1 Introduction

1.1 Computer Aided Design (CAD)/Computer Aided Engineering (CAE) Integration in a Nutshell. Seamless integration of geometric design and engineering analysis is the holy grail of product life cycle management, also known as CAD/CAE integration. Traditionally, this integration is perceived in terms of a sequential design-analysis cycle, where geometric design is iteratively subjected to analysis, followed by changes in design, and so on. Shape and topology optimization are examples of such an integration where this iterative process is completely automated; this is made possible by making sure a priori that geometric representation of the mechanical domain and the approximate solution of the analysis problem are compatible with each other, so that changes in one are easily translated into desired modifications of the other. But outside of such carefully conceived applications, the goal of complete CAD/CAE integration has proven elusive. In this paper, we examine the underlying difficulties and promising future directions for CAD/CAE integration, assuming widely accepted representations of CAD geometry and analysis.

The basic CAD/CAE integration problem is described schematically in Fig. 1. The CAD model is typically a combination of a high-level parametric feature-based representation and a fully evaluated boundary representation. The exact type of representation is less important, as long as the model supports the standard geometric queries: generating and classifying (in/on/out) points, set membership classification (which subsumes intersection queries), and distance computation. CAD systems also provide (and compete on) a wide range of model editing tools, from high-level parametric updates to free-form deformations of bounding curves and surfaces. It is common to consider material specification as part of the CAD model, but it is usually limited to isotropic or orthotropic materials; as we discuss in Sec. 5, modeling more complex materials may require solving an analysis problem.

CAE modeling loosely refers to a problem of engineering analysis solved on the associated CAD model. To be more precise, CAE modeling involves solving a boundary/initial value problems defined over the geometric domain represented by the CAD model. As such, CAE modeling is represented by several components: the boundary conditions (e.g., restraints and loads) acting on some boundary portions of the CAD model, governing equations (e.g., equilibrium and compatibility) typically in a weak form of integral equations, and constitutive equation implied by the indicated material model. A numerical approximation of the solution (e.g., displacement field) is then computed as a linear combination of some basis functions, using one of many approximation methods. Any such numerical approximation must satisfy the prescribed boundary conditions, and its construction requires differentiation, numerical integration over the support of the basis functions, and solution of (typically) linear system of equations.

All CAD/CAE integration problems may be divided into two broad categories: (1) forward integration, where analysis solutions are produced automatically on demand from a CAD model and (2) inverse integration where CAD model is modified based on CAE results. Generally speaking, the inverse problems are harder and usually require solution of the forward problems. For example, shape optimization is an example of an inverse problem that is usually solved using many forward solution iterations. It is more common to see inverse integration performed manually by a human designer who interprets the results of the analysis and decides which changes to apply to the CAD model. Accordingly, in this paper, we will deal mostly with issues of forward integration. We will avoid direct comparison of generally important numerical properties of various computational methods, such as accuracy, convergence, and stability, except when they directly affect viability and adaption of the proposed solutions to the CAD/CAE integration problem.

1.2 Approach and Outline. We will say that CAD/CAE integration is complete or “seamless” when either or both forward and inverse tasks are fully automated. Partial integration allows some manual steps or is automated only for some restricted subset of tasks or representations. One of the main goals of this paper is to examine the approaches to and the challenges in achieving complete integration. It would be impossible to survey all possible geometric representations and all possible types of engineering analysis. Instead, we will take a more pragmatic approach and focus on individual computational tasks required for any CAD/CAE integration. For concreteness, whenever is appropriate, we will use the problem of structural elastic analysis as a prototype analysis problem, but it should be clear that the integration issues and challenges are generic and apply to all other types of boundary/initial value problems. With this in mind, we identify the following individual tasks required in CAD/CAE integration:
The above problems are building blocks of all CAD/CAE integration activities. Their combinations cover a broad spectrum of engineering applications, including most problems in design and assembly modeling, shape, configuration, and material optimization, as well as manufacturing process simulation. In what follows, we will examine each of these fundamental tasks, discuss known and proposed solutions, and the extent to which they support CAD/CAE integration. We will not attempt to deal with broader issues related to advantages or disadvantages of a particular numerical technique, unless it directly affects the integration issues.

2 Linear Static Analysis

The notion of “design-analysis cycle” refers to a basic concept: the designer is iteratively creating/modifying shape, while the computer is providing feedback by (1) rendering updated geometric information and (2) performing engineering analysis of the shape. This is the basic case of forward CAD/CAE integration that we examine in this section. Formally, the basic integration approximates a well-posed boundary value problem on a geometric domain Ω subject to user-specified boundary conditions. The approximation is constructed on a finite basis chosen from some sufficiently complete space. For concreteness consider linear structural analysis, other elliptic problems are similar.

2.1 CAD/CAE Integration via Discretized Weak Form. Virtually all computational approximations of the analysis problem rely the discretized weak form Ref. [1] of the original boundary value problem. Such weak forms come in essentially two varieties: those which explicitly include boundary conditions and those which do not. In the case of linear elasticity, the first of these can be written in the notation widely accepted in finite element analysis (FEA) literature as [1]

\[- \sum_{i=1}^{n} C_i \int_{\Omega} B^T \eta_i \left[ \eta_j \right] d\Omega = - \int_{\Gamma} \eta_i F d\Gamma - \int_{\Gamma} \eta_i T d\Gamma, \]

where Ω is the geometric domain, Γj is the portion of the boundary where the external loads T are applied, η is vector of basis (shape) functions with unknown coefficients Ci, F are body forces (such as gravity), D is the material tensor, and u* represents the essential boundary conditions (restraints and displacements). In the traditional mesh-based FEA method (as well as many others), the form (1) is used and u* is interpolated by the usual boundary (shape) functions over the nodes with specified essential boundary conditions (discussed in Sec. 2.2). In contrast, if the numerical approximation (2) is used, u* is independent of any discretization and may be interpolated transfinately [2] or fitted using least-squares [3].

The operator B computes the matrix of partial derivatives for a given vector. Once the individual terms above are evaluated, the analysis problem reduces to a system of algebraic equations that must be solved for the coefficients Ci. An approximate solution to the analysis problem (in this case, displacement field u) is constructed as the linear combination

\[ u = \sum_{i=1}^{n} C_i \eta_i + u^* \]

In other words, the basic problem of forward CAD/CAE integration amounts to a fully automated evaluation of the discretized weak form (2), given the CAD model and an appropriate model problem. Specifically, assuming that the CAD model is given by its boundary representation, complete integration must support the following tasks:

1. Specifying boundary conditions by prescribing the values and directions of forces T or indicating constraints on values of displacement u on faces in the boundary representation.
2. Enforcing boundary conditions by constructing basis functions η so that the computed solution satisfies Eq. (3). For example, displacements must be identically zero on all fully restrained faces.
3. Volumetric integration of the first three terms of Eq. (2) over the domain Ω given by its boundary representation.
4. Surface integration of the loads T over the faces in the boundary representation.
5. Solving the linear system of equations (2) once all integral terms are evaluated.
6. Interpolating and visualizing the approximate solution (3) over the boundary representation of Ω.

