method is lack of direct manipulation and intuitive sculpting techniques for solid models.

In contrast, Free-Form Deformation (FFD) offers an alternative method for sculpting solid models [13]. The scheme of FFD for solids involves a mapping from R^3 to R^3 through certain trivariate Bernstein polynomials and can be easily integrated with CSG and B-rep solid modeling systems. However, this technique has difficulties to support direct manipulation on arbitrary parts of a solid object and is lack of physically meaningful operations in general.

Physics-based modeling, in contrast, offers users a means to overcome the drawback of indirect design mechanism associated with traditional geometric modeling techniques. Terzopoulos and Fleischer [15] demonstrated simple interactive sculpting using viscoelastic and plastic models. Celniker and Gossard [4] developed a prototype system for interactive free-form design based on the finite-element optimization of energy functions proposed by Terzopoulos and Fleischer [15]. Metaxas and Terzopoulos [10] proposed an approach for creating dynamic solid models by deforming common solid primitives such as spheres, cylinders, cones, or superquadrics globally and locally. Terzopoulos and Qin [12, 16] formulated a novel model for interactive sculpting of Dynamic NURBS (D-NURBS).

Because the dynamic behavior of physics-based models is also controlled by certain differential equations (e.g., Lagrangian equations of motion), it is possible to unify physics-based modeling methodology with PDE approach. Therefore, physics-based modeling augments (rather than replaces) the existing PDE methodology, offering extra advantages for shape modeling. Furthermore, since the majority of physical phenomena can be characterized by PDEs, it is necessary to bridge the gap between geometric PDE models and physics-based modeling approaches towards the realization of the full potential of PDE techniques.

Our previous work [5, 6] proposed an integrated model which combines the PDE surfaces and the physics-based modeling techniques to offer users direct sculpting capability for the PDE surfaces with generalized boundary constraints and user-specified features. However, simple PDE surfaces fall short in modeling most of the real-world objects where interior geometry and material distribution are required for both synthesis and analysis processes. This paper develops manipulation tools for the effective editing of PDE solids, extending the potential of PDE techniques in geometric design and engineering analysis of solids.

3. PDE Formulation

This section formulates PDE solids, and outlines properties of the unified principles of PDE solids and physics-based modeling.

3.1. Elliptic PDE Solids

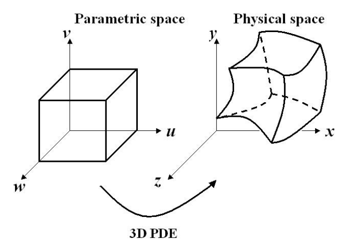


Figure 1. PDE solid from the parametric space to the physical space.

Bloor and Wilson [3] initially employed the second-order elliptic PDE to construct solids:

$$(a^2\frac{\partial^2}{\partial u^2}+b^2\frac{\partial^2}{\partial v^2}+c^2\frac{\partial^2}{\partial w^2})\mathbf{X}(u,v,w)=\mathbf{0}, \qquad (1)$$

where u, v and w are parametric coordinates, a, b and c are smoothing coefficients controlling contributions of partial derivatives along u, v, and w directions to the resulting PDE solids, and $\mathbf{X}(u,v,w) = \begin{bmatrix} x(u,v,w) & y(u,v,w) & z(u,v,w) \end{bmatrix}^\top$ defines the PDE solid coordinates in 3-space. Fig. 1 illustrates a PDE solid through the mapping of the parametric space to the physical space.

Our goal is to directly manipulate PDE solids defined by boundary surfaces. Therefore, direct control of the boundary surfaces is more desirable. To make previously-developed fourth-order elliptic PDE surface sculpting techniques available for solid modeling, it also makes sense to consider the fourth-order elliptic PDE for solids:

$$(a^2 \frac{\partial^2}{\partial u^2} + b^2 \frac{\partial^2}{\partial v^2} + c^2 \frac{\partial^2}{\partial w^2})^2 \mathbf{X}(u,v,w) = \mathbf{0}, \quad (2)$$

where a, b, and c stand for $a(u,v,w)$, $b(u,v,w)$ and $c(u,v,w)$, the coefficient functions of u, v, and w that offer local control for the behavior of PDE solids. We replace the constant smoothing coefficients a, b, and c in previous work using arbitrary functions over u, v, and w, to offer users more flexibility of interactive manipulation. Usually they are set to be constant value over the parametric domain except those regions of interest during the manipulation process. Although our system is focused on this type of elliptic PDEs, our mathematical derivation and its associated numerical techniques can be readily generalized to other PDEs.

