Scan&Solve™ : FEA without Meshing

White Paper

Intact Solutions, LLC
www.meshfree.com
email: info@intact-solutions.com

August 1, 2009

Abstract:

Scan&Solve™ software for engineering analysis from Intact Solutions is based on a patented meshfree technology that liberates Finite Element Analysis (FEA) from the dependence on and limitations of meshing. The salient feature of the technology is separate handling and controls of geometric and physical computational models that are seamlessly combined at solution run time. The advantages of this approach include unprecedented flexibility in handling geometric errors, small features, complex boundary conditions, and interfaces, while maintaining most of the benefits of classical finite element analysis. Scan&Solve™ can be applied to any geometric model and used within any geometric modeling system that supports two fundamental queries: point membership testing and distance to boundary computation. This white paper describes the technical background behind the Scan&Solve™ technology, summarizes its implementation, and demonstrates it advantages.

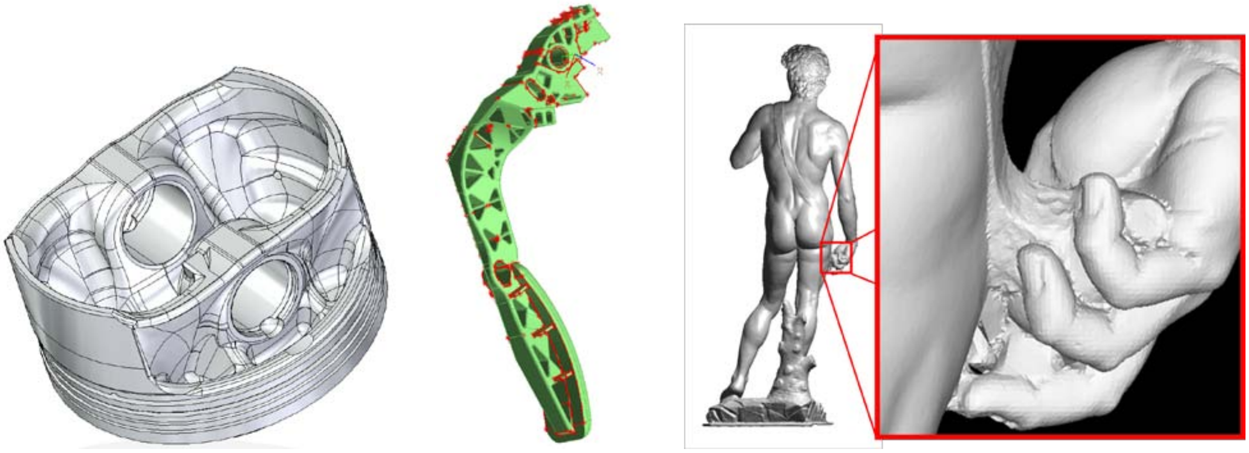
1. Introduction

The tyranny of meshing

Over the last fifty years, Finite Element Analysis (FEA) has become the predominant tool in engineering analysis, offered by virtually all Computer Aided Engineering (CAE) vendors and used in a majority of analysis and simulation applications. Yet, the acceptance and adoption of CAE tools has been slow, and the exceedingly popular and widely accepted predictions of rapidly accelerated growth in the CAE arena do not appear to have materialized.[1] Why CAE tools have consistently been lagging the technological advances in CAD is a complex and multi-faceted issue, but it is clear that CAE tools are still difficult to apply to geometric models in many realistic situations. By all accounts, the main culprit is not the FEA method itself, but the process of preparing geometric data in a form acceptable for FEA. Specifically, all commercial FEA codes require that the geometric model be converted into a conforming mesh of elements via an expensive and heuristic procedure known as meshing. The resulting mesh is not intrinsic to the original geometric model, introduces additional errors, is expensive to compute, and affects the quality (or lack of it) in the FEA solutions.

[1] The origins of both Computer Aided Design (CAD) and Computer Aided Engineering (CAE) software can be traced to 1950's and 1960's when theoretical foundations of both disciplines were developed in anticipation of then revolutionary computer technology. Fifty years later, Product Life Cycle Management (PLM) is a rapidly growing $15-20 billion global market, of which CAE software and services represent only 10%. CAD related activities account for much of the rest.

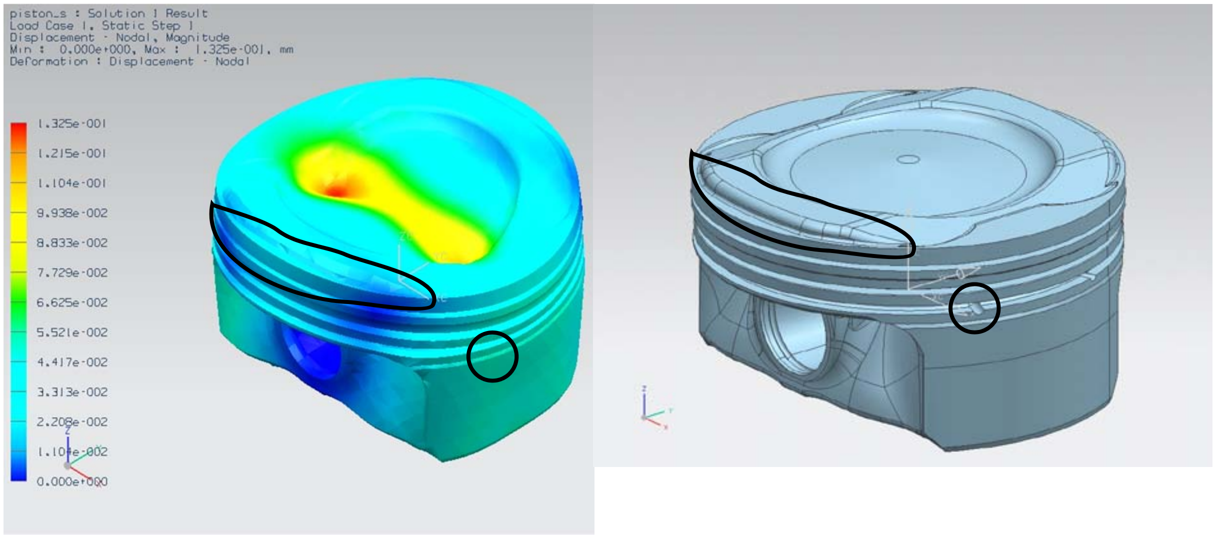
By definition, meshes must adapt to the smallest geometric detail in geometric model, leading to excessively large meshes, and making accurate meshing impractical for any geometric model with small features or geometric errors. This applies, for example, to all models shown in Figure 1 either due to presence of small features or because of small geometric errors in the geometric model.



(a) (b) (c)

Figure 1: Accurate meshing is difficult or impossible for: (a) piston with small features, (b) pedal with geometric errors, and (c) David with noisy triangulated surface.

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 are only partially automated, and break the integration between geometric design and engineering analysis that now operate on two distinct loosely related geometric models. For example, the FEA solution for a solid in Figure 2 relies on a mesh that significantly distorted the original geometry and removed potentially important geometric details.



```
piston_s : Solution 1 Result
Load Case 1, Static Step 1
Displacement - Nodal, Magnitude
Min :  0.000e+000, Max :  1.325e-001, mm
Deformation : Displacement - Nodal
```

Figure 2: The Finite Element solution (left) relies on a mesh that significantly simplifies and distorts the geometry of the solid (piston) on the right. Two of many such distortions are indicated above.

Many man-years and millions of dollars have been and continue to be invested into improving finite element meshing technologies. But it is important to recognize that the above limitations are intrinsic to all mesh-based approaches, and cannot be resolved by incremental improvements in meshing technologies. These very same limitations prevent wider adoption of finite element analysis in many scientific, engineering and consumer applications, ranging from engineering to art and medicine – not because finite element analysis is difficult, but because the tedious and error prone process of data preparation and meshing make it impractical.

Scan&Solve™ : engineering analysis in situ

Scan&Solve™ was developed specifically to liberate FEA from the tyranny of meshing, while preserving most of the advantages of this classical and widely accepted method of engineering analysis. The basic idea is simple: create separate geometric and physical representations of the model in question and combine them only when necessary, without requiring expensive and error-prone data conversions or always using the most authentic representation available. The concept is illustrated in Figure 3.

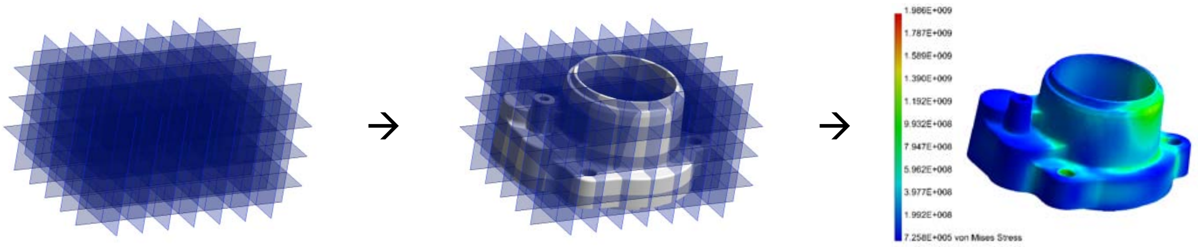


Figure 3: Scan&Solve™: a geometric model is immersed into a 3D grid of space; the basis functions of the grid are modified by the solid and the boundary conditions at run time to solve the field problem.

The analysis model is constructed on a (typically, but not necessarily) uniform orthogonal grid of space that initially knows nothing about the model being analyzed. It can be thought of as a 3D "graph paper". The usual (variety of) basis functions[2] are associated with the vertices of this mesh. The geometric model exists in the same space, in its native unaltered form, and is not aware of the mesh surrounding it. The geometric model can come from any source, as long as it is clear which points in space belong to the model and which do not, and it is possible to compute the distance from any given point to the model's boundary.

To perform analysis on a geometric model, such as structural analysis, one must specify boundary conditions such as restraints and loads on the boundary of the native geometric model. The analysis problem will then be solved on the uniform orthogonal mesh, but the usual FEA procedure is modified at run time to account for the existence of the geometric boundaries, restraints, and loads via the Scan&Solve™ process. Section 4.2 shows and discusses Scan&Solve™ solutions for native geometric models in Figure 1.

[2] Basis functions are often called "shape functions" in FEA terminology.

The theoretical foundations of the Scan&Solve™ process are explained in Section 2, and implementation details are broadly outlined in Section 3. Experimental results and comparison with known solutions is described in Section 4. We conclude this section by a brief comparison of the Scan&Solve™ and the classical mesh-based FEA, shown in the Table below.