Critical examination of the above tasks reveals that task (1) is a matter of user interface, task (5) is a generic numerical computation problem, and tasks (4) and (5) are essentially solved problems that rely on tessellating (triangulating) boundary representations. It follows that tasks (2) of enforcing boundary conditions and (3) of volumetric integration hold the key to completing CAD/CAE integration in this basic situation. Surprisingly, these two tasks have proved to be more challenging than expected.
We have not made any assumptions about the basis functions \( \eta_i \), but it is well known that any reasonable choice requires that \( \eta_i \) has a compact support. This in turn transforms all integrals over the domain \( \Omega \) or its boundary \( \Gamma \) into integrals over finite support of the basis functions, leading to more efficient and robust banded linear system. This also implies that any such approximate computation involves the concept of spatial discretization of the space into a collection of supports, or “elements”—whether they are represented explicitly or implied. Below we categorize and discuss the known approaches to spatial discretization, but all of them can be divided into two broad categories: (conforming) mesh-based and (nonconforming) meshfree.

2.2 Conforming Mesh-Based Methods

2.2.1 Classical Finite Element Analysis. The simplest, the oldest, and the most widely accepted spatial discretization is finite element meshing where the boundary representation of domain \( \Omega \) is converted into a network of finite elements [1]. The union of all elements represents \( \Omega \) approximately by approximately conforming to its boundary \( \Gamma \).

The finite elements serve as supports for the basis functions \( \eta_i \) that are associated with the element nodes. If a basis function \( \eta_i \) is chosen so that it’s value at node \( i \) is 1 and 0 at \( j \neq i \) then coefficient \( C_i \) is the value of the solution (displacement) to the boundary value problem. If the boundary condition \( u^* \) is specified over a face \( \Gamma_k \) in the boundary representation of the CAD model, it can be enforced at the boundary nodes of the mesh by simply evaluating \( u^* \) at these nodes and subtracting from the assembled system equations terms associated with the corresponding force due to these enforced displacements (as well as some other minor adjustments—sometimes referred to as the “zero-one rule” [4]). On all other points of the boundary, \( u^* \) is interpolated from these nodes and is enforced only approximately. Furthermore, since the basis functions are known a priori, they can be differentiated and integrated easily over the individual elements using Gauss quadrature rules. In other words, representing the CAD model as a conforming mesh solves the two remaining difficulties of CAD/CAE integration: enforcing boundary conditions and volumetric integration.

The above observations, as well as the successes and advantages of mesh-based approaches (and FEA in particular) are well documented in the literature [1]. Here, we are specifically interested in how they support CAD/CAE integration, as proposed by many researchers and advocated by many vendors. It should be clear that all integration tasks depend on the ability to automatically convert a boundary representation of the CAD model into a finite element mesh. The problem of automated mesh generation is almost as old as solid modeling; great advances have been made over last 20 years (see Ref. [5] for a sampling of recent results in meshing), and most commercial CAE tools have automated the process for large classes of geometries. Unfortunately, a number of intrinsic challenges, such as those shown in Fig. 2 remain unsolved:

- **Robustness**: It is well known that boundary representations of CAD models suffer from geometric inaccuracies, local topological inconsistencies, and other unsolved robustness problems that could lead to incorrect results and system failures [6]. By its very nature, any meshing procedure must deal with all such robustness issues in order to construct a mesh—a combinatorially and topologically perfect approximation of the CAD model.
- **Small features**: By definition, all meshes must resolve all geometric details in the geometric model. There are basically two options. Conforming to the smallest features produces excessively large uniform meshes or leads to meshes with rapidly changing mesh density and severely distorted elements. Alternatively, some small features may be ignored as “geometric noise” but this approaches leads to the same robustness issues and deterministic procedures for deciding which features may be ignored is lacking.
- **Lack of guarantees**: Meshing procedures are based on a collection of heuristic procedures that may not terminate, introduce significant geometric and topological errors, and poor elements, affecting quality of the solution.

The adopted industry-wide solution is to simplify the geometric model (for example, by smoothing or by removing blends and fillets), to defeature it (for example, by eliminating small holes and protrusions), and to heal and repair it (for example, gaps, self-intersection errors, tiny edges, and surfaces, etc.). Unfortunately, these additional heuristic steps could significantly distort the original geometry and remove potentially important geometric features. They are also only partially automated and break the integration between geometric design and engineering analysis that now operate on two distinct loosely related geometric models.

The above difficulties with meshing undermine CAD/CAE integration at all stages in the product design process. They undermine interoperability of systems because CAD model translation is intrinsically imprecise; they do not support conceptual design where geometry is both imprecise and fluid; they cause difficulties in detailed design due to frequent local modifications and small features in geometry; and they hinder parametric design because meshing may not work for all parts in the parametric family. These challenges in three-dimensional conforming meshes motivated the search for alternative spatial discretization approaches that we briefly discuss below.

---

1Solidworks/Cosmosworks is an example of particularly popular and commercially successful mesh-based CAD/CAE integration.
2.2.2 Isogeometric Analysis. Fundamentally, a solid’s boundary representation is also a mesh, a two-dimensional mesh, formed by faces, edges, and vertices in the boundary representation. The geometry (embedding) of each cell may be represented by variety of functions, including Bézier, B-splines, NURBs, and implicitly defined patches. Why not use the same functions as a basis for approximating the analysis problem? This idea is most recently conceptualized as “isogeometric analysis” [7], but was also advocated by others. For example, Sabin proposed using B-splines as a basis for finite element analysis in Ref. [8], and in Ref. [9] the authors advocate using NURBs and constructive solid geometry (CSG) as a basis for analysis. The isogeometric approach with NURBs as basis functions on a finite-element mesh brings several advantages, most importantly automatic elimination of all geometric representation errors for coarse and refined meshes [7]. However, the isogeometric approach does not eliminate the fundamental limitation of the mesh-based approach, namely the heuristic nature of meshing procedures to generate 3D finite element mesh and inability to cope with geometric errors and small features in a typical boundary representation.

2.2.3 Boundary Element Integral Method (BEM). Boundary element integral methods [10] discretize the numerical solution of the problem using fundamental point-solutions (Green’s functions) that satisfy the governing equation exactly. In this case, the solution procedure is reduced to satisfying the prescribed boundary conditions, which is achieved using variational methods. The main advantage of the method over FEA lies in the fact that it offers a boundary-only approximation; spatial discretization reduces to meshing of and integration over the two-dimensional boundary of the solid, which are essentially solved problems. The two-dimensional approximation also leads to greatly reduced degrees of freedom and much smaller linear systems.