To solve (1) and (2), at least six boundary conditions are required in order to derive a unique solution. We restrain u, v, w to vary between 0 and 1, because reparametrization does not change PDE solid geometry if u, v, or w belongs to $[a, b]$. The six boundary PDE surfaces define three surface pairs on the solid boundaries at $u = 0, u = 1, v = 0, v = 1$, $w = 0$, and $w = 1$. They are of the following forms:

$$\begin{array}{l} \mathbf{X}(0,v,w) = \mathbf{U}_0(v,w), \mathbf{X}(1,v,w) = \mathbf{U}_1(v,w), \\ \mathbf{X}(u,0,w) = \mathbf{V}_0(u,w), \mathbf{X}(u,1,w) = \mathbf{V}_1(u,w), \\ \mathbf{X}(u,v,0) = \mathbf{W}_0(u,v), \mathbf{X}(u,v,1) = \mathbf{W}_1(u,v), \end{array} \quad (3)$$

where these six surfaces may share corresponding boundary curves with each other, and they are all open surfaces along their boundaries. Furthermore, because a PDE surface can be derived from a set of Coons-like or Gordon-like boundary curves, boundary conditions in the form of arbitrary curve network are also possible to uniquely define PDE solids. This type of general and arbitrary boundary conditions provide users more flexible tools to model solid objects with fewer parameters, and are capable of modeling solids that must pass through a set of curves that serve as general constraints.

3.2. Numerical Simulation

We resort to the numerical techniques such as finite-difference approximation and iterative method for linear equations to solve the PDE solids for guaranteed solution especially when additional constraints are enforced. Numerical algorithms also facilitate the material modeling of anisotropic distribution and its realistic physical simulation. Among many mature techniques, we employ finite-difference approach in our framework with multi-grid like subdivision method to improve the system performance.

The finite-difference method divides the continuous parametric domain of a PDE into discrete grids and approximates all partial derivatives of the sampling points on the grids by the differences among their neighbors to transform a continuous PDE to an algebraic equation system. The system of algebraic equations can then be solved numerically either through a direct procedure or an iterative process to obtain an approximate solution of the continuous PDE.

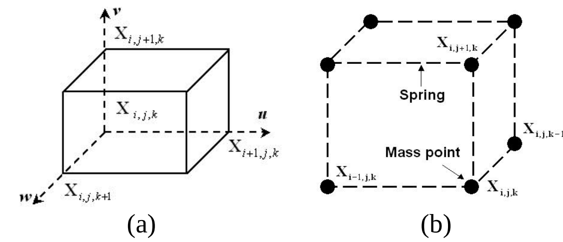


Figure 2. (a) The point discretization of part of a PDE solid; (b) mass-spring network in the vicinity of the sample point $\mathbf{X}_{i,j,k}$.

We use central-difference approximation to approximate the partial derivatives on trivariate solid geometry, by dividing the $[u, v, w]$ domain into l, m, and n discretized points (Fig. 2 (a)), respectively. Now, (1) and (2) can be rewritten in the form as:

$$\mathbf{A}\mathbf{X} = \mathbf{b}, \quad (4)$$

where $\mathbf{A}$ is a discretized differential operator in $(lmn) \times (lmn)$ matrix form. $\mathbf{A}$ is also controlled by the coefficient functions $a(u,v,w)$, $b(u,v,w)$, and $c(u,v,w)$, and

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{X}_{1,1,1} & \mathbf{X}_{1,1,2} & \cdots & \mathbf{X}_{l,m,n} \end{array} \right]^{\top},$$

$$\mathbf{b} = \left[\begin{array}{cccc} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_{l \times m \times n} \end{array} \right]^{\top}.$$

Our PDE solid is open along all of u, v, and w directions, so the computation of partial derivatives near to the six boundary surfaces requires forward/backward differences approximation. Boundary constraints determine all the point coordinates lying on the pre-defined boundary surfaces (e.g., $\mathbf{X}_{1,i,j}$ and $\mathbf{X}_{l,i,j}$, where $1 \le i \le m, 1 \le j \le n$). Arbitrary boundary conditions can be easily enforced using finite-difference method. Note that, in spite of certain combinations of constraint imposition shown in our experiments, in general this type of elliptic PDEs allows the boundary surfaces to be explicitly formulated in arbitrary form. This permits designers to choose (various) constraints based on diverse design tasks.