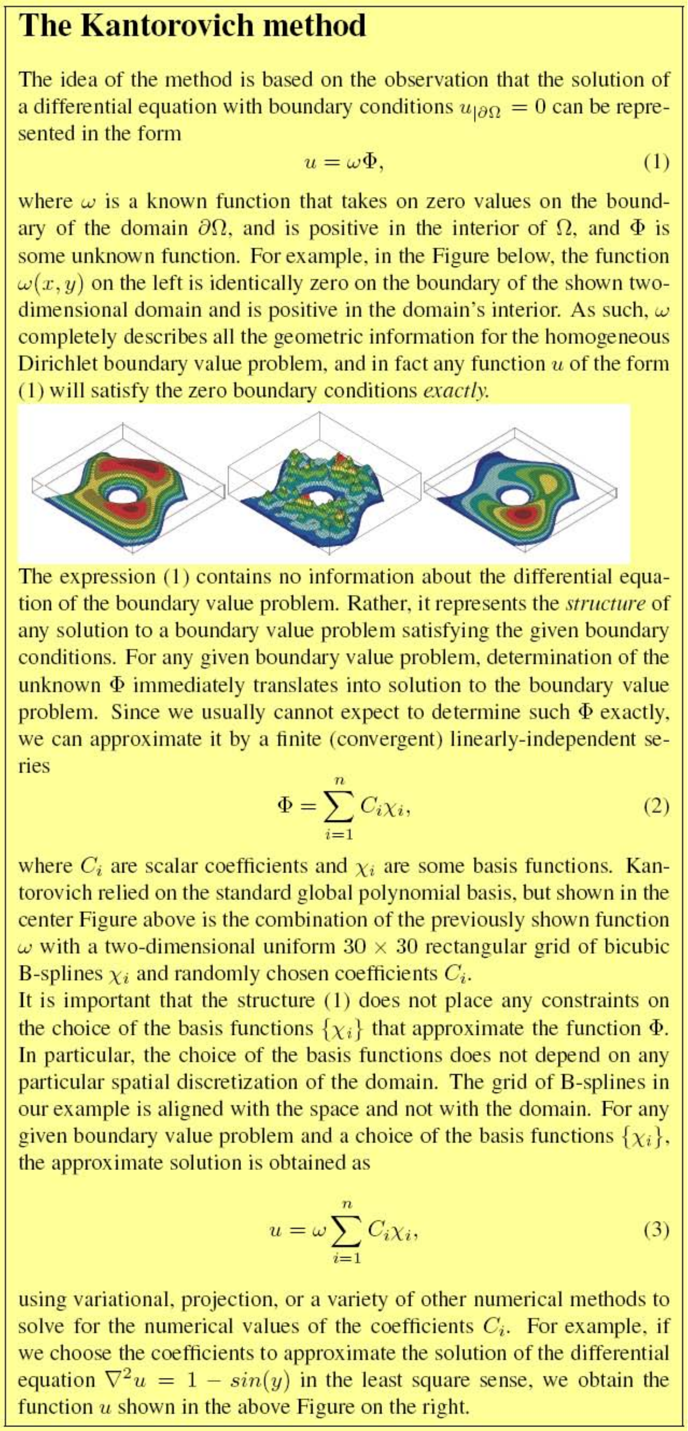
Mesh-based FEA	**Scan&Solve™**
Geometry approximated by the mesh	Native geometry is used
Preprocessing: heuristic simplification & meshing	Preprocessing: none
Meshing must resolve all geometric errors and tolerances	Geometric errors are irrelevant as long as points can be classified and distances to the boundary can be computed
Mesh size is determined by the smallest feature size	Mesh size is determined by the desired resolution of the analysis model (uniform grid)
Small features must be removed	Small features are preserved and handled automatically
Boundary conditions: enforced at the nodes only	Boundary conditions: enforced on all points of the boundary
Derivatives: pre-computed	Derivatives: pre-computed & run time
Integration: Gauss points of finite elements	Integration: Gauss points determined at run time
Basis functions: local support	Basis functions: local support
Sparse linear system	Sparse linear system
Geometric accuracy control: fixed and limited by the mesh	Geometric accuracy control: determined by accuracy of geometric computations (point test, distance); adaptive.
Analysis accuracy control: h-, p-, and k-refinement	Analysis accuracy control: h-, p-, and k-refinement

Table 1: Comparison of mesh-based FEA and Scan&Solve™ technology.

2. How Scan&Solve™ works

General theory and historical background

Scan&Solve™ is rooted in a classical numerical technique developed in the 1950's by a well known mathematician, Leonid Kantorovich.[3] The technique was described in [5] and has become known as the Kantorovich method. The key idea of the method is summarized in Figure 4 and amounts to representing the solution of the analysis problem as a product of two functions: one known function ω implicitly represents the boundaries of the shape where the field function is zero, and a second unknown piecewise function Φ which captures the analytic behavior of the field throughout the domain. Originally, the latter function was represented in a global polynomial basis, but can be also constructed using many other well known basis functions, such as B-splines, radial basis functions, or finite element shape functions among others. In the context of structural analysis, the solution field function **u** is displacement, and the zero boundary conditions correspond to the boundaries that are rigidly fixed. From a computational point of view, the intrinsic advantage of the method lies in the clean and modular separation of the geometric information represented by the function ω from the differential equation and numerical procedure used to determine the analytic component Φ.



The Kantorovich method

The idea of the method is based on the observation that the solution of a differential equation with boundary conditions $u_{|\partial\Omega} = 0$ can be represented in the form

$$u = \omega\Phi, \tag{1}$$

where ω is a known function that takes on zero values on the boundary of the domain $\partial\Omega$, and is positive in the interior of Ω, and Φ is some unknown function. For example, in the Figure below, the function $\omega(x, y)$ on the left is identically zero on the boundary of the shown two-dimensional domain and is positive in the domain's interior. As such, ω completely describes all the geometric information for the homogeneous Dirichlet boundary value problem, and in fact any function u of the form (1) will satisfy the zero boundary conditions *exactly*.

The expression (1) 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 Φ immediately translates into solution to the boundary value problem. Since we usually cannot expect to determine such Φ exactly, we can approximate it by a finite (convergent) linearly-independent series

$$\Phi = \sum_{i=1}^{n} C_i\chi_i, \tag{2}$$

where C_i are scalar coefficients and χ_i are some basis functions. Kantorovich relied on the standard global polynomial basis, but shown in the center Figure above is the combination of the previously shown function ω with a two-dimensional uniform 30×30 rectangular grid of bicubic B-splines χ_i and randomly chosen coefficients C_i.
It is important that the structure (1) does not place any constraints on the choice of the basis functions $\{\chi_i\}$ that approximate the function Φ. 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 $\{\chi_i\}$, the approximate solution is obtained as

$$u = \omega \sum_{i=1}^{n} C_i\chi_i, \tag{3}$$

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 the above Figure on the right.

Figure 4: The Kantorovich Method

Over the years, the Kantorovich method has been rediscovered by others many times and extended in various ways. Most recently, Generalized Finite Element Analysis (GFEM) and Extended Finite Element Method (EFEM) [2] propose different methods for supplementing the usual shape functions using various enhancement functions, ω, in order to alleviate meshing problems or to capture asymptotically known behaviors (for example, in the vicinity of cracks, interfaces, or stress concentrations). A Ukrainian academician, V.L. Rvachev, recognized

[3] Kantorovich was also awarded the Nobel Prize in Economics in 1975.

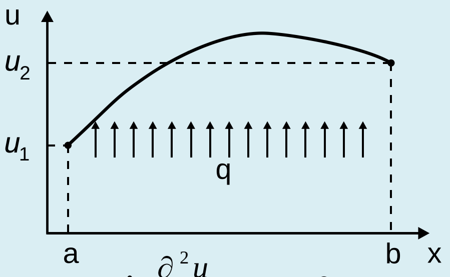
Kantorovich's representation of the field as a special form of the Taylor series expansion by the powers of the function ω and showed that the notion of the solution structure generalizes to any and all types of engineering analysis [7, 8, 9]. He also developed a theory of R-functions specifically as a method for constructing these functions, ω, for arbitrary shapes [7, 10, 11]. Unfortunately, the constructions based on R-functions require solving a number of difficult algorithmic problems for realistic 3D solids and do not appear to scale well with increase in geometric complexity.

The Scan&Solve™ technology fulfills the promise of the sound theoretical foundations laid by Kantorovich, Rvachev, and others, achieving full automation, generality, and scalability for realistic finite element analysis problems. The key principle behind Scan&Solve™ is the identification of the somewhat mysterious function ω with the more familiar concept of the Euclidean distance function. As we will explain below, this association allows transformation of all finite element based solutions into a straightforward and efficient computational pipeline that can be used within any CAD/CAE system. Before we explain how this pipeline is built in Section 4, it is instructive to compare the classical FEM and Scan&Solve™ using a simple 1-D example.

FEM vs. Scan&Solve™: One dimensional example

For illustration purposes, consider a common one-dimensional example shown in Figure 5. A string is subject to an axial tension λ, fixed at the two end points x=a and x=b, and is loaded by the vertical distributed force q. The problem is to determine the shape of the string by solving for the vertical displacement u(x).

Figure 5: String in tension. Problem formulation.

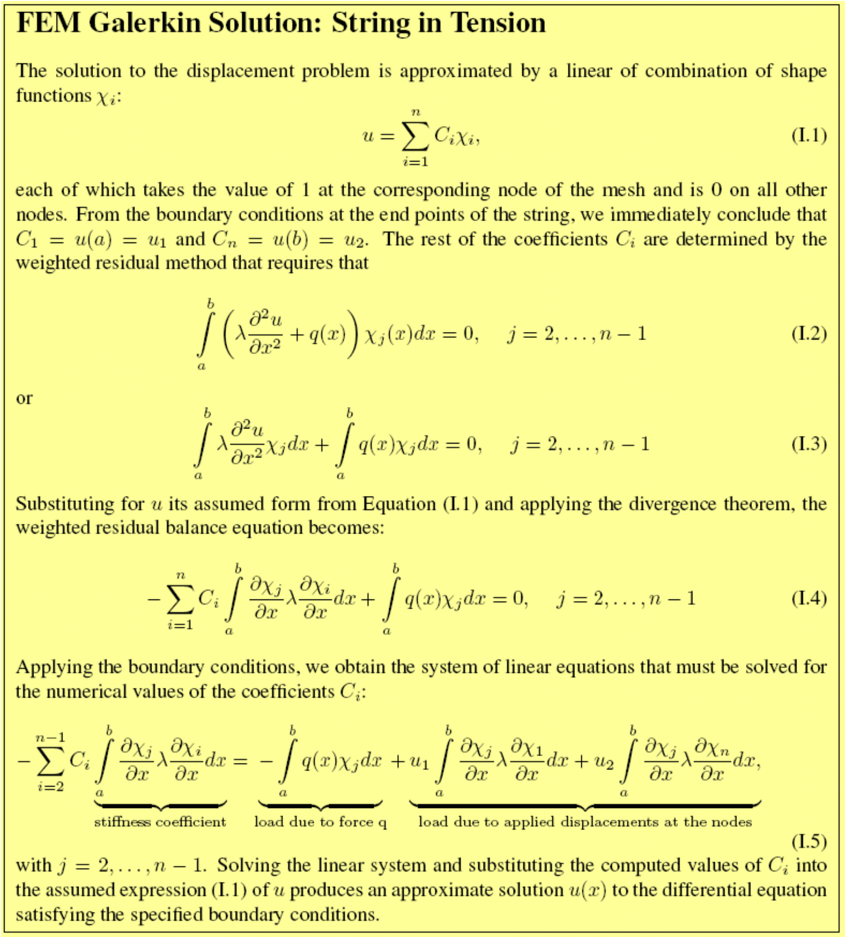


$$\lambda \frac{\partial^2 u}{\partial x^2} + q = 0$$

$$u\big|_{x=a} = u_1$$

$$u\big|_{x=b} = u_2$$

The classical Galerkin (weighted residual) formulation of the Finite Element Method is detailed in Figure 6 and is illustrated graphically in Figure 7. In this very simple 1D example, the finite element mesh is simply a subdivision of the line segment ab into smaller line segments. Linear basis functions are associated with the nodes in that subdivision, yielding the "saw tooth" pattern. The first and the last nodes must coincide with the end points a and b, respectively, so that the displacements u_1 and u_2 may be enforced at these locations. Equation (I.5) in Figure 6 represents a system of linear equations, where the left hand side is a matrix of stiffness coefficients and the right hand side amounts to integration of known quantities: distributed force q and imposed boundary conditions. The basis functions have compact support (which means that they overlap only with their neighbors), implying that the stiffness coefficients can be computed locally and efficiently, and the resulting stiffness matrix is sparse and banded.