However, the BEM produces fully populated matrices which make this method computationally expensive. As a result, the computational advantages can only be realized when the model problem’s surface-to-volume ratio is quite low [11]. The method suffers from other problems as well. Nonlinearities can typically only be handled by volume integrals, which diminish the surface-only advantage. Also, there is currently no way to address variable material properties within a domain [11], because no fundamental point-solutions (Green’s functions) are known for variable materials over a given domain. For these, as well as other reasons of cumbersome implementation, the BEM has largely been confined in recent years to point-source radiation problems (such as may be found in the realms of acoustics and electromagnetism) and crack propagation (see, for example, Ref. [12]).

2.2.4 Domain Decomposition Methods. A number of researchers attempted to reduce the meshing difficulties by decomposing the domain into a set of simpler shapes that support the analysis in one form or another, often using special functions. For example, the method of external finite elements [13] used in the Procision system represented the domain as a union of simpler domains, and implementation of generalized finite elements in Ref. [14] combines the finite elements solutions on mapped domains with special functions. Some researchers suggested representing the domain as a Boolean (set) combination of premeshed primitives, in a manner similar to CSG representations used in early CAD systems [15,16].

The main difficulty with all such methods is that the subdomain or primitive meshes usually overlap and their respective meshes are incompatible with each other. The solutions constructed on such meshes are usually combined using either penalty functions, Lagrange multiplier, or some other special constructions that often lead to numerical stability issues or manual constructions. Furthermore, all such approaches alleviate the difficulties in conforming volumetric meshing but do not eliminate them. The decomposition procedures are still heuristic, undermining integration with the geometric model; and geometric errors and small features issues remain unresolved, since the CAD model still needs to be preprocessed into another geometric form.

2.3 Meshfree Approaches. The difficulty with conforming mesh-based methods led to a number of efforts to bypass explicit meshing of the domain Ω altogether and to find other ways to achieve CAD/CAE integration. The key idea is that the supports of the basis functions do not need to replace the CAD model but can be used in addition to the CAD model that takes care of all required geometric queries and computations. This is a major paradigm shift, because it allows introduction of many more approximation bases, while retaining the use of native CAD geometry.

We broadly classify all such meshfree methods into three categories, based on one of the three key underlying principles:

- A chosen basis may be enhanced with additional a priori known functions to capture geometry and/or known asymptotic behavior of the analysis. This approach is exemplified by extended (XFEM) and generalized (GFEM) the classical mesh-based finite elements approach.
- The supports of basis functions may be allocated to cover the geometric domain based on the desired accuracy of approximation, and without explicitly representing their connectivity. Most commonly, radial basis functions are associated with spherical supports, giving rise to various point cloud methods (PCMs).
- Any chosen basis must satisfy the prescribed boundary conditions. Solution structure methods (SSMs) achieve freedom from any particular mesh by building in boundary conditions into the chosen basis a priori.

Below, we briefly survey these three approaches to meshfree integration of design and analysis, but it should be clear that they are not mutually exclusive. We will refer to all such methods as meshfree, but we avoid term “meshless,” because all analysis methods rely on some form of spatial discretization in order to solve the two key problems: enforcement of boundary conditions and volumetric integration over the supports of basis functions. Our classification is based on how the meshfree methods handle geometric issues; other types of classifications have appeared elsewhere, for example in Ref. [17].

2.3.1 Generalized and Extended Finite Element Methods. Generalized finite element method (GFEM) and the extended finite element method (X-FEM) [17] propose different methods for supplementing the usual shape functions using various enhancement functions, in order to alleviate meshing problems or to capture asymptotically known behaviors (for example, in the vicinity of cracks, interfaces, or stress concentrations).

X-FEM was first introduced in Ref. [18] to model crack discontinuities. Adding special discontinuous “enrichment” functions to the existing FE basis functions enabled capturing discontinuities across the crack interface while eliminating the time-consuming task of remeshing the geometric model. Enrichment functions also help to capture the singularities at the crack tip which increases the accuracy of the numerical solution. At the beginning X-FEM was developed for 2D problems [19,20], but later it was generalized to model 3D crack propagation [21,22]. Modeling of the crack propagation required splitting the existing finite elements into separate parts. Combination of the X-FEM with level set functions [23,24] substantially simplified tracking of the crack and provided additional geometrical flexibility to the method. In this implementation the zero set of the level set function describes geometry of the crack and is used to place the enrichment functions. Modeling of the crack propagation does not require remeshing of the geometric domain rather a simple change of the level set function.

While providing means to treat discontinuities and singularities, X-FEM has not eliminated the need for mesh generation. To
overcome the difficulties in creation and handling 3D meshes of finite elements, a new group of FEM has been developed. These methods, called generalized finite element methods [25], can use spatial meshes that may or may not conform to the shape of a geometric model. GFEMs are based on the Galerkin method and use basis functions that constitute a partition of unity. These methods are usually implemented using particle shape functions [17,26–28]. The main advantage of the generalized FEMs is the ability to construct basis functions over arbitrary spatial grids. At the same time, however, this advantage becomes one of the weaknesses of these methods: since the GFEM basis functions do not satisfy the Kronecker Delta property, treatment of Dirichlet (essential) boundary conditions requires special attention. In Ref. [17], Babuška mentions a few numerical approaches which can be used to enforce Dirichlet boundary conditions. These include the penalty method, Lagrange multiplier, Nitsche and related methods, the collocation method, combination of meshless/meshfree and finite element methods, and the Kantorovich method.

2.3.2 PCMs. Point cloud methods do not represent the CAD model by a mesh, but, as shown in Fig. 3(a), cover the domain by supports of (typically) radial basis functions, that is functions that are unity at the center of the support and decay radially and inversely to the distance from the center. This category of methods includes smooth particle hydrodynamics (SPH) [30,31], the diffuse element method (DEM) [32], the reproducing kernel particle method (RKPM) [26,27], the hp-cloud method [28], and many others. The collection of the centers can be thought of a point cloud, as illustrated in Fig. 3(a). Since the points are not connected by an explicit mesh, and the supports are not aligned with the domain’s boundaries, some argue that PCMs are “meshless.” The points may seemingly be allocated and moved by the user and/or application as needed to support geometric changes, accuracy, and adaptivity of the numerical solution. Because the supports cover the domain but do not conform to its boundary, small features may no longer need to be resolved, and geometric repairs are not needed as long as the CAD model supports the necessary geometric queries.