3.3. Physics-based Modeling of PDE Solids

As a generalization of elastic deformable surface models, a deformable solid is characterized by its positions $\mathbf{X}(u,v,w,t)$, velocities $\dot{\mathbf{X}}(u,v,w,t)$ (which stands for $\frac{\partial \mathbf{X}(u,v,w,t)}{\partial t}$), and accelerations $\ddot{\mathbf{X}}(u,v,w,t)$ (i.e., $\frac{\partial^2 \mathbf{X}(u,v,w,t)}{\partial t^2}$) along with material properties such as mass, damping, and stiffness distributions. These quantities are defined as functions $\mu(u,v,w)$, $\gamma(u,v,w)$, and $\rho(u,v,w)$, respectively, which often can be considered to be constant at certain time. However, these material distributions are allowed to be modified by users interactively and directly over the solid domain. In general, a continuous dynamic solid can be discretized into a collection of mass-points connected by a network of springs across nearest neighbors (and/or along both diagonals). Other springs can be incorporated into the discretized solid if certain types of dynamic behavior are more desirable. We use a mass-spring model because of its simplicity and efficiency.

Applying Lagrangian mechanics, we obtain a set of second-order differential equations that govern the physical behavior of the underlying physics-based model:

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{D}\dot{\mathbf{X}} + \mathbf{K}\mathbf{X} = \mathbf{f}, \tag{5}$$

where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are the mass, damping, and stiffness matrix, respectively. The force at every mass-point in the mesh is the sum of all possible external forces: $\mathbf{f} = \sum \mathbf{f}_{ext}$. The internal forces are generated by springs, where each spring has force $\mathbf{f}_{int} = k(\mathbf{l} - \mathbf{l}_0)$ according to Hook's law. The rest length of each spring is assigned during the PDE initialization and can be modified interactively.

We associate the Lagrangian mechanics with the discretized PDE (refer to (4)) for the unified framework by attaching mass points to geometric grid and adding springs between immediate neighbors on the PDE discretization along u, v, and w, as shown in Fig. 2 (b), then we obtain a dynamic version of PDE solids:

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{D}\dot{\mathbf{X}} + (\mathbf{K} + \mathbf{A})\mathbf{X} = \mathbf{b} + \mathbf{f}, \tag{6}$$

where both the velocity and the acceleration of $\mathbf{X}$ can be discretized along time axis analogously:

$$\ddot{\mathbf{X}} \approx \frac{(\mathbf{X}^{t+\Delta t} - 2\mathbf{X}^{t} + \mathbf{X}^{t-\Delta t})}{\Delta t^2}, \dot{\mathbf{X}} \approx \frac{(\mathbf{X}^{t+\Delta t} - \mathbf{X}^{t})}{\Delta t}.$$

The behavior of the dynamic PDE solids are controlled by both of the Lagrangian equation of motion and the given PDE with boundary and additional geometric constraints. The movement of the sample points of the integrated mass-spring PDE solid model under sculpting is decided by the PDE with constraints, the mass, damping distributions of the points, as well as the stiffness of the springs connecting those points. During the sculpting session of boundary surfaces, we use our physics-based PDE surface model to achieve interactive surface manipulation. Any deformation of the boundary surfaces will propagate to the interior of the PDE solids accordingly. At the equilibrium, if stiffness distributions as well as the external force $\mathbf{f}$ are zero, (6) reduces to (4) with additional physical properties. By allowing a PDE solid to dynamically deform, users will have a natural feeling when interactively manipulating the PDE solid, which is lacking without Lagrangian equations of motion. Furthermore, material properties can be introduced to govern the behavior of the underlying PDE solid. This *hybrid* formulation permits users to obtain a solid that satisfies both geometric criteria and functional requirements at the same time.

3.4. Iterative Method

We use iteration-based techniques to solve (4) and (6). Certain variants of iterative techniques exist for solving the above linear equations [14]. We solve them using Gauss-Seidel iteration. To further speed up the convergence rate of Gauss-Seidel iteration, we take into account the error factor that is characterized by the difference between the approximation and the real value. This leads to the method of Successive Over-Relaxation iteration, or SOR iteration.

However, the discretization of PDE solids results in a much larger number of linear equations than in the surface case. This causes the slow convergence rate of iterative methods. To achieve a faster solution, we take advantage of the multi-grid like subdivision method to speed up the numerical integration. It first solves the PDE solid at the coarsest resolution, and refines the solution for finer grids. The convergence rate of iteration can be greatly increased. It also allows users to control the error bound of the approximated solution.

4. PDE Sculpting Toolkits

This section details various interactive techniques for PDE solid sculpting.