FEM Galerkin Solution: String in Tension

The solution to the displacement problem is approximated by a linear of combination of shape functions χ_i:

$$u = \sum_{i=1}^{n} C_i \chi_i, \tag{I.1}$$

each of which takes the value of 1 at the corresponding node of the mesh and is 0 on all other nodes. From the boundary conditions at the end points of the string, we immediately conclude that $C_1 = u(a) = u_1$ and $C_n = u(b) = u_2$. The rest of the coefficients C_i are determined by the weighted residual method that requires that

$$\int_a^b \left(\lambda \frac{\partial^2 u}{\partial x^2} + q(x) \right) \chi_j(x) dx = 0, \quad j = 2, \ldots, n-1 \tag{I.2}$$

or

$$\int_a^b \lambda \frac{\partial^2 u}{\partial x^2} \chi_j dx + \int_a^b q(x) \chi_j dx = 0, \quad j = 2, \ldots, n-1 \tag{I.3}$$

Substituting for u its assumed form from Equation (I.1) and applying the divergence theorem, the weighted residual balance equation becomes:

$$-\sum_{i=1}^{n} C_i \int_a^b \frac{\partial \chi_j}{\partial x} \lambda \frac{\partial \chi_i}{\partial x} dx + \int_a^b q(x) \chi_j dx = 0, \quad j = 2, \ldots, n-1 \tag{I.4}$$

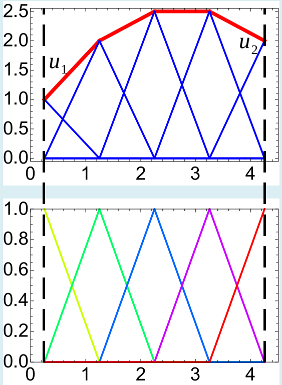
Applying the boundary conditions, we obtain the system of linear equations that must be solved for the numerical values of the coefficients C_i:

$$-\sum_{i=2}^{n-1} C_i \underbrace{\int_a^b \frac{\partial \chi_j}{\partial x} \lambda \frac{\partial \chi_i}{\partial x} dx}_{\text{stiffness coefficient}} = \underbrace{-\int_a^b q(x) \chi_j dx}_{\text{load due to force q}} + \underbrace{u_1 \int_a^b \frac{\partial \chi_j}{\partial x} \lambda \frac{\partial \chi_1}{\partial x} dx + u_2 \int_a^b \frac{\partial \chi_j}{\partial x} \lambda \frac{\partial \chi_n}{\partial x} dx}_{\text{load due to applied displacements at the nodes}}, \tag{I.5}$$

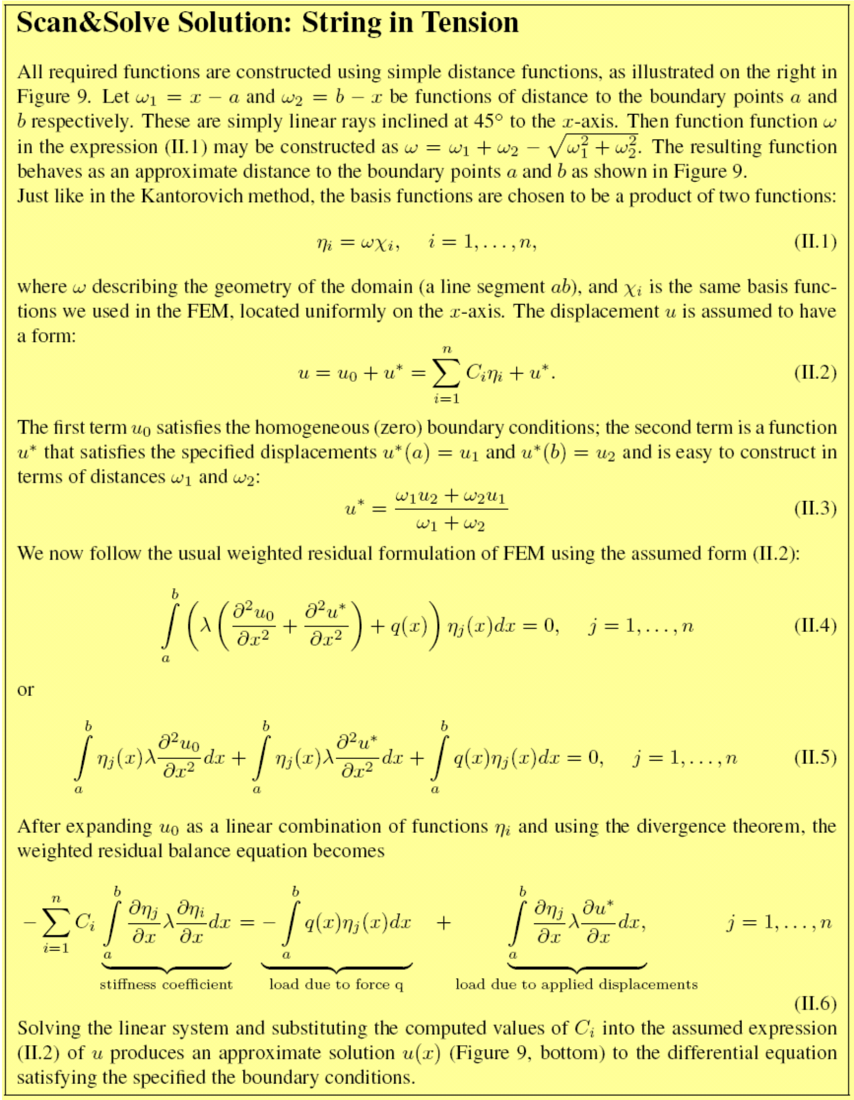
with $j = 2, \ldots, n-1$. Solving the linear system and substituting the computed values of C_i into the assumed expression (I.1) of u produces an approximate solution $u(x)$ to the differential equation satisfying the specified boundary conditions.

Figure 6: FEM solution of the 1D string problem

Figure 7: Finite element solution of the 1D string in tension problem.



Solution of the same problem using the Scan&Solve™ method is shown in Figures 8 and 9. Instead of meshing the domain (line segment ab), a uniform grid is associated with the x-axis, and a basis function is associated with each grid point. Notice that in this case the basis functions are the same as in the FEM formulation, but the nodes of the mesh no longer coincide with the boundary points a and b; in other words, the grid and these initial basis functions do not know where the boundary of the domain is. Following the Kantorovich method, the new basis functions are created as the product of the function ω and the FEM basis functions (Figure 9). These functions take on zero values at the end points a and b. Using the distances to a and b, we construct a single function, u*, that interpolates the prescribed values u_1 and u_2 at the end points a and b respectively [8]. When we add u* to the linear combination of the newly constructed basis functions, we obtain a representation for displacement that satisfies the fixed boundary conditions at the end points of the string, even though the grid points do not coincide with a or b! We now repeat the usual weighted residual formulation, ending in Equation (II.6) which is structurally very similar to the FEM equation (I.5) in Figure 6. One significant difference between the two equations is the last term because Scan&Solve™ enforces the boundary conditions at every point on the boundary, whether they coincide with the nodes of the mesh or not. The other terms are essentially the same as in FEM, and the stiffness matrix is still sparse, but evaluation of the integrals is more complicated because the domain (line segment) is not meshed and the integrands include the distance functions.



Scan&Solve Solution: String in Tension

All required functions are constructed using simple distance functions, as illustrated on the right in Figure 9. Let $\omega_1 = x - a$ and $\omega_2 = b - x$ be functions of distance to the boundary points a and b respectively. These are simply linear rays inclined at 45° to the x-axis. Then function function ω in the expression (II.1) may be constructed as $\omega = \omega_1 + \omega_2 - \sqrt{\omega_1^2 + \omega_2^2}$. The resulting function behaves as an approximate distance to the boundary points a and b as shown in Figure 9.
Just like in the Kantorovich method, the basis functions are chosen to be a product of two functions:

$$\eta_i = \omega\chi_i, \quad i = 1, \ldots, n, \tag{II.1}$$

where ω describing the geometry of the domain (a line segment ab), and χ_i is the same basis functions we used in the FEM, located uniformly on the x-axis. The displacement u is assumed to have a form:

$$u = u_0 + u^* = \sum_{i=1}^{n} C_i\eta_i + u^*. \tag{II.2}$$

The first term u_0 satisfies the homogeneous (zero) boundary conditions; the second term is a function u^* that satisfies the specified displacements $u^*(a) = u_1$ and $u^*(b) = u_2$ and is easy to construct in terms of distances ω_1 and ω_2:

$$u^* = \frac{\omega_1 u_2 + \omega_2 u_1}{\omega_1 + \omega_2} \tag{II.3}$$

We now follow the usual weighted residual formulation of FEM using the assumed form (II.2):

$$\int_a^b \left(\lambda\left(\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u^*}{\partial x^2}\right) + q(x)\right)\eta_j(x)dx = 0, \quad j = 1, \ldots, n \tag{II.4}$$

or

$$\int_a^b \eta_j(x)\lambda\frac{\partial^2 u_0}{\partial x^2}dx + \int_a^b \eta_j(x)\lambda\frac{\partial^2 u^*}{\partial x^2}dx + \int_a^b q(x)\eta_j(x)dx = 0, \quad j = 1, \ldots, n \tag{II.5}$$