However, PCMs complicates the two key integration tasks: enforcement of boundary conditions and volumetric integration required to assemble the system of linear equations (2). The first volumetric integral must be evaluated over over-lapping supports for every pair of incident radial basis functions. The incidence relationship between the closest neighbors in a set \( P \) of points is precisely the Delaunay tetrahedralization of the convex hull \( P \). In other words, determining the supports of the integrals in Eq. (2) requires computing a nonconforming simplicial mesh. Furthermore, only portions of the supports that lie inside \( \Omega \) are relevant, which requires additional nontrivial processing at run time. An

\[
\delta_{ij}^k \prod_{l=1}^{n} (x_l - x_j)^2.
\]

**Fig. 3** (a) PCMs typically cover the domain \( \Omega \) by circular supports of radial basis functions; the Delaunay triangulation captures the incidence between adjacent basis functions. (b) SSMs allow using any sufficiently complete set of basis functions, in this case B-splines on a uniform grid.

3.3 SSMs. XFEM, GFEM, and PCM approach to analysis eliminated the major bottleneck in CAD/CAE integration—meshing, by allowing one to choose basis functions that do not conform to the geometry of the domain, but this freedom of choice poses challenges to the two key integration tasks as discussed above. Is it possible to choose a different set of basis functions that satisfy the prescribed boundary conditions and are easy to integrate over the domain \( \Omega \)? Expression (3) suggests that such basis functions must vanish on portions of the boundary \( \Gamma^r \). Kantorovich [38] proposed the construction of such functions \( \eta_j \) by setting

\[
\eta_j = \omega x_j^k
\]

\( \omega \) is a non-negative sufficiently smooth function that vanishes on the boundary \( \Gamma^r \) where essential boundary conditions are indicated. This effectively transforms the expression (3) into the solution structure of the boundary value problem, irrespective of the particular choice of basis functions \( \psi \). A variety of basis functions, including classical FEA shape functions, radial basis functions, and trigonometric polynomials, but B-splines on a uniform orthogonal grid are particularly popular due to their simplicity and attractive computational properties [3,39,40]. In particular, choosing B-splines as basis functions, as shown in Fig. 3(b), effectively solves the last remaining CAD/CAE integration problem of computing volumetric integrals. The support of every trivariate B-spline is a cube intersected with geometric domain \( \Omega \) and integrating over such a support is a solved problem when the CAD model \( \Omega \) is given by its boundary representation [41].

It is interesting that the Kantorovich method (see Appendix A) in fact predicts most of the other discretization approaches discussed in this paper [38] and has been adapted and extended by many researchers leading to a broad category of SSMs. A Ukrainian academician, V.L. Rvachev, recognized Kantorovich’s representation of the field as a special form of the Taylor series expansion by the powers of the function \( \omega \) and showed that the notion of the solution structure generalizes to any and all types of engineering analysis; [42–44]. SSM can be viewed as a method that transforms any system of basis functions to the basis functions that satisfy prescribed boundary conditions. It can be applied to any engineering analysis method including finite element, generalized finite element, and meshless/meshfree methods [45–46]. Over the last 10 years, a number of improvements to SSM have been suggested [3,47,48]. Notably, Höllig proposed the method of weighted extended B-splines (WEB-splines), where functions \( \omega \) are constructed procedurally and the B-splines are weighted to improve the convergence and stability properties [3]. In Ref. [47], the authors demonstrated that functions \( \omega \) may be constructed as smooth approximation of the Euclidean distance function that is scan-converted via standard Euclidean distance transform [49–51].

2.4 Summary of Linear Static Case. Progress continues to be made in mesh-based FEA solutions to CAD/CAE integration,
but the intrinsic problems with geometric errors, small features, and lack of guarantee are likely to persist for some time. The meshfree approaches share the advantages and challenges of working with native CAD models, but SSM in particular provides the additional benefits of enforcing boundary conditions exactly and allowing one to choose any and all types of basis functions based on desired computational properties. These two advantages make SSM the most promising approach for CAD/CAE integration. A number of practical implementation have been reported, including WEB-splines for simple 2D Poisson problems [3], Fieldmagic, a publicly available 2D system from Ref. [52] that was built on principles described in Ref. [53], performs fully automated meshfree analysis for a range of physical problems (thermal, elasticity, plate bending and vibration, electrostatics, and several others), and most recently fully automated 3D Scan& Solve system from Intact Solutions for linear structural analysis [54].

3 Deformations

Let us now assume that we have achieved complete CAD/CAE integration for basic static engineering analysis, and the computed solution requires that the original geometry is modified. The most common example is that the original CAD model must be deformed by the computed displacement, but other examples include shape optimization or perhaps a redesign to meet some specified criteria. This is a typical case of the inverse CAD/CAE integration, but note that modified geometry requires updating the corresponding analysis model. We distinguish two cases discussed below: (1) geometric changes are continuous deformations and the results of analysis can be updated incrementally, and (2) geometric changes are substantial enough that the analysis model must be recomputed.

3.1 Continuous Deformations

In the case of calculating continuous deformation responses to loads, the traditional mesh-based approaches rely on a coordinate transformation (called a deformation gradient [55]) to cast the model strains in the new deformed configuration (often referred to as a Lagrangian approach). The deformation gradient, $F$, is given by

$$\frac{\partial F}{\partial t} \equiv \frac{\partial x_i}{\partial x_m}$$  \hspace{1cm} (5)

where the superscripts and subscripts to and t refer to the configuration at time, $t$, with respect to configuration 0 (the original configuration), respectively (see Ref. [55] for a description of the notational convention). The configuration at time $t$ is found from incremental solution coefficients $C$ according to

$$\dot{x}_m = \eta \cdot \dot{C} \cdot \left[ C - \frac{1}{2} \eta \cdot \dot{C} \right] \cdot \left[ C - \frac{1}{2} \eta \cdot \dot{C} \right] \cdot \eta \cdot \dot{C} \cdot x_m$$  \hspace{1cm} (6)

where $\eta$ is the fixed basis and $\dot{x}_m$ is the configuration with respect to the undeformed coordinates at time, $t - 1$. Equation 5 can then be immediately calculated according to

$$\frac{\partial x_i}{\partial x_m} = \frac{\partial \eta_i}{\partial x_m} \cdot \eta_k$$  \hspace{1cm} (7)

where summation is implied over repeated indices. One can see that this quantity changes with every load increment and must therefore be repeatedly calculated within an iterative solution process. The deformation gradient forms part of the finite strain tensor, which is needed both to calculate the tangent stiffness matrix, as well as for postprocessing the response at each load increment. One may consider this situation as a solved problem, in the sense that mesh-free formulations can be expected to perform in the same way as mesh-based systems and no additional CAD/CAE challenges are expected.