4.1. Solid Initialization

Our system supports two types of initialization for the PDE solid: initial boundary surfaces and initial boundary curves. At the start of the initialization phase, the user must specify the boundary type, i.e., whether the boundary conditions are given as pre-defined surfaces, or connected boundary curve network for the PDE solid.

For initialization with pre-defined boundary surfaces, the system can obtain the already defined surfaces that are pieced together and form the outline of PDE solid from file or use the previous techniques to generate PDE surfaces. Then using the surfaces as boundary conditions, we can obtain a PDE solid bounded by these surfaces as the solution of (4). Fig. 4 shows an example.

If users decide to employ the curve network as boundary conditions, there will be at least 12 curves required to define the Coons-patch like boundary conditions for the six boundary surfaces. There are two steps in this case: (1) derive the boundary surfaces from the boundary curves users specified; (2) solve (4) to obtain the corresponding PDE solid. We use the Coons-like boundary conditions for the boundary curve network because every two neighboring surfaces share one boundary curve. To make sure the solved PDE surfaces satisfy such conditions, the shared boundary curves need to be defined. We can even define the boundary surfaces more precisely by adding more curves as boundary conditions, which leads to the Gordon-like boundary conditions of the boundary surfaces. Fig. 5 shows examples of curve network as boundary conditions.

The coefficient functions $a(u, v, w)$, $b(u, v, w)$, and $c(u, v, w)$ can also influence the solution of the PDE solid. These three functions control the relative blending and the level of variable dependence among u, v, and w directions. Consequently, users can control how boundary conditions influence the interior of a solid by modifying the length scale at arbitrary location (i.e. $a_{i,j,k}$, $b_{i,j,k}$, and $c_{i,j,k}$). In general, users can define the coefficient functions $a(u, v, w)$, $b(u, v, w)$, and $c(u, v, w)$ interactively over $\mathbf{X}$.

4.2. PDE Solid Subdivision

The large number of sample points of a PDE surface/solid results in the slow convergence of iterative techniques. We develop a multi-grid like approximation based on simple subdivision schemes to improve the computation performance. Since there are two types of boundary conditions, i.e., curve network and surfaces, we propose different subdivision schemes to handle two types of boundary constraints, respectively.

4.2.1 Curve Subdivision

If boundary conditions come from curves, we shall first compute boundary surfaces. We start with a small number of sample points at the coarsest scale of the PDE boundary surfaces, then the approximated solution of the PDE surface can be easily derived after several iterations. Then, the PDE solid at the coarsest scale is solved. Users can refine the coarse mesh through subdivision and use the new subdivided mesh as an initial guess for subsequent iteration steps. The finer grid is then computed iteratively to achieve a more accurate and smoother solution of the boundary PDE surfaces as well as the PDE solid. For further refinement over the finest grid, the multi-grid subdivision starts with the up-sampling of all boundary curves through the use of four-point interpolatory subdivision scheme [7] in order to guarantee the smoothness requirement of the refined curves.

4.2.2 Surface Subdivision

If boundary conditions come from connected surfaces, the subdivision scheme should be slightly modified. We start with the coarsest resolution of the boundary surfaces through down-sampling to obtain a coarse solution of the solid. Then during the refining process, we sample more points over boundary surfaces until it reaches the finest resolution. After that, the subdivision process may continue to reach even finer resolution. In this scenario, we consider the given boundary surfaces as constrained PDE surfaces, requiring four curves as the boundary conditions and the originally defined surface sample points as hard constraints. We then use the four-point interpolatory subdivision scheme to subdivide the boundary curves and compute unknown surface points by solving the surface PDE subject to the subdivided boundary curves and original surface points as hard constraints.

4.3. Boundary Manipulation

Users can modify the global shape of a PDE solid through boundary manipulation. Our system permits users to directly modify the boundary PDE surfaces that define the PDE solid. To modify a PDE solid through boundary conditions, users must select a boundary surface for the editing purpose, then use the sculpting toolkits provided by our system to modify the selected surface.

Typical interactive toolkits for the direct sculpting of PDE surfaces include:

- **Point Editing:** PDE surfaces can be interactively sculpted by enforcing additional constraints on a set of selected points as well as their normal and curvature. Users can modify a PDE surface by selecting a set of points on the surface grid, then dragging them to the desired position where the surface must interpolate. Users can also manipulate the surface normal and the curvature along parametric directions at any point to achieve a local editing capability in the vicinity of the data point.

- **Curve Constraints:** We further provide editing tools that afford the intuitive specification of curve-based constraints. Users can select an arbitrary source curve on the PDE surface, then specify a cubic B-spline curve as the destination curve which then is mapped to the selected surface curve, and the PDE surface will be modified accordingly.