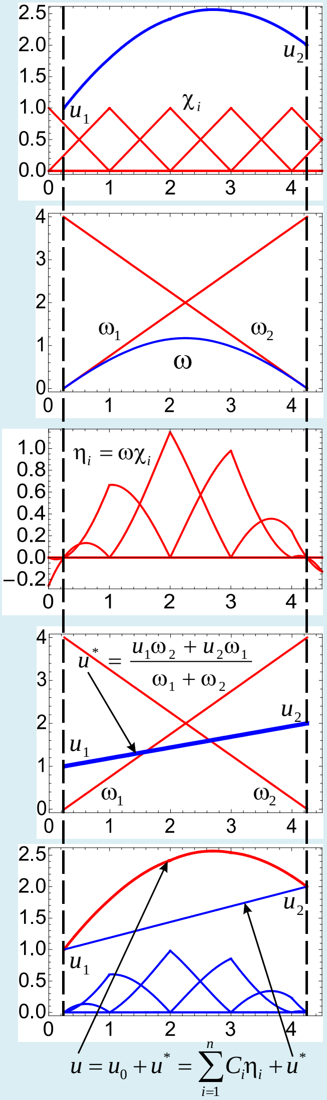
After expanding u_0 as a linear combination of functions η_i and using the divergence theorem, the weighted residual balance equation becomes

$$-\sum_{i=1}^{n} C_i \underbrace{\int_a^b \frac{\partial\eta_j}{\partial x}\lambda\frac{\partial\eta_i}{\partial x}dx}_{\text{stiffness coefficient}} = \underbrace{-\int_a^b q(x)\eta_j(x)dx}_{\text{load due to force q}} + \underbrace{\int_a^b \frac{\partial\eta_j}{\partial x}\lambda\frac{\partial u^*}{\partial x}dx}_{\text{load due to applied displacements}}, \quad j = 1, \ldots, n \tag{II.6}$$

Solving the linear system and substituting the computed values of C_i into the assumed expression (II.2) of u produces an approximate solution $u(x)$ (Figure 9, bottom) to the differential equation satisfying the specified the boundary conditions.

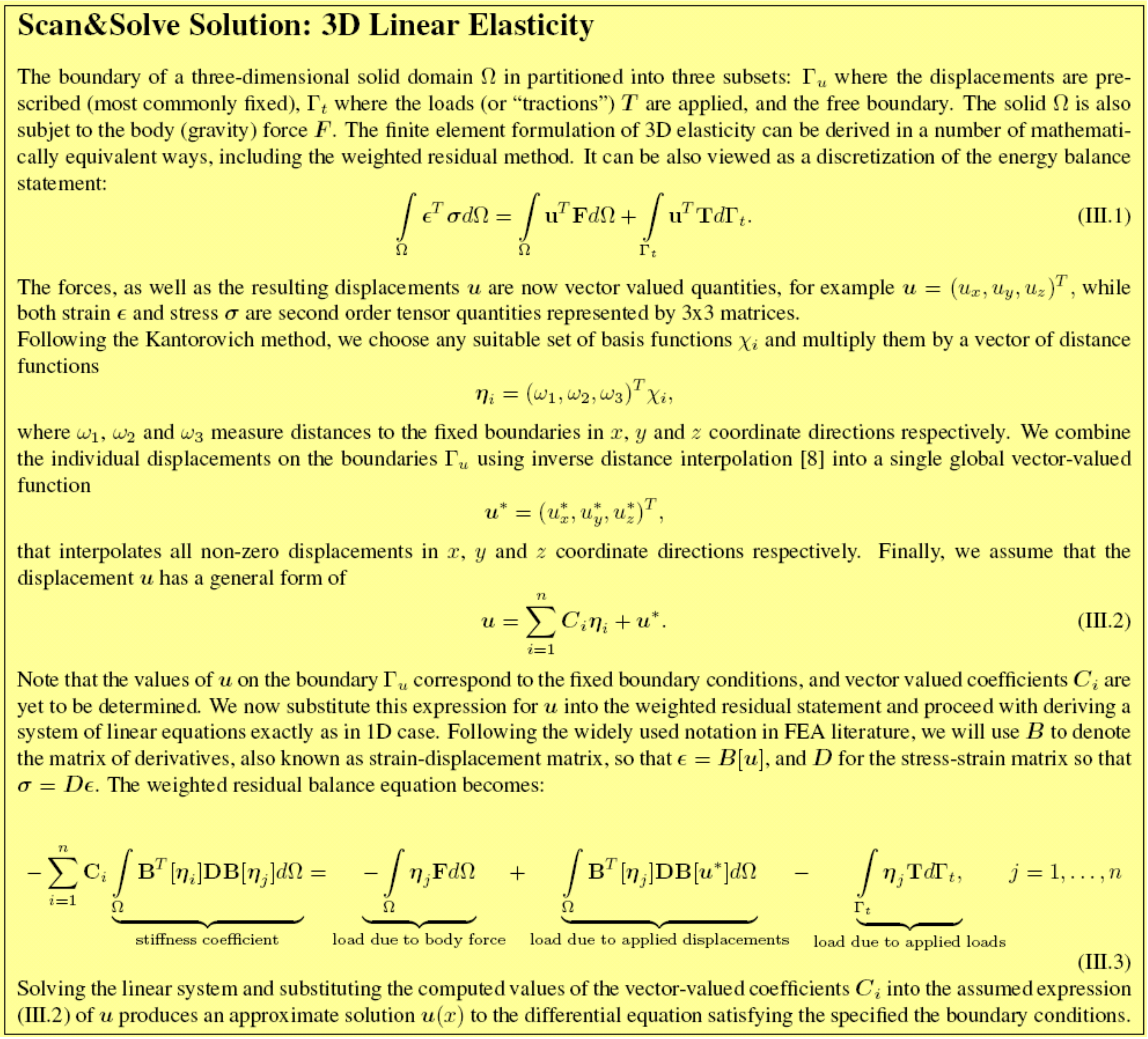
Figure 8: Scan&Solve™ solution of the 1D string problem.

Figure 9: Scan&Solve™ solution of the 1D string in tension problem



3D Linear Structural Analysis with Scan&Solve™

The preceding 1D example highlights the essential similarities and differences between FEM and Scan&Solve™. Of course, meshing a line segment is trivial, and the added complexity of Scan&Solve™ may appear to defeat the purpose. However, the Scan&Solve™ approach applies to any and all problems in engineering analysis and in any number of dimensions. The formulation of linear elastic structural analysis is summarized in Figure 10 and illustrated with a simple example in Figure 11.



Scan&Solve Solution: 3D Linear Elasticity

The boundary of a three-dimensional solid domain Ω in partitioned into three subsets: Γ_u where the displacements are prescribed (most commonly fixed), Γ_t where the loads (or "tractions") T are applied, and the free boundary. The solid Ω is also subjet to the body (gravity) force F. The finite element formulation of 3D elasticity can be derived in a number of mathematically equivalent ways, including the weighted residual method. It can be also viewed as a discretization of the energy balance statement:

$$\int_\Omega \epsilon^T \sigma d\Omega = \int_\Omega \mathbf{u}^T \mathbf{F} d\Omega + \int_{\Gamma_t} \mathbf{u}^T \mathbf{T} d\Gamma_t. \tag{III.1}$$

The forces, as well as the resulting displacements u are now vector valued quantities, for example $u = (u_x, u_y, u_z)^T$, while both strain ϵ and stress σ are second order tensor quantities represented by 3x3 matrices.
Following the Kantorovich method, we choose any suitable set of basis functions χ_i and multiply them by a vector of distance functions

$$\eta_i = (\omega_1, \omega_2, \omega_3)^T \chi_i,$$

where ω_1, ω_2 and ω_3 measure distances to the fixed boundaries in x, y and z coordinate directions respectively. We combine the individual displacements on the boundaries Γ_u using inverse distance interpolation [8] into a single global vector-valued function

$$u^* = (u_x^*, u_y^*, u_z^*)^T,$$

that interpolates all non-zero displacements in x, y and z coordinate directions respectively. Finally, we assume that the displacement u has a general form of

$$u = \sum_{i=1}^{n} C_i \eta_i + u^*. \tag{III.2}$$

Note that the values of u on the boundary Γ_u correspond to the fixed boundary conditions, and vector valued coefficients C_i are yet to be determined. We now substitute this expression for u into the weighted residual statement and proceed with deriving a system of linear equations exactly as in 1D case. Following the widely used notation in FEA literature, we will use B to denote the matrix of derivatives, also known as strain-displacement matrix, so that $\epsilon = B[u]$, and D for the stress-strain matrix so that $\sigma = D\epsilon$. The weighted residual balance equation becomes:

$$-\sum_{i=1}^{n} \mathbf{C}_i \underbrace{\int_\Omega \mathbf{B}^T[\eta_i]\mathbf{DB}[\eta_j]d\Omega}_{\text{stiffness coefficient}} = \underbrace{-\int_\Omega \eta_j \mathbf{F} d\Omega}_{\text{load due to body force}} + \underbrace{\int_\Omega \mathbf{B}^T[\eta_j]\mathbf{DB}[u^*]d\Omega}_{\text{load due to applied displacements}} - \underbrace{\int_{\Gamma_t} \eta_j \mathbf{T} d\Gamma_t,}_{\text{load due to applied loads}} \qquad j = 1, \ldots, n \tag{III.3}$$