3.2 Large and Discontinuous Deformations

The notion of continuous deformation is useful when geometric changes in the CAD and the analysis model are small, or more precisely, the deformation preserves the topology of the CAD model (i.e., it is a homeomorphism) and the deformation is nonsingular (i.e., the gradient does not vanish). When either of these conditions is violated, the analysis model cannot be updated incrementally and must be resolved, irrespectively of the chosen analysis method. However, it has been shown [56–58] that meshfree PCM formulations allow the simulation of continuous deformations that accommodate much greater distortions than mesh-based methods. The SSM has also shown success in modeling such deformations for the case of plates and shells of arbitrary shape [59,60]. In both cases, this results in a deformation gradient that is far less sensitive to distortions and supporting a wider range of incremental analysis scenarios.

When the deformation gradient becomes singular or when CAD models undergo topological changes, continuous deformation no longer applies and the analysis model must be reconstructed. This in turn requires construction of a new basis; in the case of traditional mesh-based methods, the mesh must be modified in some way to adapt to the new geometric domain either by adding or by eliminating elements, or by completely reconstructing it [61]. By contrast, in a mesh-free solution, when large or discontinuous deformations are encountered, although the basis must be analogously modified, this is generally a substantially less complicated and costly procedure that has already been used by SSM approaches in parametric modeling [62] and shape optimization [63]. Figure 4(b) shows the second situation for a simple two-dimensional ring defined over an overlapping B-Spline support grid. It is only necessary to recompose which grid cells overlap the new deformed geometry, provided the initial grid is large enough to accommodate both the initial and final configurations. If this is not the case, intermediate grids can always be generated (based on the current deformed domain and bounding box) and elements re-created. This is a mesh-free analog of re-meshing, but would be far cheaper as structured grids may be used for this purpose.

4 Contact Problems

Broadly, contact problems begin with a determination of how multiple parts or bodies within an assembly interact. This interaction is either prescribed by design or calculated based on some criterion. Such criteria usually involve what has often been called “contact kinematics” [64], in which the relative distance, or “gap” between part boundaries is calculated. These gaps are represented by gap functions $g_k$ for each potential contact pair of surfaces that are determined by a process that typically involves some hashing procedure [65]. The gap functions simply track the minimum normal distance (gap) between surfaces and determine whether the interface is
open (no contact condition) or closed (contact). In this sense, all contact problems in CAD/CAE integration fall under the following two very basic categories:

- **Static Contact**: Part interfaces whose relative status never changes. In this case, the contact constraints may be considered known, and the location of the constraint(s) is either prescribed or calculated based on the initial gap function value. This is still a basic linear analysis.
- **Dynamic Contact**: Part interfaces whose relative status may change due to external loading. Here, a constraint is applied in the closed condition and absent or removed in the open condition. The constraint location along surfaces potentially in contact must be calculated based upon gap function value (i.e., it is applied wherever the gap is closed, and removed wherever it is open). This is a nonlinear analysis and so requires iteration.

4.1 Static Contact. When the contact conditions are known, the CAD/CAE integration issue reduces to that of specifying constraints on the boundary condition on each of the two contacting domains. In the simplest case of the “glued” elastic contact, this simply means that the contacting bodies are subject to the same displacements. This presents immediate challenge for the mesh-based methods, because the contacting bodies usually rely on incompatible meshes that do not connect at the nodes where displacements are computed. Incompatible part meshes require special interpolation/projection algorithms to apply the interface constraint. Historically, these have come in two varieties: node-to-surface algorithms [66–69] and surface-to-surface (or contact segment) algorithms [70–72]. Another approach is to introduce fictitious interface “mortar” elements [73–75] which enforce the equality of the solution (displacement) using Lagrange multipliers and iterative procedures that do not always converge. It has long been known [76] that these methods suffer from problems due to inaccuracies or ambiguities involved in the point-to-surface projections. Specifically, these may cause spurious stresses due to effective singularities related to the nonsmooth nature of the mesh (see Fig. 5(a)). Projection ambiguities can become quite serious when no unique normal direction can be found (see Fig. 5(b)), which is always the case for lower order elements on curved surfaces. It is interesting to note that a common solution to this problem involves the addition of spline surfaces which interpolate contact interface nodes [77].

The meshfree particle-based PCM approaches complicate the situation further, because they may not have enough degrees of freedom on the contact boundary to enforce the constraints. The meshfree SSM approaches are unique in their ability to enforce the known contact constraints, because they can be encoded in the solution structure a priori (independent of any choice of basis). For example, in the case of “glued” contact, the continuity of displacements across the contact boundaries is enforced automatically by the solution structure (3) that relies on common basis functions $\eta_i$. This separation of solution and geometric space is quite useful and robust for applying constraints. In Ref. [78], Sinekop proposed to deform the solution structure (3) by a coordinate transformation that contains properties of the materials on both sides of the contact boundary. This results in a solution structure that enforces equality of the displacements and stresses across the interface boundary. Other types of contact conditions may also be reflected in the solution structure. For example, the frictionless contact of the rigid and elastic bodies is a modification of the solution structure (3) described by the solution structure [42]

$$u_1 = u_0 \nabla \omega_1 + \nabla \omega_1 \times (u \times \nabla \omega_1)$$  \hspace{1cm} (8)

where $\omega_1$ is an approximate distance field to the contact boundary, $u_0$ prescribed displacement on the contact boundary, and $u$ is a displacement vector (3) that satisfies the boundary conditions on the external boundaries of the geometric domain. Purely sliding contact corresponds to setting the component of displacement $u_3 \equiv 0$ in the direction normal to the contact boundary, while displacement in the tangential plane remains unrestricted.

4.2 Dynamic Contact. Now consider a more general problem of dynamic contact where the contact conditions depend on the analysis results and must be determined as part of the simulation. In this case general, Eq. (2) is augmented by an additional constraint equation that has a general form of

$$\int_{\Gamma_c} \lambda_o \delta g_c d\Gamma_c = 0$$  \hspace{1cm} (9)

where $\lambda_o$ is the normal force on the contact surface and $\Gamma_c$ is the contact surface.

Dynamic contact problems present additional challenges in the traditional mesh-based implementation of Eq. (9) that must now dynamically and continuously interpolate the incompatible contacting meshes. Computing gap function for two meshed solids is a computationally expensive and numerically sensitive task; the computed functions may not be continuous, leading to further difficulties in constraint application and accuracy, as well as poor rates of convergence due to a lack of consistency (failure to pass the patch test).

Although contact formulations have been developed for particle-based methods (see, for example Refs. [79–81], most studies seem to focus on large deformation, rigid-flexible contact, with the exception of one assessment of the SPH method [82], in which it was discovered that spurious stress oscillations were induced on the contact surfaces (in one instance causing failure!). It should be expected that contact results will differ for particle methods depending on the precise particle formulation and will mirror the ability of these formulations to accommodate essential boundary conditions. A study of this issue can be found in Ref. [83] for the EFG method, in which it was concluded that the most stable solution was to attach traditional meshed elements to nodes on or near the boundary.