- **Area Manipulation:** Analogous to the curve tool, our system can map a user-specified B-spline destination patch onto a region of interest over the PDE surface. By selecting an area on the PDE surface and defining a B-spline patch sampled with the same number of grid points as those in the source region, our system maps the B-spline patch to the specified area as additional constraints, and the modified PDE surface will satisfy the mapping constraint.

According to the interactive modification of the selected boundary surface, the PDE solid will be deformed accordingly. Fig. 6 shows two examples of boundary manipulation with curve constraints. The operations of the boundary PDE surfaces provide a means for the direct manipulation of the PDE solid.

4.4. Direct Solid Manipulation

Solid sculpting by way of boundary manipulation is far from adequate. One attractive advantage of PDE-based solid modeling is that the solid interior is also controlled by PDEs without the need of time-consuming specification on interior material and its distribution. PDE solids provide an integrated scheme that not only expands the B-rep method to cover the interior information but also supports Boolean operations associated with CSG models. More importantly, users can deform the interior of a PDE solid by enforcing additional hard constraints inside the solid without changing the boundary conditions. Additional constraints inside the solid introduce a set of new equations into the system to replace the corresponding original equations. Consequently, (4) becomes

$$\mathbf{A}_c \mathbf{X} = \mathbf{b}_c, \tag{7}$$

where $\mathbf{A}_c$ and $\mathbf{b}_c$ are obtained by replacing k equations in the original system with new ones resulted from the constraints with $k > 0$. The modified equation system can be solved using the aforementioned techniques. The interactive operations inside a PDE solid include local region sculpting and solid trimming.

4.4.1 Region Manipulation

Traditional PDE solids only support boundary manipulation which leads to global deformation of the entire solids. It is more desirable to offer users editing functionalities on the interior properties with interactive interface. We develop a set of toolkits that allow designers to specify any interior region of a solid, and only enforce local deformation in the selected region. Alternatively, we can freeze the selected region and disallow any changes in the specified region. In our system, this can be done through: (1) interactively specifying a region in $[u, v, w]$ domain, (2) employing some basic CSG-based tools such as spheres and cubes to navigate the entire parametric domain to define the region of interest, or (3) embedding datasets within the PDE solid in order to define the particular region. Subsequently, any changes within the region will have no effect on points outside the region. The localized deformation can be achieved easily because only those equations corresponding to the points of the specified regions in (4) will be solved. In principle, all hard constraints can be viewed as some sorts of local deformation. Fig. 7 shows examples of local deformation.

4.4.2 Solid Trimming

One of the disadvantages of parametric solids is that it is difficult to model objects of arbitrary topology. Trimming operation offers an alternative way to model objects with irregular shape. Our system offers users trimming functionalities on a PDE solid for the sculpting purpose. After the region of interest is selected, users can remove material from the PDE solid either inside or outside the specified region. Multiple selected regions are also supported in our system, permitting the trimming on multiple regions simultaneously. Furthermore, we can use the idea of CSG models to place trimming tools of the simple shape primitives such as sphere, cube, or cylinder, at any position inside the parametric domain, then move the shape along the u, v, or w directions, all the regions covered by its navigating path will be chosen/discarded according to the specified Boolean operations. This type of tools allows the CSG construction of complex objects based on PDE solids. Another trimming operation comes from the region-fixing method introduced in above section, i.e., we allow users to embed datasets into PDE parametric domain and map them to the physical space to obtained the desired shapes. The mapping of the datasets to different PDE solids will result in different shapes. In essence, this is analogous to the principle of free-form deformation. The trimming operations on PDE solids can greatly expand the coverage of PDE solid applications, making it possible to obtain a PDE solid with complex boundaries and arbitrary topological types. Fig. 8 shows several trimming examples.

4.5. PDE Solid Dynamics

Because the run time of standard numerical solvers depends on the number of sampling points on a PDE solid, which is much larger than the number of sample points of PDE surfaces, it generally takes minutes to obtain the final stable shape of PDE solids due to the large number of equations to be solved. When the number of sample points is extremely large, the computation is time-consuming. This is less attractive from the standpoint of interactive sculpting as continuous visual feedback between consecutive states is strongly desirable. To ameliorate it, we consider the integrated mass-spring model of PDE solids whose dynamic behavior is governed by (6). The external force $\mathbf{f}$ can be computed implicitly based on various constraints placed on the boundary and the interior of the solid which may change the positions of selected sample points or regions and cause deformation of the solid according to the governing PDE formulation. By dividing the time domain into small time steps and approximating both velocities and accelerations of data points through successive time intervals with finite-difference method, we can dynamically manipulate the PDE solid with forces in real-time by solving

$$(\mathbf{M} + \Delta t\mathbf{D} + \Delta t^2\mathbf{K} + \Delta t^2\mathbf{A}_c)\mathbf{X}^{t+\Delta t} = \\ \Delta t^2(\mathbf{b}_c + \mathbf{f}) + (2\mathbf{M} + \Delta t\mathbf{D})\mathbf{X}^t - \mathbf{M}\mathbf{X}^{t-\Delta t}. \quad (8)$$