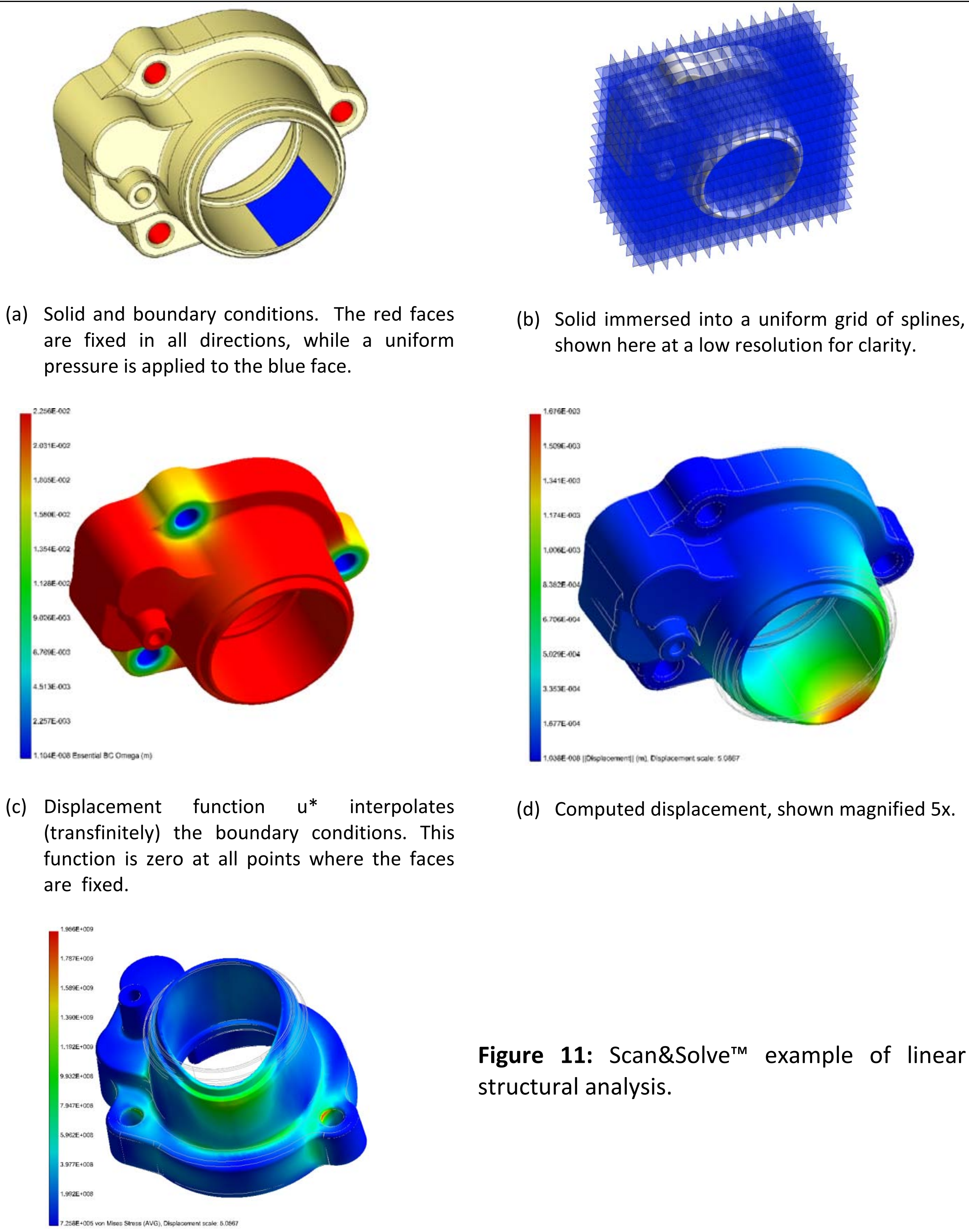
Solving the linear system and substituting the computed values of the vector-valued coefficients C_i into the assumed expression (III.2) of u produces an approximate solution $u(x)$ to the differential equation satisfying the specified the boundary conditions.

Figure 10: Scan&Solve™ formulation for linear elastic structural analysis.

The full 3D problem of static elasticity is a straightforward generalization of the 1D string in tension problem, except all relevant quantities (displacements, forces, basis functions, and coefficients) are now vector valued (with components in each of the three coordinates x, y, z). Thus, the one dimensional string in Figure 5 becomes a three-dimensional solid Ω with a two-dimensional boundary Γ, part of which is fixed and part of which is loaded by external forces (Figure 11(a)). The tension λ is generalized to the stress-strain matrix **D**, and partial differentiation of the basis function is replaced by application of the strain-displacement matrix **B** commonly used in FEA literature. The solid is not meshed, but it is immersed into pre-meshed space, as shown in Figure 11(b). In this case, the mesh is uniform but it does not have to be. A basis function, χ_i, commonly a three-dimensional B-spline, is associated with every node of the mesh. Using distances to the fixed portions of the boundaries, a global function u* is constructed to interpolate all prescribed displacements as shown in Figure 11(c). The interpolation procedure is a straightforward generalization of expression (II.3) used to interpolate the displacements indicated at the ends of the string in tension [8].

The resulting system of linear equations (III.3) represents a force-balance equation for a 3D solid and is structurally very similar to the system (II.6) for a string in tension. In fact, there is a one-to-one correspondence between the integral terms, except for the last integral in (III.3) which represents the loads due to traction forces applied over the boundary of the solid. The corresponding term is missing in the equations (II.6) for the string, only because no boundary forces were indicated in the string problem formulation. Solving the resulting system of linear equations for the unknown coefficients C_i and substituting the computed values of the coefficients into expression (III.2) immediately solves the displacement problem as shown in Figure 11(d). The stresses (Figure 11(e)) are recovered from displacement through application of the usual relationship σ=**DB**[u] in any FEA procedure.

It should now be clear that Scan&Solve™ belongs to the same (variational) class of numerical solutions as classical FEA, but it eliminates the need for meshing the domain, Ω, in favor of additional computations performed on the native geometric representation. These computations are used to assemble and solve the system of equations (III.3) and are at the very heart of the Scan&Solve™ technology. The next section will discuss them in more detail.



(a) Solid and boundary conditions. The red faces are fixed in all directions, while a uniform pressure is applied to the blue face.

(b) Solid immersed into a uniform grid of splines, shown here at a low resolution for clarity.

(c) Displacement function u* interpolates (transfinitely) the boundary conditions. This function is zero at all points where the faces are fixed.

(d) Computed displacement, shown magnified 5x.

(e) Computed stresses.

Figure 11: Scan&Solve™ example of linear structural analysis.

3. System design and operation

As can be seen from its formulation and equation (III.3), mathematically Scan&Solve™ is very similar to other FEA approaches, but its implementation differs substantially from most FEM codes because it places minimal requirements on the geometric model and the user. At no time is the model required to be meshed or converted into another representation, and at no time is the user constrained by the mesh or required to interact with it. The power and flexibility of Scan&Solve™ is achieved through a number of modern computational techniques developed by Intact Solutions over the last decade. The high level schematic description of the Scan&Solve™ is shown in Figure 12.

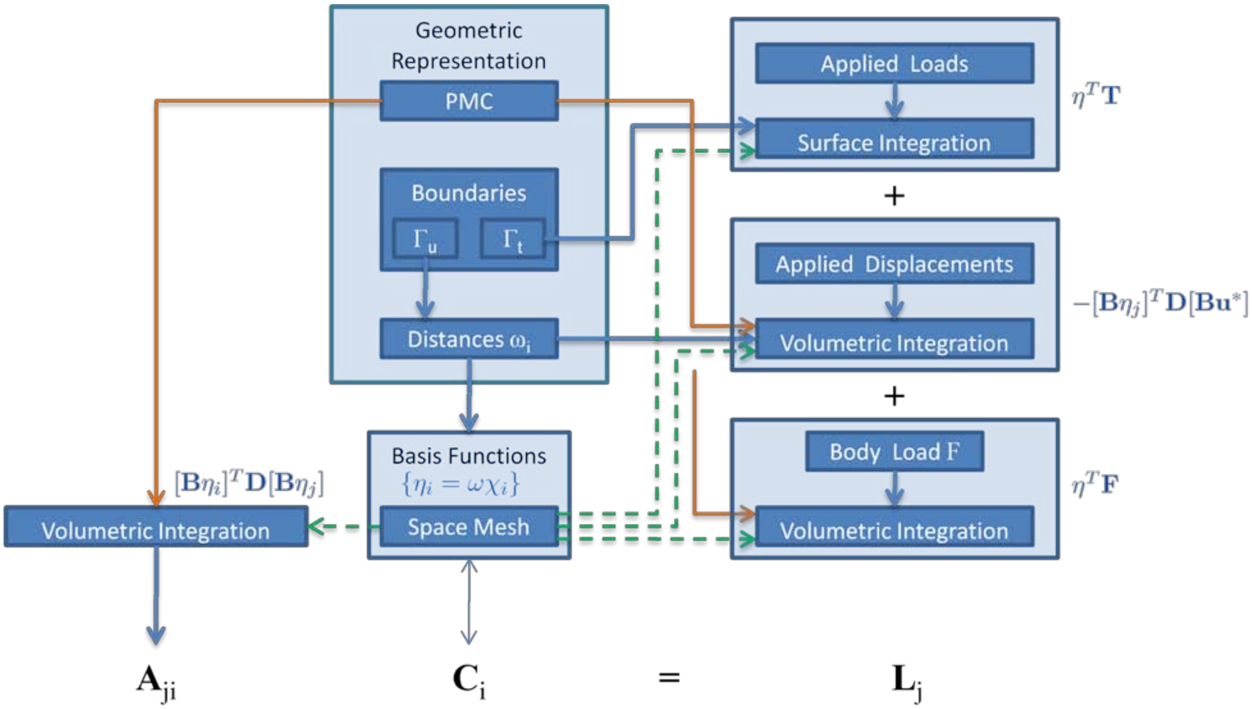


Figure 12. Schematic architecture of the Scan&Solve™ system

The inputs to the system are identical to those expected in any engineering analysis system. They must include some unambiguous geometric representation of the domain (solid) and its boundary. Portions of the boundary Γ_u are fixed or restrained by applied displacements, and external loads are applied to other portions of the boundary Γ_t. Gravity or other accelerations may or may not be assumed to act on the solid as a body force. The Scan&Solve™ system does the rest, including fully automatic generation of the linear system of equations, solving for the displacements, and computing stresses inside the solid and its boundary. The sequence of computations is roughly as follows.

A space grid of basis functions χ_i (usually uniform B-splines) is automatically created by the system. The user can (but does not have to) indicate the resolution of the grid, its non-uniformity, or choose a different type of basis functions. Initially these basis functions are not aware of the solid's boundaries. Next the system computes a collection of approximate distance fields, referred to as ω_i, using a proprietary technique to rapidly evaluate Euclidean distance to the portions of the domain boundary that are restrained. Since distance is a property intrinsic to geometry, this can be done without conversion to other representations, or approximation by a finite element mesh. The computed distance fields are used to modify the initial set of basis functions to produce a new collection of basis functions η_i that now enforce fixed displacement conditions at the indicated boundaries. When non-zero displacements are indicated, they are interpolated into a single function u*, again using the constructed distances [8].

At this point, Scan&Solve™ has all the information it needs to complete the solution process. The coefficients of the basis functions η_i will serve as degrees of freedom that are determined by solving a linear system of algebraic equations assembled by the Scan&Solve™ process. The assembly process requires repeated numerical evaluation of the four integrals in equation (III.3) that are easily identifiable in the schematic diagram of Figure 12. Two types of integrals must be computed: volume integrals over the interior of the geometric model and surface integrals over portions of the boundary where loads are applied. Numerical integration is a simple matter of summing weighted samples of a function taken at specified quadrature (integration) points. Unlike finite element analysis, these quadrature points must be allocated at run time since the geometric domain does not conform to the spatial grid of the basis functions. Both types of integration allocate these quadrature points using operations provided by all mainstream CAD systems. To accommodate volume integration, the geometric representation needs only answer whether a point supplied by the integration algorithm is inside or outside the domain [6]. This operation is known as point membership classification or PMC, and is fundamental to all geometric modeling CAD systems. Surface integration requires only polygonization of that portion of the boundary where the loads are applied. Surface polygonization is also a key component of CAD systems that provide a visual display or output for rapid prototyping.