When the surfaces of contacting bodies are continuous and smooth, the problems listed above may be overcome using the SSM approach, through the use of a nonconforming fixed grid (as shown in Ref. [84]). The first reason is that using a nonconforming fixed-grid basis allows one to connect element degrees of freedom.
directly and unambiguously at part interfaces (provided they share the same basis support), thereby offering smoother solutions. This process is facilitated by first identifying “inner” and “outer” grid cells as in Ref. [3]. Figure 6 shows two surfaces in potential contact sharing degrees of freedom in the overlapping boundary cells containing them. Since the boundary cells highlighted in green on body 1 are coincident with those of body 2, the stiffness matrix entries associated with these cells simply add together. The second reason for increased accuracy for smooth surfaces is that surface normal calculations come directly from part surface definitions—not interpolated element approximations (they are exact).

In fact, we observe that gap function gn for each pair of potentially contacting solids may also be defined and constructed in a meshfree manner that does not depend on a particular discretization of the boundary. For example, if solids S and T are represented as sublevel sets of smooth non-negative real valued functions \( \zeta_S \) and \( \zeta_T \), respectively, the two solids intersect if and only the inner product \( \langle \zeta_S, \zeta_T \rangle > 0 \). This in turn allows the construction of the differentiable gap function gn as a convolution of \( \zeta_S \) and \( \zeta_T \) over the configuration space \( SE(3) \). See Ref. [85] for more details.

The use of a structured nonconforming element grid results in an additional benefit in the contact detection phase. This benefit can be realized in a four-part process [84]. First, a structured grid is placed over a bounding box encompassing all bodies to be modeled. In the second step, a single global grid configuration is created which reflects the final grid refinement of all bodies, and inner and outer grid cells for all domains within the grid support are determined. Next, all overlapping and adjacent cells determined in the previous step are identified. In the final pass, this grid is split along cells containing the inner and boundary cells of individual bodies. Contact element pairs (for surfaces in potential contact) within the new split grids are easily identified through the use of a list of offset index values stored as a result of the splitting phase. These four steps are illustrated in Fig. 6.

5 Heterogeneous Materials

Historically, mechanical CAD has dealt predominantly with geometric issues in design of components and their interaction in assemblies. Material properties have been treated as an attribute that may be selected from a suitable database, presumably in a manner that would maximize compliance, reduce weight, etc. This approach is reasonable when materials are isotropic and homogeneous and has no bearing on CAD/CAE integration. Anisotropic materials are slightly more complicated but can be modeled by introducing either local or global coordinate systems, though few commercial CAD systems support such specifications today, and it is more common to find that anisotropy is specified with the analysis model. In recent years, a number of new heterogeneous materials have emerged. Traditional homogeneous materials are being replaced by functionally graded materials whose properties vary spatially and by composite and nanomaterials whose properties are being controlled at different scales. New manufacturing processes, such as, for example, plasma spraying and rapid prototyping, enable creation of unique materials which can withstand very harsh conditions. The progress in the development of fuel cells is unthinkable without nanomaterials and nanocoatings. But introduction of such heterogeneous materials significantly complicates CAD/CAE integration as we discuss below.

5.1 Spatially Varying Materials. A simplest example of spatial material variation is a discrete transition in material properties, when two different homogeneous materials are joined together—either as a single part or as a static assembly of components. In this case, the CAD/CAE integration issues are identical to those arising in the static contact problems discussed in Sec. 4 and amount to specifying and enforcing the appropriate contact boundary conditions at the material interface boundaries.

By contrast, functionally graded materials, for example, those manufactured by plasma spray deposition, vary continuously throughout the geometric domain. This poses additional challenges to both CAD and CAE systems, because the interior of CAD models is usually not parameterized. An obvious approach to modeling such spatial variation is to adapt a mesh-based finite element approach as was proposed in Ref. [86]. However, this scheme requires the spatial mesh to adapt to the geometry of the model, as well as to the distribution of the material properties and to the analysis results. Regions with a rapid change of the material properties require denser grids. Two apparent drawbacks of this scheme include: large memory requirements to represent and archive the material properties, and introduction of unwanted discontinuities and other artifacts into the physical fields being modeled. For example, piecewise constant material properties cause discontinuities in strains on the boundaries where material properties are discontinuous. Representations of continuously varying material properties using trivariate simplex splines and B-spline have been proposed in Refs. [87,88]. These representations result in continuously differentiable distributions of the material properties. They also enable multiresolution and local adaptive subdivision, but still may require substantial memory resources. Nor is it clear how such material properties may be controlled or edited by the user, and how they may be updated in response to design changes. Maintaining multiple distinct meshes that must be recreated and remain compatible at all times is not a practical solution.

A number of meshfree approaches to modeling material distributions have been proposed. For example, spatially variable material properties can be represented by polynomials of a known...
The coefficients of the polynomial have to be either specified by the user or computed to approximate a given distribution of the material properties. This approach does not introduce any additional artifacts to the modeling results, but may sometimes lead to inconsistent representation of the material properties due to approximation errors. Feature-based heterogeneous object modeling with blending [90] associating material properties with geometric and material features and enables construction of the continuously varying distributions of the material properties using B-spline lofting. This approach simplifies specification of the material properties, but it may require redefinition of the material properties when the geometric model changes.

A material representation based on the solution structure method was proposed in Ref. [91]. It uses distances to the geometric and material features to specify and interpolate variable material properties prescribed by user on geometric features of a CAD model. Since this representation uses distances to the features, besides specifying material properties, the user may also define their rate of change away from the features. A remainder term of the solution structure provides degrees of freedom that can be used to optimize the distribution of the material properties for the particular design objectives. Since geometry, material properties and physical field are represented in the same form, this material representation scheme enables shape-material optimization in the particular design objectives. Since geometry, material properties and physical field are represented in the same form, this material representation scheme enables shape-material optimization in the same computational framework. Such a representation is particularly effective if SSM is also used for analysis, because it requires no additional utilities for CAD/CAE integration [92]. The SSM representation of heterogeneous materials was recently proposed as a basis for development of a standard for heterogeneous material representation [93].

5.2 Multiscale Materials. Composite materials constitute a large class of heterogeneous materials that require specialized treatment during design, analysis, and manufacturing stages. These materials consist of fibers or particles of a very strong material (carbon or glass fibers, grains, reinforcement elements, etc.) combined by a matrix material [94]. Usually, the size of the reinforcement elements is very small in comparison with the size of a part. At the microscale, composite materials can be considered as piecewise homogeneous and high periodically structured materials whose interface boundaries have been glued together. Representing geometry of such materials for a large mechanical component is not practical using traditional CAD tools. However, when woven and other periodic structures are represented implicitly by functions vanishing on their boundaries, coordinate transformations may be used to replicate them throughout any component without increasing the complexity of representation [95]. At the macroscale, where we are mostly interested in the material’s behavior as a whole, they can be treated as anisotropic homogeneous materials [96]. In this case, interaction of the microstructural reinforcement elements with the matrix material is not considered. For analysis purposes, the composite material is replaced with a “homogenized” material with the equivalent “integral” physical properties [97,98]. This type of analysis, however, cannot address some very important issues of reliability and disintegration of the material. In order to predict the material’s failure, crack development and delamination the material’s microstructure has to be taken into account.