The material properties such as mass/damping quantities and stiffness distributions can be modified as additional constraints to control the behavior of the PDE solid. Because in our mass-spring model, the solid points gradually evolve along consecutive time steps when enforcing additional constraints, the number of iterations to solve (8) is very small, which greatly improves the time performance of the direct sculpting of PDE solids.

5. Implementation and Results

This section outlines the functional components of our system and presents our experimental results.

5.1. System Architecture

We have expanded our prototype software environment that permits users to interactively manipulate PDE surfaces with various local/global constraints to PDE solids, which allows the interactive sculpting of PDE solids via boundary conditions and interior operations. Our system can model both PDE surfaces and solids with various manipulation toolkits. The system is written in Visual C++ and runs on Windows95/98/NT/2000. Fig. 3 illustrates the architecture of our modeling environment for PDE objects. In particular, our system provides the following functionalities:

Boundary Conditions. Users can interactively input and edit boundary surfaces or boundary curves, and obtain PDE solids satisfying these conditions. Moreover, the system offers a multi-grid like subdivision scheme to improve time performance of the iterative solver.

Dynamic Models. Our system supports novel physics-based PDE solids including: (1) finite-difference discretization using mass-spring models; (2) multi-grid like subdivision for model refinement and performance speedup. Material properties and dynamic behavior greatly enhance the interactive manipulation of conventional PDE solids.

Manipulating Boundary Curves and Surfaces. Users can use various manipulation routines to deform the boundary surfaces including: (1) editing points and their normal and curvature at arbitrary locations; (2) enforcing a set of curve constraints; (3) deforming a set of user-specified regions; and (4) applying local operations only on a user-selected area.

Interior Operations. In addition, users can also work directly on interior of the PDE solids through: (1) interior deformation with additional constraints inside the solid; (2) trimming the specified region for complex geometry and arbitrary topology; and (3) modifying coefficient functions as well as material properties such as mass, damping, and stiffness distributions locally.

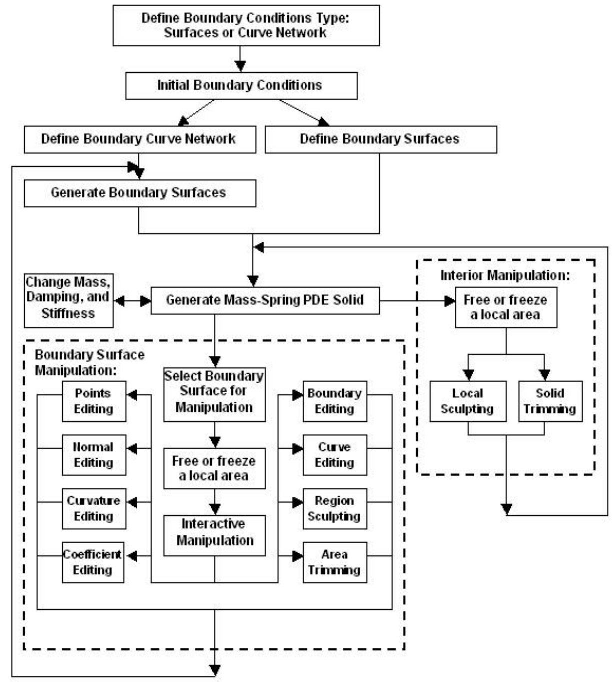


Figure 3. System architecture and functionalities.

5.2. Results and Discussion

We use two iterative techniques (Gauss-Seidel and SOR) together with multi-grid like subdivision techniques to solve the PDE solid subject to various constraints. Table 1 details our experiments of PDE solid examples obtained by boundary curves (Cube) and boundary surfaces (Sphere) and their performance.