The result from the integration processes is a system of linear equations: a sparse matrix **A** whose elements are the usual stiffness coefficients and a vector **L** containing the contributions of the loads and restraints. This system of linear equations is solved using standard linear solver tools to yield the coefficients $\mathbf{C}_i$ of the basis functions η_i in the solution represented in the form of equation (III.2). Substitution of these coefficients allows the solution to be evaluated and differentiated [12] at any point to determine physical quantities of interest (e.g. displacement, strain, stress) at any point from the domain. The solution may also be sampled and visualized on the surface of the solid using standard post-processing techniques, including surface polygonization and graphics hardware.

To summarize, Scan&Solve™ preserves most computational advantages of FEA, but avoids any meshing by reducing all computations to a well-defined sequence of simple and efficient computations: point and function sampling, distance to boundary computation, and solution of sparse system of linear equations.

4. Experimental results

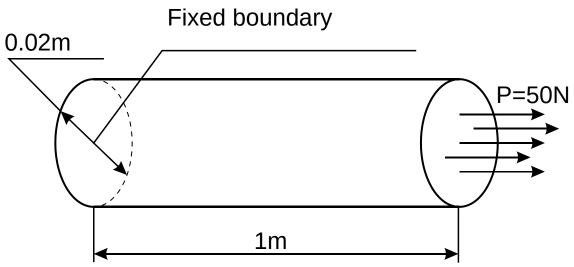
In this section we validate the Scan&Solve™ approach by a few carefully selected numerical experiments and demonstrate its power on several non-trivial real world example.

Comparison with known analytic and experimental solutions

Only a few very simple problems in linear elasticity admit exact analytic solutions, and only a few experimental solutions have been obtained over the years. We will rely on three such known solutions to provide experimental evidence for correctness and consistency of the results computed by Scan&Solve™.

Experiment 1: linear displacements and constant stresses

In a simple experiment, we use Scan&Solve™ to simulate the basic tensile stress test. The purpose of this experiment is to validate that linear displacements and constant stresses can be accurately computed by Scan&Solve^{TM}. One can think of this experiment as an equivalent of a `patch test' used in validating mesh-based FEA solutions. The setup of this numerical experiment is shown in Figure 13. Plots in Figure 14 show the computed distributions of the components of the displacement vector and von Mises stress obtained by Scan&Solve^{TM}, confirming that the components of the displacement vector *u*, *v* and *w* are linear functions of the corresponding coordinates, and that the value of von Mises stress is constant in every cross section of the rod. Figure 15(a) shows the dependence of the computed solutions on the size Δ of the support of the basis function, indicating that the solution converged, and that the value of the von Mises stress predicted by Scan&Solve^{TM} is in very good agreement (the relative error ε is less than 0.01%) with the theoretical value of the tensile stress (Figure 15(b)).



Material properties

E = 68.95 GPa
ν = 0.33

Tensile stress $$\sigma = \frac{4P}{\pi D^2} = \frac{4 \cdot 50}{\pi \cdot 0.02^2} \approx 159154.943\ (\text{Pa})$$

Figure 13: Experiment 1: simulating the tensile test.

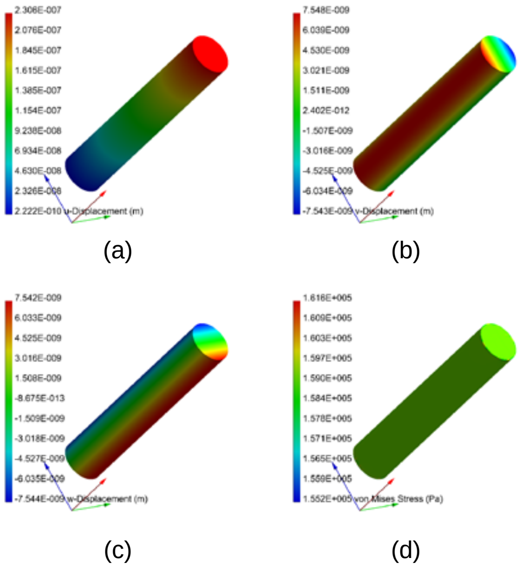
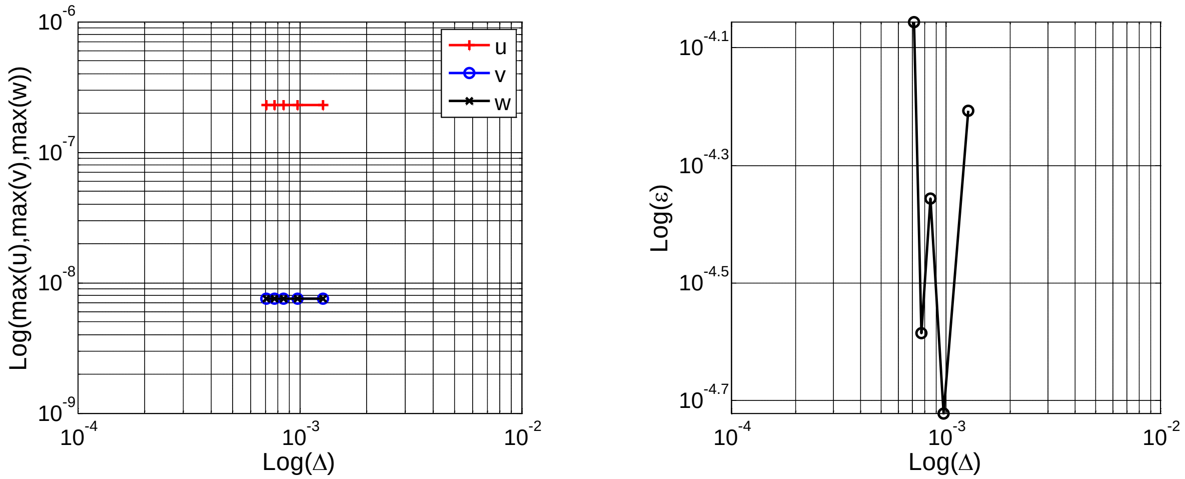


Figure 14: Linear displacements and constant stresses. Plots in (a,b,c) show that the components *u*, *v* and *w* of the displacement vector are linear functions of the corresponding coordinates; (d) von Mises stress is constant in every cross section of the rod.



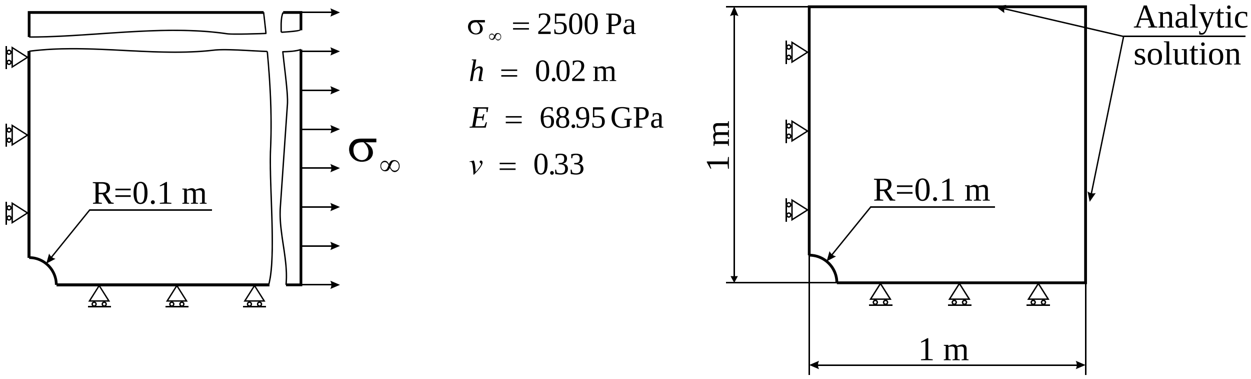
(a) Convergence of the maximum values of the components of the displacement vector;

(b) Dependence of the relative error ε of von Mises stress with respect to the element size Δ.

Figure 15: Experiment 1: Convergence studies.

Experiment 2: comparison with analytic solution.

The second numerical experiment compares the Scan&Solve^{TM} solution with the known analytic solution for a loaded infinite 2D plate shown in Figure 16(a). A close 3D finite approximation of this problem was created in Scan&Solve™ system using thin but finite 3D plate (2m x 2m, 0.02m thick), and by applying loading conditions to the finite boundaries of the plate that match exactly to the known analytical solution. We then solve for displacements and stresses inside the thin solid and compare results to the analytic solution. Plots in Figure 17 demonstrate exponential convergence of the displacement and von Mises stress obtained by Scan&Solve^{TM} as the element size Δ decreases. Figure 17 also illustrates that Scan&Solve^{TM} solutions are in a good agreement with analytic results: error of the displacement does not exceed 0.1% and von Mises stress is within 1% of the analytically predicted value.



$$\sigma_\infty = 2500\ \text{Pa}$$
$$h = 0.02\ \text{m}$$
$$E = 68.95\ \text{GPa}$$
$$v = 0.33$$

(a) Loaded infinite plate.

(b) Setup of a numerical experiment for a finite 3D plate that simulates the problem in (a).

Figure 16: Experiment 2: Simulating analytic solution.

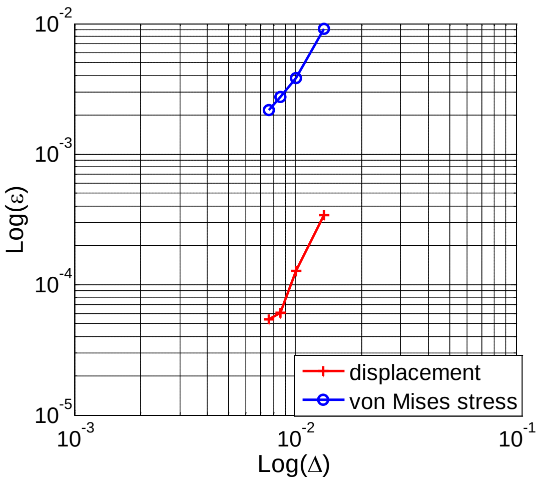


Figure 17: Scan&Solve™ solution vs. analytic solution for Experiment 2: exponential convergence of the displacement and von Mises stress is demonstrated using L_2 norm and relative error ε.

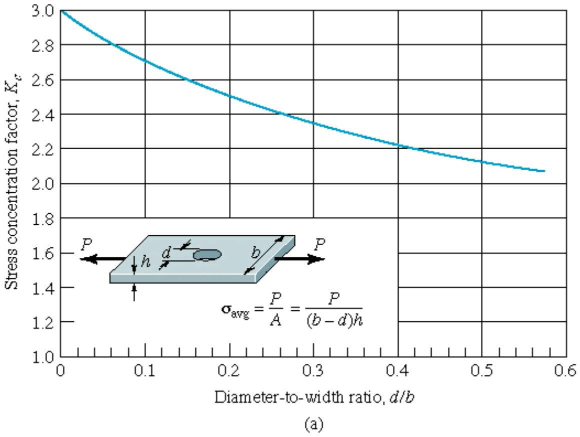


Figure 18: Experiment 3: Stress concentration factor for rectangular plate with central hole. [Adapted from J.A.Collins, Failure of Materials in Mechanical Design,1981.]