The key challenge for CAD/CAE integration with composite materials is that the failure analysis has to be performed on a part as a whole at the macroscale, but at the same time the material’s behavior at the microscale has also to be taken into account. This can be achieved by using multiresolution approaches which include adaptive meshing and solution refinement [99–102]. Meshfree methods, and especially SSM approaches, can also use hierarchical basis functions such as, for example, hierarchical B-splines [3,103] and wavelets [104–106]. Simulation of composite materials with periodic structure of the reinforcement elements can be also efficiently be performed by the solution structure method [107]. Partition of unity methods could be used to connect basis functions defined over grids with different resolution [99]. These numerical techniques have an apparent drawback: numerical complexity of the problem grows fast as ratio of part’s size to the size of the reinforcement element increases. A different approach to analysis of composite materials have been proposed in [97,108–110]. It consists in development of the specialized finite element basis functions that could mimic behavior of the material’s microstructure on the macroscale.

Similar challenges arise in modeling and simulation of nanomaterials that are defined by a matrix material with added nanoparticles. Nanoparticles such as, for example, carbon nanotubes, have extremely high mechanical properties. Using nanoparticles as the reinforcement elements holds the promise of designing and manufacturing light and at the same time high strength materials. Recent advances in nanomaterials and nanotechnology [111] inspired development of engineering analysis methods that are capable of overcoming computational limits imposed by tremendous differences in the dimensional scales at which computations have to be performed [112].

But, in addition to the huge difference of dimensional scales, analysis of nanomaterials has to account different physical processes at different scales. At the nanoscale, physical processes are described by the laws of molecular dynamics. At the microscale and macroscale, equations of continuum mechanics can be used. This substantially complicates development of efficient and reliable numerical analysis methods capable to address these issues. To date, a few numerical multiscale techniques have been developed [113–116]. The main idea behind these techniques is to split the computational domain into three computational subdomains [114,116]. One computational subdomain is located at the nanoscale, another one at the macroscale and the third one serves as a “bridge” between “nano” and “macro” subdomains. In “nano” zone physical problem is solved using ab-initio principles utilizing the equations of molecular dynamics. “Macro” analysis is performed using the equations of continuum mechanics. “Bridge” zone serves as a transition between continuum and molecular dynamics. In some sense, these approaches are similar to the ones used in multiphysics simulations (see discussion in Sec. 6).

6 Multiphysics Integration

Realistic CAE models of complex real-world physical phenomena usually involve several interacting physical fields at similar space/time scales. Common examples of such models include multiphase flow, phase transition and fluid-structure interaction problems. Mathematically such problems are described by coupled (weakly or strongly) partial differential equations [117]. Weak coupling is used when there is a one-way communication among the interacting physical fields. It reduces the full scale multiphysics problem to a sequence of individual physical problems. Each problem in the sequence is solved one-by-one and the information on the interacting fields is passed from one problem to another. Problems in thermoelasticity provide a simple example of weakly coupled problems. Strong coupling is used for highly nonlinear problems with two-way communication among the individual problems. It reduces the solution of a multiphysics problem to a system of nonlinear algebraic equations. Strong coupling allows one to achieve better accuracy and faster convergence, but it requires substantial computer memory resources to store the algebraic system. Aeroelasticity, fluid-structure interaction, piezoelectric, piezoresistive, thermoelastic, electroelastic, and thermoelastic damping problems require strong coupling. Some strongly connected multiphysics problems may be reformulated as a sequence of weakly connected problems. But this may have a negative effect on the accuracy and convergence of the solution.

From the point of view of the integration of design and analysis, one of the key challenges in solving all multiphysics problems is flow and consistency of the field data between the coupled physical problems. Since each field is spatially discretized and
approximated in a distinct basis, the data flow problem involves several different and possibly incompatible discretizations (meshes or covers) of the CAD model. If the interacting physical fields share a common volumetric domain, mesh-based engineering analysis methods employ a common spatial grid to perform all data exchange and communication. Since this spatial grid has to incorporate constraints specified by each physical problem, this usually results in a much denser grid than required for any individual analysis. Assuming that the integration problem is solved for each individual physical problem, computer implementation of such an interaction scheme is fairly straightforward but comes at a fairly high computational cost since all physical quantities are now stored at every node of this very fine resolution grid.

In other multiphysics problems, such as, for example, aeroelasticity or fluid-structure interaction problems, physical fields interact on the common boundary of two disjointed geometric domains. Such multiphysics problems may also require different numerical methods to be used: the finite element method is used for the structural subproblem while the finite volume method is employed to model motion of the fluid. This in turn results in incompatible meshes and approximations of the physical fields on the interface boundary, leading to difficulties analogous but more difficult than in contact modeling. One obvious source of mesh incompatibility is the fact that the meshes for individual physical problems are often generated by different programs. But it is also well known that in order to obtain accurate results, spatial meshes used for fluid flow modeling have to be much denser than those used for structural analysis.

Methods to couple the computational fluid dynamics and structural analysis discretizations include direct interpolation, penalty, Lagrange multipliers, and transition (mortal) elements [118–122]. Direct interpolation and penalty method generally fail interface patch test [117] and could result in spurious modes if mesh resolution is different on both sides of the interface boundary. According to Ref. [118], the best results can be obtained if coupling is performed using localized Lagrange multipliers method with mortar elements. Lagrange multipliers approach can also be used to interface finite element and finite volume methods in one computational framework to model fluid-structure interaction [123].

It is reasonable to expect that meshfree methods should be able to avoid many difficulties arising in coupling of incompatible spatial discretizations. For example, the immersed boundary method [124] originally developed by Peskin, involves both Eulerian and Lagrangian variables coupled by the Dirac delta function. The immersed boundary method uses fixed Cartesian and moving curvilinear grids for Eulerian and Lagrangian variables, respectively. The interface boundary is described by a smooth approximation of the Dirac delta function that can be constructed over the grid points. A combination of the immersed boundary and level set methods have recently been proposed [121]. The level set method enables decoupling the motion of the fluid-structure interface boundary from the underlined Cartesian grid, which, in turn, enables modeling of very complex motion of the interface boundary [125]. To enforce essential boundary conditions, the immersed boundary method can also be implemented using a ghost-cell methodology [126,127]. It computes the flow using nonuniform Cartesian grids while the immersed boundary is represented by grids of unstructured triangular elements. Another variation of the immersed boundary method uses the discontinuous Galerkin formulation to impose Dirichlet boundary conditions strongly [128,129]. The approach proposed in Ref. [130] enables enforcement of the essential boundary conditions by coupling of the immersed boundary method with an integral equation formulation. To connect the finite element and integral equation solutions, this approach utilizes representation of the physical domain by signed distance function. This technique results in the discretized equations whose conditioning does not depend on the shape and number of cut-elements. In this sense, this approach to fluid-structure interaction is close in spirit to SSM techniques.