Model	G-S	SOR
Cube-4	387.848	367.349
Cube-2	37.333	32.142
Cube-s	141.396	120.536
Sphere-4	304.029	457.790
Sphere-2	19.989	16.404
Sphere-s	31.436	28.626

Table 1. CPU time (in seconds) for different solvers. G-S is Gauss-Seidel iteration, SOR is SOR iteration, σ=1.25. The "-4", "-2" and "-s" stand for the 4th, 2nd order PDE, and the 4th order PDE with subdivision, respectively.

Besides traditional boundary conditions of PDE techniques, our system allows users to specify and enforce a large variety of additional constraints on a set of points, cross-sectional curves, and surface areas on the boundary surfaces. These constraints provide more freedom to designers, making the design process of PDE solids more cost-effective. The curve-based boundary conditions make it even easier for designers to achieve the desired shape of the PDE solid. We can also enforce additional constraints directly inside the PDE solid and apply the trimming operations, which facilitate the construction of PDE solids of arbitrary topology. We develop our prototype system using finite-difference techniques because they are simple, easy to implement, and suitable for the incorporation of complicated, flexible constraints. In general, the time and space complexity is increased with higher resolution as well as increased accuracy. Our multi-grid like subdivision method for various levels of refinement achieves anticipated results in our experiments. The examples shown in this paper are rendered using POV-Ray and the trimming datasets are provided by 3DCAFE.com.

6. Conclusion

We have developed a set of interactive algorithms that support both global and local deformation of PDE solids subject to various constraints. We proposed a unified methodology that marries PDE solids with PDE surfaces and physics-based techniques. The PDE solids can be defined by either boundary surfaces or a set of curves as generalized boundary conditions. This technique offers users more freedom and a more natural interface to manipulate PDE solids satisfying a set of design criteria and functional requirements. The interactive editing of PDE surfaces as boundary conditions of PDE solids provides powerful sculpting toolkits for solid modeling. Physics-based modeling permits the dynamic behavior of PDE solids to be governed by physical laws, making PDE solids more realistic and more interactive than the traditional solid modeling techniques. Our software environment provides users a wide range of powerful manipulation toolkits for the boundary surfaces, including point-based manipulation, cross-sectional design and the manipulation of non-isoparametric curves, as well as region deformation. These enhancements permit users to model and edit PDE surfaces and corresponding solids intuitively with ease. The deformation and trimming operations inside the solid provide a way to model objects of arbitrary topology using PDE techniques. Our unified approach and novel PDE techniques greatly expand the geometric coverage and the topological flexibility of conventional PDE solids, improve the utility of PDE solids for modeling and design applications, as well as help the realization of the full potential of PDE technology in visual computing fields.

Acknowledgments

This research was supported in part by the NSF CAREER award CCR-9896123, the NSF grant DMI-9896170, the NSF ITR grant IIS-0082035, and a research grant from Ford Motor Company.

References

[1] M. I. G. Bloor and M. J. Wilson. Generating blend surfaces using partial differential equations. *Computer Aided Design*, 21(3):165–171, 1989.

[2] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free-form surfaces. *Computer Aided Design*, 22(4):202–212, 1990.

[3] M. I. G. Bloor and M. J. Wilson. Functionality in solids obtained from partial differential equations. *Computing Suppl. 8*, pages 21–42, 1993.

[4] G. Celniker and D. Gossard. Deformable curve and surface finite elements for free-form shape design. *Computer Graphics*, 25(4):165–170, 1991.

[5] H. Du and H. Qin. Direct manipulation and interactive sculpting of PDE surfaces. *Computer Graphics Forum*, 19(3):C261–C270, 2000.

[6] H. Du and H. Qin. Dynamic PDE surfaces with flexible and general constraints. In *Proceedings of the Eighth Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2000)*, pages 213–222, Hong Kong, 2000.

[7] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transaction on Graphics*, 9(2):160–169, 1990.

[8] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers, Inc., 1989.

[9] T. W. Lowe, M. I. G. Bloor, and M. J. Wilson. Functionality in blend design. *Computer Aided Design*, 22(10):655–665, 1990.

[10] D. Metaxas and D. Terzopoulos. Dynamic deformation of solid primitives with constraints. In *SIGGRAPH 92*, pages 309–312, Chicago, IL, USA, 1992.

[11] M. Mortenson. *Geometric Modeling, Second Edition*. Wiley Computer Publishing, 1997.

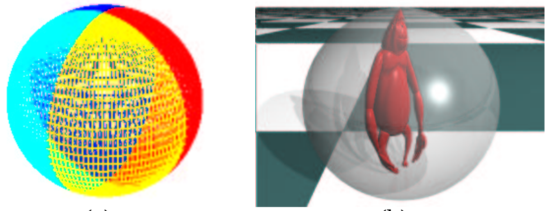
[12] H. Qin and D. Terzopoulos. D-NURBS: A physics-based framework for geometric design. *IEEE Transaction on Visualization and Computer Graphics*, 2(1):85–96, 1996.