Experiment 3: Comparison with empirical data.

The final numerical experiment compares the stress concentration factor computed by Scan&Solve^{TM} with the widely accepted and empirically determined value (Figure 18). Geometry of the plate and loading conditions are shown in Figure 19. The empirical data predicts the stress concentration factor of 2.5, which is defined as the ratio of the max stress at the boundary of the hole to the average stress over the end of the plate (see [4] for more details). Figures 20 and 21 show the results computed by Scan&Solve^{TM}. Figure 20 indicates that the maximum values of the components of the displacement field converged and do no change as the element size decreases. As expected, the plots in Figures 21(a) and (b) also demonstrate good agreement between empirical data and the stress concentration factor computed by Scan&Solve^{TM}. The relative error in the stress concentration factor is less than 1%.

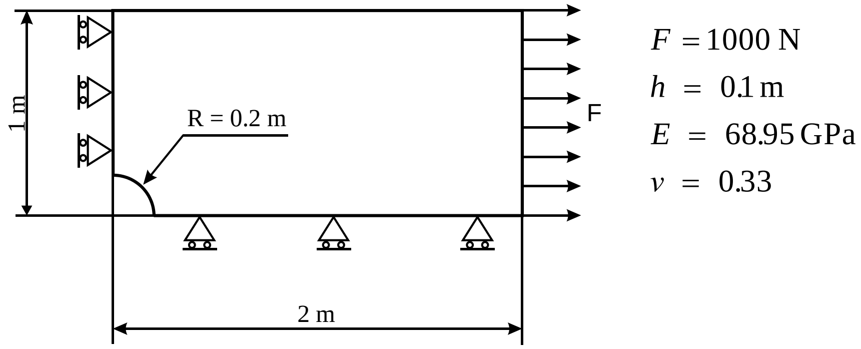


Figure 19: Geometry of the plate and loading conditions for Experiment 3.

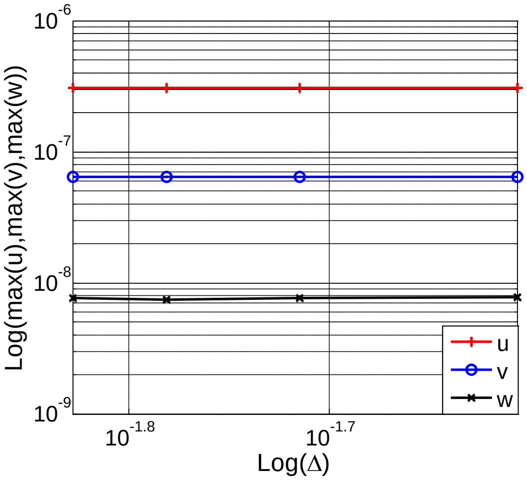
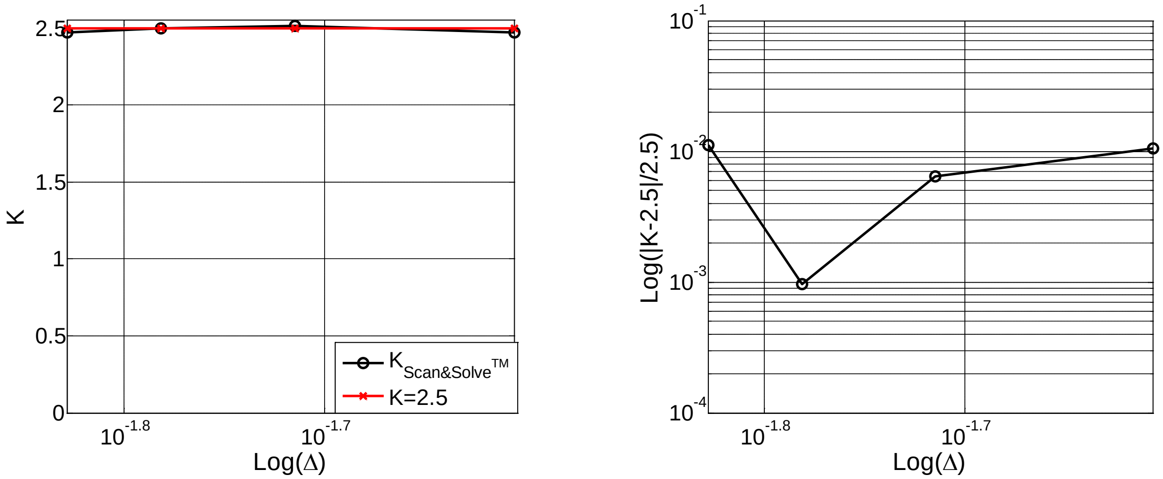


Figure 20: Experiment 3: convergence of the maximum values of the displacement components as the element size Δ decreases.



(a) Stress concentration factors reported by Scan&Solve^{TM} are in good agreement with the experimental data in Figure 18.

(b) Relative error of stress concentration factors is less than 1%.

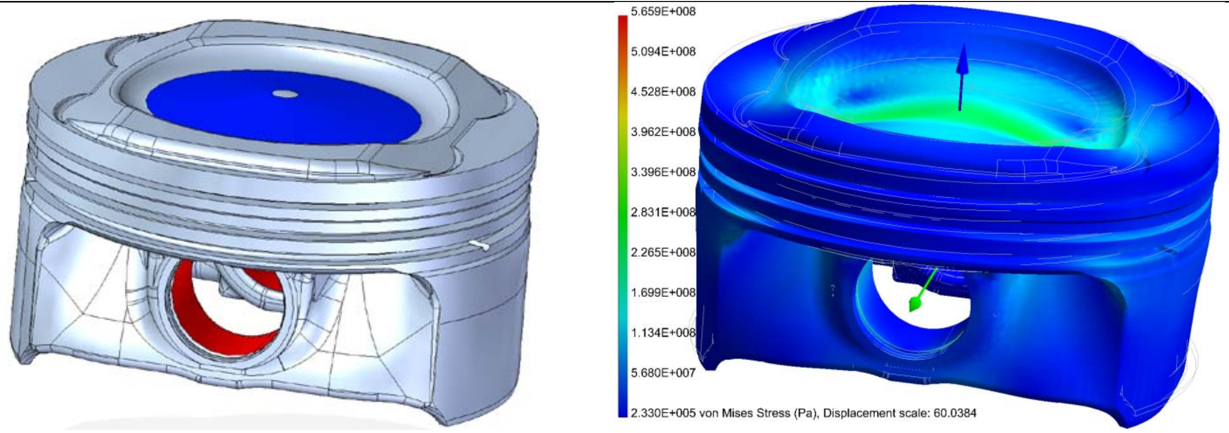
Figure 21: Experiment 3: Comparison with empirical data.

4.2 Real world examples

Scan&Solve™ is rooted in sound mathematical foundations, and the above experiments confirm that it performs as expected, computing correct answers in carefully controlled experiments. These experiments do little to demonstrate the claimed advantages of Scan&Solve, particularly when it comes to handling complex real-world geometry. We conclude this section with three examples (introduced in Figure 1) chosen to highlight the power, flexibility, robustness of the technology.

Example 1: Native CAD model; no defeaturing or simplification

Figure 22(a) shows a solid model of a piston, with indicated boundary conditions, and Figure 22(b) shows the displacement field computed by Scan&Solve™. Despite the large number of small features in this model (holes, blends, fillets), Scan&Solve™ can use any desired resolution for stress analysis. In this case, with 60,000 basis functions, the solution is computed in 3 minutes 10 seconds on a midrange Pentium class computer. The solution procedure is fully automatic, no geometric features have been removed or simplified, and no preprocessing is required to change the resolution of stress analysis.

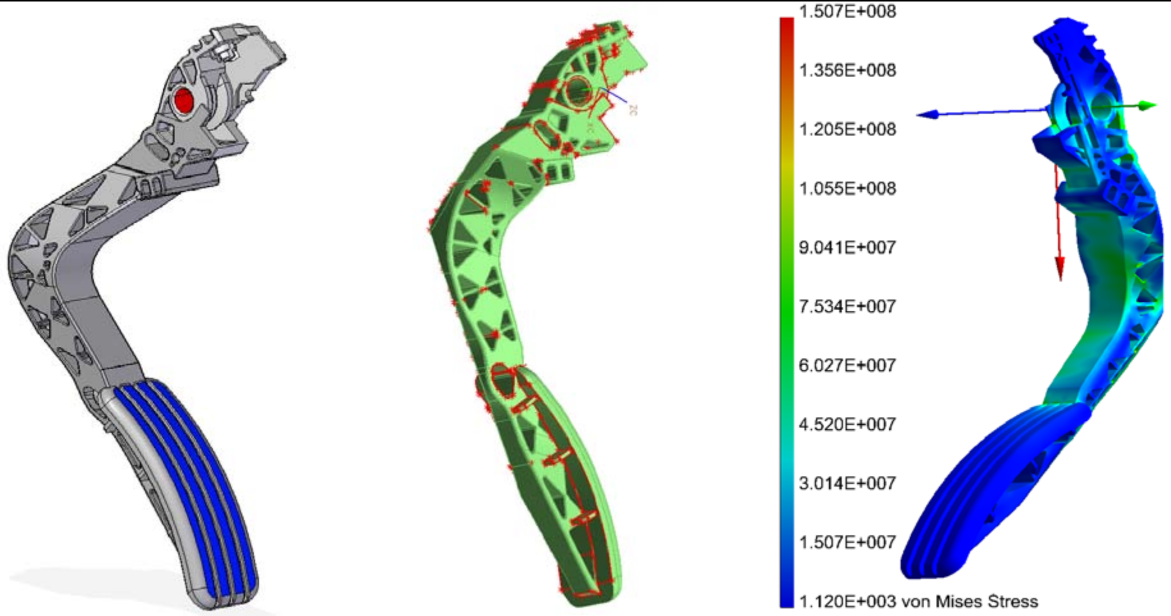


(a) (b)

Figure 22: Scan&Solve™ solution for a complex piston geometry. (a) The boundary conditions. The red faces are fixed, while the blue face has a pressure applied. (b) The computed von Mises stress distribution over the piston with the deflection magnified 60x.

Example 2: Imported CAD model; no need for healing and repairs

Figure 23(a) shows another example. In this case, the solid model of an automotive pedal is subjected to loads as shown. Because the CAD model was translated between CAD systems, the routine quality check of the geometric model reveals a large number (more than a hundred) of small errors and inconsistencies in the pedal’s boundary representation, including tiny edges and surfaces, overlapping and intersecting faces, mismatching tangents, and so on. These errors, highlighted in Figure 23(b), are likely to cause problems in most mesh-based FEM systems, but not in Scan&Solve, because it relies only on the relatively simple, robust and stable geometric computations that are supported in most CAD systems even in the presence of the small errors. If a model is good enough for point membership and distance computations, then it is also good enough for Scan&Solve™. Figure 23(c) shows the displacement solution for the model in Figure 23(a). With 58,000 basis functions, the total computation time is less than 6 minutes.