7 Conclusions: Toward Fully Integrated Applications

We have intentionally avoided discussing any specific application in the broad area of CAD/CAE integration; instead we focused on geometric interoperability issues in fundamental integration tasks that comprise many such applications. For example, shape and topology optimization relies on linear static analysis, shape deformation, and possibly heterogeneous material modeling; engineering analysis of assemblies requires solutions to problems of linear static analysis, contact, deformation, and heterogeneous material modeling; and so on. We do not claim that our list of fundamental tasks is complete (for example, we have not discussed the issues of sensitivity computation), but it includes the most common tasks that are required by the vast majority of applications spanning the whole product life cycle engineering. While we focused on structural mechanics as a prototypical CAE application, key issues in CAD/CAE integration are similar in other domains. For example, meshing problems appear to dominate integration issues in computational fluid dynamics, and a number of meshfree approaches have been proposed [124,131–133] that parallel the developments in structural mechanics.

Critical examination of geometric issues in CAD/CAE integration with respect to these tasks reveals a mixed picture. Great advances are being made in both mesh-based and meshfree approaches to engineering analysis, but none of the fundamental integration tasks have been fully solved today. While mesh-based approaches appear to have an edge in their simplicity and computational efficiency, they are intrinsically limited when it comes to dealing geometric inaccuracies, small features, large deformations, and contact simulation. Meshfree PCM approaches have been demonstrated with great success in the solution of nonlinear problems (see Secs. 3 and 4), but problems involving consistency and the imposition of essential boundary conditions remain [83]. By contrast, meshfree methods involving nonconforming B-Spline basis grids and the SSM do not suffer from these problems, and applications of the SSM to modeling contacts, cracks, and multiphysics may require nontrivial extensions. For example, extending the structure (8) to nonlinear problems involving changing contacts has yet to be demonstrated. In Ref. [84], the method was used for small-deflection Hertzian-type elastic contact problems, but the contact constraint was applied via a penalty formulation rather than the SSM. The method of enrichment functions used by Belytschko and Fleming [80] is clearly a type of SSM and its generalization appears to be applicable to other types of boundary conditions to enforce equilibrium within nonlinear solution iterations.

Thus, a number of challenges remain, particularly in the areas of contact modeling and multiphysics, but also even in the basic case of linear static analysis. Because the meshfree methods cover the geometric domain but do not conform to it, small features and geometric errors may not be fully resolved. This simplifies complete CAD/CAE integration, but the effect of such inaccuracies and small features remains to be quantified, and potentially could be a source of significant analysis errors that require standard posteriori analysis and multiresolution techniques [99,102,134]. As of this writing, the issues of computational efficiency have not been fully resolved either. On one hand, the advantages of meshfree methods come with a computational overhead of constructing, differentiating, and integrating the basis functions over unmeshed domain at run time; on the other hand, the meshfree methods do not require computationally intensive and nondeterministic meshing procedures and are generally easier to parallelize, making them much more suitable for emerging heterogeneous computing platforms and architectures.

Acknowledgment

This research is supported in part by the National Science Foundation grants CMMI-0856778, CMMI-1029553, CMMI-1042211, and CMMI-0900219.
Due to the complexity of the document, the text is not legible. However, it appears to be a page from a technical paper discussing the Kantorovich method in the context of solving boundary value problems using differential equations.

Appendix A: The Kantorovich Method

This classical numerical technique for approximating solutions of partial differential equations with homogeneous boundary conditions was described by Leonid Kantorovich in Ref. [38] and has become known as the Kantorovich method. The key idea of the method is based on the observation that the solution of a differential equation with homogeneous Dirichlet boundary conditions $u_{|\partial \Omega} = 0$ can be represented in the form

$$u = \omega \Phi$$  \hspace{1cm} (A1)

where $\omega$ is a known function that takes on zero values on the boundary of the domain $\partial \Omega$ and is positive in the interior of $\Omega$, and $\Phi$ is some unknown function. For example, in the Fig. 7, the function $\omega(x, y)$ is identically zero on the boundary of the shown two-dimensional domain and is positive in the domain’s interior. As such, $\omega$ completely describes all the geometric information for the homogeneous Dirichlet boundary value problem, and in fact any function $u$ of the form (A1) will satisfy the homogeneous boundary conditions exactly. In the context of structural analysis, the solution field function $u$ is displacement, and the homogeneous boundary conditions correspond to the boundaries that are rigidly fixed.

The expression (A1) contains no information about the differential equation of the boundary value problem. Rather, it represents the structure of any solution to a boundary value problem satisfying the given boundary conditions. For any given boundary value problem, determination of the unknown $\Phi$ immediately translates into solution to the boundary value problem. Since we usually cannot expect to determine such $\Phi$ exactly, we can approximate it by a finite (convergent) linearly independent series

$$\Phi = \sum_{i=1}^{n} C_i \zeta_i$$  \hspace{1cm} (A2)

where $C_i$ are scalar coefficients and $\zeta_i$ are some basis functions. Kantorovich relied on the standard global polynomial basis, but many other well-known basis functions, such as B-splines, radial basis functions, or finite element shape functions among others can be used to represent $\Phi$. For example, function, whose plot is shown in Fig. 7(b), is a combination of the function $\omega$ (Fig. 7(a)) and bicubic B-splines $\zeta_i$ defined over uniform $30 \times 30$ Cartesian grid with randomly chosen coefficients $C_i$.

It is important that the structure (A1) does not place any constraints on the choice of the basis functions $\{ \zeta_i \}$ that approximate the function $\Phi$. In particular, the choice of the basis functions does not depend on any particular spatial discretization of the domain. The grid of B-splines in our example is aligned with the space and not with the domain. For any given boundary value problem and a choice of the basis functions $\{ \zeta_i \}$, the approximate solution is obtained as

$$u = \omega \sum_{i=1}^{n} C_i \zeta_i$$  \hspace{1cm} (A3)

using variational, projection, or a variety of other numerical methods to solve for the numerical values of the coefficients $C_i$. For example, if we choose the coefficients to approximate the solution of the differential equation $\nabla^2 u = 1 - \sin(y)$ in the least square sense, we obtain the function $u$ shown in Fig. 7(c). From a computational point of view, the intrinsic advantage of the methods lies in the clean and modular separation of the geometric information represented by the function $\omega$ from the differential equation and numerical procedure used to determine the analytic component $\Phi$.

References