[13] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH 86*, pages 151–158, Dallas, TX, USA, 1986.

[14] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.

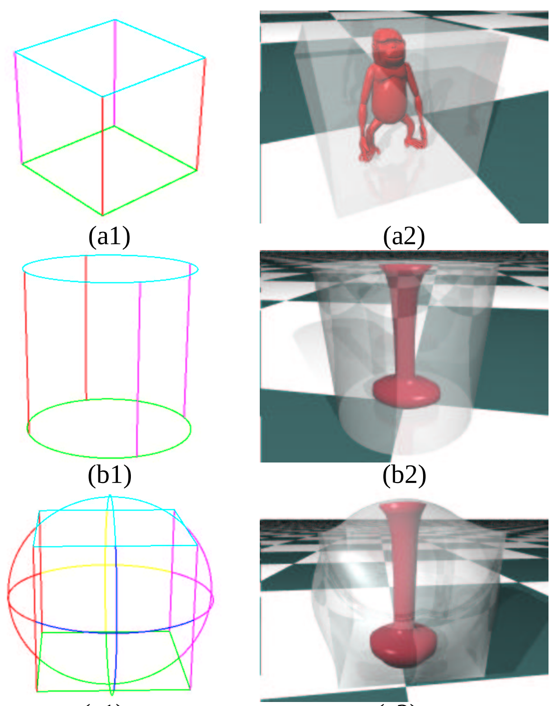
[15] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4(6):306–331, 1988.

[16] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transaction on Graphics*, 13(2):103–136, 1994.



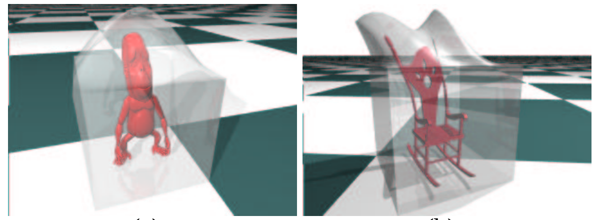
(a) (b)

Figure 4. A PDE solid generated from given boundary surfaces: (a) boundary conditions, (b) the solid (displayed using transparent material) subject to (a) with a trimming dataset inside the solid.



(a1) (a2) (b1) (b2) (c1) (c2)

Figure 5. Examples of PDE solids subject to boundary curve network: (a1) and (b1) are two sets of Coons-like boundary curves; (a2) and (b2) are corresponding PDE solids; (c1) are Gordon-like boundary conditions, and (c2) is the PDE solid subject to (c1). The PDE solids are displayed using transparent material with trimming datasets.

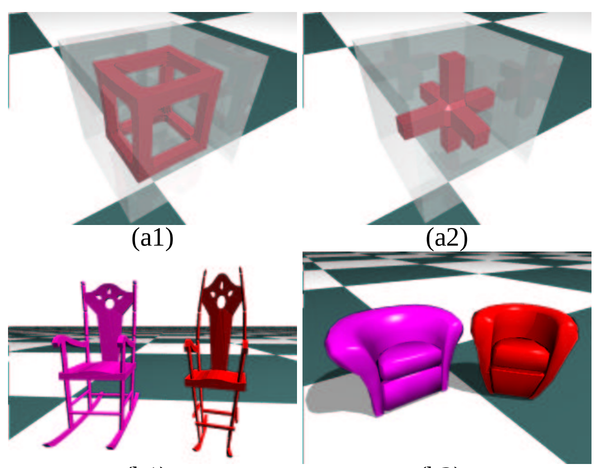


(a) (b)

Figure 6. Modifying PDE solids via curve constraints of boundary surfaces.

(a) (b)

Figure 7. Direct modification of the trimmed PDE solid: (a) Directly moving a selected point on the trimmed data, on the left is the original dataset, and on the right is the modified trimmed PDE solid; (b) the deforming sequence of an trimmed object by rotating selected interior regions.



(a1) (a2) (b1) (b2)

Figure 8. Trimming examples. (a1) and (a2) are using CSG trimming operations in a PDE solid, and the trimmed parts are shown in red covered by transparent original solids. (b1) and (b2) are trimming examples using datasets for different PDE solids. In each of (b1) and (b2), the object on the left is trimmed from a PDE cube, and the object on the right is trimmed from a PDE sphere.