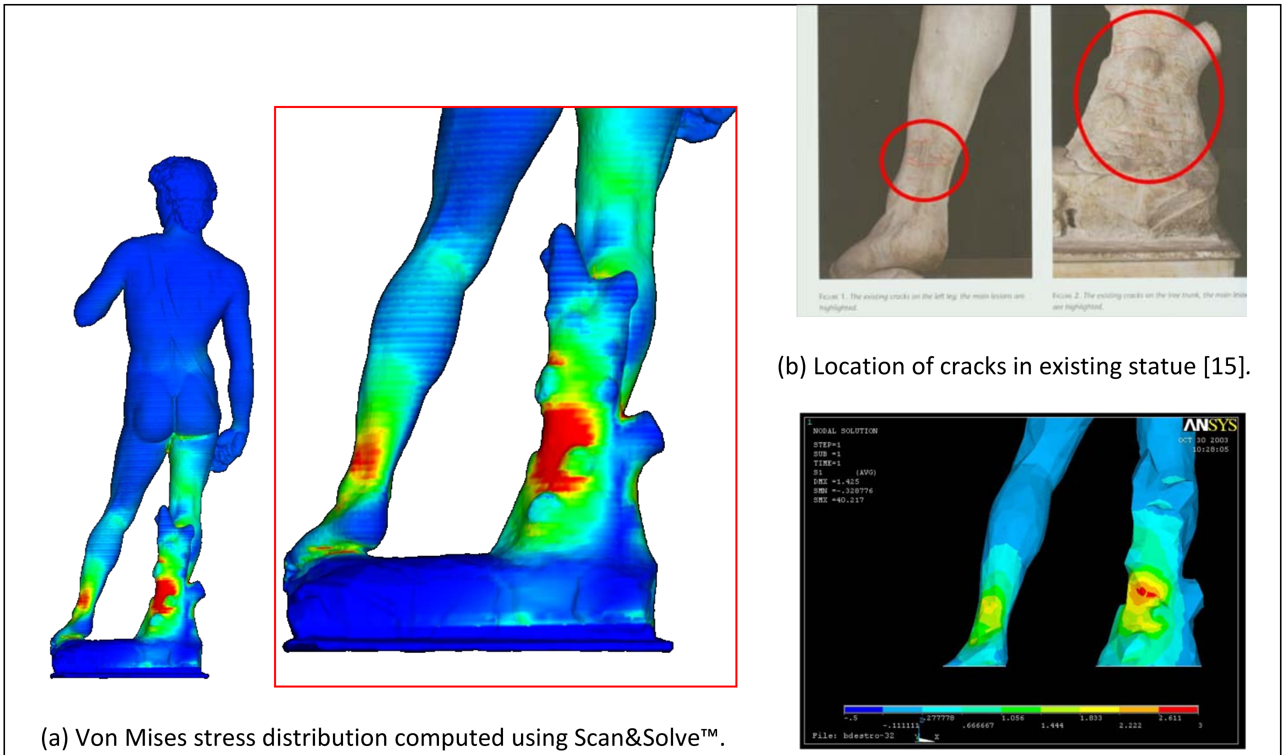


(a) (b) (c)

Figure 23: Scan&Solve stress analysis of automotive pedal. (a) The boundary conditions. The red faces are fixed, and a 600N load is distributed over the blue faces. (b) The geometric errors and inconsistencies that prevent meshing. (c) The computed von Mises stress within the pedal.

Example 3: Native geometry, any representation, any resolution

Our final example was recently profiled by the National Science Foundation and in many widely circulated publications [13, 14]. Figure 24 (a) shows the first principal stress for the statue of the Michelangelo's David under gravity, computed using Scan&Solve™. The results confirmed that high stresses in the area shown arose when David's orientation was 3° off vertical. Furthermore, these results correlate well with both earlier computed FEM solutions and experimentally observed cracks (Figure 24 (b)). In this case, Scan&Solve™ computed the displacements and stresses directly from the triangulated model provided by the Digital Michelangelo Project at Stanford University. Minimal automated preprocessing was needed to create a valid model supporting point and distance queries. No simplification, smoothing, or meshing was required. It may be instructive to compare the computed solution with the previously published solution using FEM based on manually smoothed and simplified model (Figure 24 (c))



(b) Location of cracks in existing statue [15].

(a) Von Mises stress distribution computed using Scan&Solve™.

(c) First principal stress distribution computed using FEA [15].

Figure 24: Stress analysis on Michelangelo's David.

5. Conclusions

> *"It is unworthy of excellent men to lose hours, like slaves, in labors of calculation which could be safely relegated to anyone else if machines were used. "*
>
> Gottfried Wilhelm Leibnitz

Finite Element Analysis revolutionized engineering by enabling analysis and simulation to be performed by analysts for simple and complex problems that could not be solved by hand, and at speeds that are not possible without use of computers. With Scan&Solve™ meshfree technology, finite element analysis enters a new era where the same powerful tools of analysis and simulation can be used by non-specialists and specialists alike, without any preprocessing, with automation and speed that extend the reach of FEA throughout engineering and beyond, including art, medicine, military, and consumer applications. The discussion in this paper was confined to linear stress analysis, but the concepts and methods of Scan&Solve™ apply to any and all analysis problems. We conclude by briefly summarizing some of the more exciting applications of Scan&Solve™.

Analysis in Situ

Scan&Solve™ can be integrated with any CAD or geometric modeling system that supports the required queries: point membership classification (PMC) and distance to boundary computation. This means that the models no longer need to be massaged, simplified, healed, and translated. The engineering analysis can be performed directly on the native representation and within the familiar representational framework, whether this model is a boundary representation, STL file, volumetric representation, constructive solid geometry, or a polygonized model.

Analysis fast

Despite the perceived computational overhead, performance of Scan&Solve™ is already approaching that of commercial FEA systems, and is likely to exceed them on models containing small features, errors, and noisy boundaries. However, the deterministic simplicity of the Scan&Solve™ computational pipeline virtually guarantees additional dramatic improvements via optimized algorithms, improved hardware, and the ongoing shift towards increasingly parallel and multithreaded computing (e.g. using GPU).

Analysis on demand, anywhere

Full automation of engineering analysis challenges the established economic assumptions and introduces the possibility of analysis and simulation in situations deemed impractical only a few years ago. Geometric models and boundary conditions, as well as automatically computed solutions, may be easily transmitted over Internet suggesting that automated analysis over Internet is now a reality. Any digital model that can be manufactured (by traditional methods or by 3D printing) may and should also be analyzed routinely for its physical performance.

Analysis of scans, images and other acquired models

A wealth of digital data has been acquired over the last several decades, but by all indications, digital content creation is still in its infancy. Today most of this data is used for archiving, visual, or geometric planning purposes; physical analysis of such models is expensive and impractical on a large scale because it requires tedious, error-prone, manual data preparation. But point membership queries and distance computations may be performed on all models, including those scanned, imaged, digitized, or sensed, enabling direct application of Scan&Solve™ analysis. Whether the models are dental scans, CT scans of bones, 3D reconstructions from laser-scanned data or photogrammetry, or reverse engineered models – they can now be analyzed with Scan&Solve™ easily and economically.

Seamless integration of design and analysis

Integration of engineering design and analysis has remained an elusive goal, not in small part due to meshing and data model translations that introduce discrepancies between design and analysis models, thus breaking the design-analysis cycle. With Scan&Solve™ seamless integration of design and analysis has become a reality. The design and analysis models are always consistent, allowing repeated analysis of parametric and free-form changes in geometry, shape and topology optimization [3], as well as effortless (and even simultaneous) “what if” studies.

Accuracy and errors

Recall that with Scan&Solve™ geometric control of accuracy is separated from the analysis accuracy. An important consequence of this separation is that one can perform rough/precise analysis on coarse/accurate geometric models, allowing for both conceptual studies and detailed analysis within the same computational framework. In particular, analysis of geometrically imprecise models may lead to more realistic results in engineering analysis, quantification of these results in terms of geometric errors and uncertainty, and application of engineering analysis to tolerance, sketched, and conceptual models.

For more information on Scan&Solve™ technology and products, please contact us at

Intact Solutions, LLC
www.meshfree.com
info@intact-solutions.com

References

1. J.E. Akin, *Finite Element Analysis with Error Estimators*, Elsevier, 2005
2. I. Babuška, U. Banerjee, J. Osborn, Survey of meshless and generalized finite element methods: A unified approach, in *Acta Numerica*, Vol. 12, 2003, Cambridge University Press, pp. 1–125.
3. J. Chen, V. Shapiro, K. Suresh, I. Tsukanov, Shape Optimization with Topological Changes and Parametric Control, *International Journal for Numerical Methods in Engineering*, Vol.71(3), 2007, pp.313-346
4. J.A. Collins, *Failure of Materials in Mechanical Design*, Wiley, New York, 1981
5. L.V. Kantorovich, V.I.Krylov, *Approximate Methods of Higher Analysis*, Interscience Publishers, 1958
6. B. Luft, V. Shapiro, I. Tsukanov, Geometrically Adaptive Numerical Integration, *Proceedings of 2008 ACM Symposium on Solid and Physical Modeling,* Stony Brook, NY, June 2008, pp. 147-157
7. V.L. Rvachev, *Theory of R-functions and Some Applications*, Naukova Dumka, 1982, (In Russian)
8. V.L. Rvachev, T.I. Sheiko, V. Shapiro, I. Tsukanov, Transfinite Interpolation Over Implicitly Defined Sets, *Computer Aided Geometric Design*, Vol. 18(3), 2001, pp.195-22
9. V.L. Rvachev, T.I. Sheiko, V. Shapiro, I. Tsukanov, On Completeness of RFM Solution Structures, *Computational Mechanics*, special issue on meshfree methods, Vol. 25(2-3), 2000, pp.305-316
10. V. Shapiro, Semi-analytic geometry with R-functions, in *Acta Numerica,* 2007, Cambridge University Press pp. 239-303
11. V. Shapiro, I. Tsukanov, Implicit Functions with Guaranteed Differential Properties, *In Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications*, June 1999, Ann Arbor, MI, pp.258-269
12. I. Tsukanov, M. Hall, Data Structure and Algorithms for Fast Automatic Differentiation, *International Journal for Numerical Methods in Engineering*, Vol. 56(13), 2003, pp.1949-1972
13. Tell Them Where it Hurts*, the National Science Foundation (NSF) Press Release 08-040*, March 2008, http://www.nsf.gov/news/news_summ.jsp?cntn_id=111276&org=olpa&from=news
14. Pressure Points, *Discover Magazine*, May 2009, page 8.
15. A. Borri, et al., Diagnosis of lesions and static analysis of the Michelangelo's David, *Technical Report*, Dipartimento di Ingegneria Civile d Ambientale-Universita degli Studi Perugia