1

From CAD and FEA to Isogeometric Analysis: An Historical Perspective

1.1 Introduction

1.1.1 The need for isogeometric analysis

It may seem inconceivable to young engineers, but it was not long ago that computers were nowhere to be seen in design offices. Designers worked at drawing boards and designs were drawn with pencils on vellum or Mylar[1]. The design drawings were passed to stress analysts and the interaction between designer and analyst was simple and direct. Times have changed. Designers now generate CAD (Computer Aided Design) files and these must be translated into analysis-suitable geometries, meshed and input to large-scale finite element analysis (FEA) codes. This task is far from trivial and for complex engineering designs is now estimated to take over 80% of the overall analysis time, and engineering designs are becoming increasingly more complex; see Figure 1.1. For example, presently, a typical automobile consists of about 3,000 parts, a fighter jet over 30,000, the Boeing 777 over 100,000, and a modern nuclear submarine over 1,000,000. Engineering design and analysis are not separate endeavors. Design of sophisticated engineering systems is based on a wide range of computational analysis and simulation methods, such as structural mechanics, fluid dynamics, acoustics, electromagnetics, heat transfer, etc. Design speaks to analysis, and analysis speaks to design. However, analysis-suitable models are not automatically created or readily meshed from CAD geometry. Although not always appreciated in the academic analysis community, model generation is much more involved than simply generating a mesh. There are many time consuming, preparatory steps involved. And one mesh is no longer enough. According to Steve Gordon, Principal Engineer, General Dynamics / Electric Boat Corporation, “We find that today’s bottleneck in CAD-CAE integration is not only automated mesh generation, it lies with efficient creation of appropriate ‘simulation-specific’ geometry.” (In the commercial sector analysis is usually referred to as CAE, which stands for Computer Aided Engineering.) The anatomy of the process has been studied by Ted Blacker, Manager of Simulation Sciences, Sandia National Laboratories. At

Isogeometric Analysis: Toward Integration of CAD and FEA by J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs

P1: ABC/ABC P2: c
c01 JWBK372-Cottrell June 12, 2009 15:11 Printer Name: Yet to Come

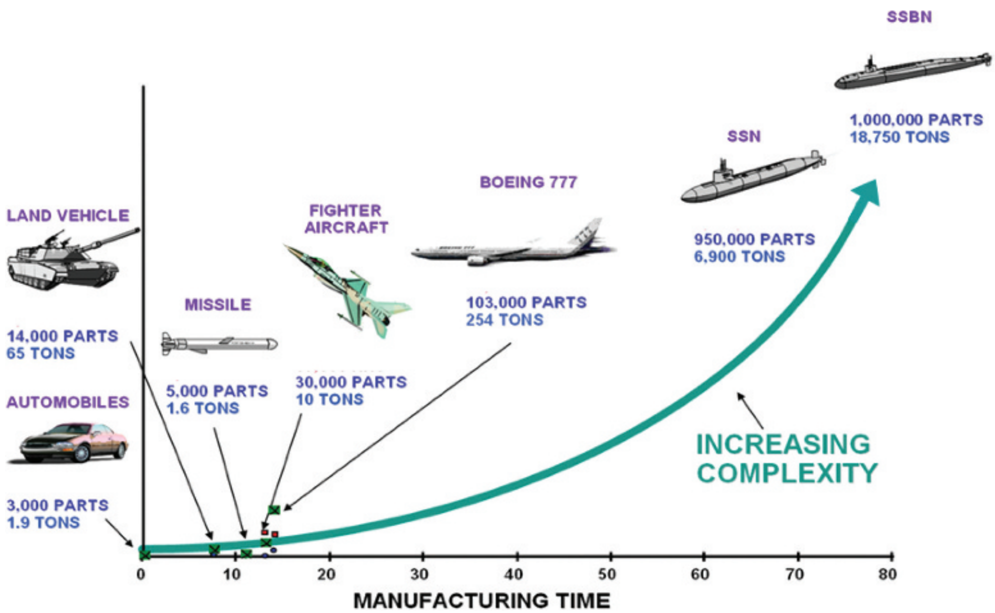


Figure 1.1 Engineering designs are becoming increasingly complex, making analysis a time consuming and expensive endeavor. (Courtesy of General Dynamics / Electric Boat Corporation).

Sandia, mesh generation accounts for about 20% of overall analysis time, whereas creation of the analysis-suitable geometry requires about 60%, and only 20% of overall time is actually devoted to analysis per se; see Figure 1.2. The 80/20 modeling/analysis ratio seems to be a very common industrial experience, and there is a strong desire to reverse it, but so far little progress has been made, despite enormous effort to do so. The integration of CAD and FEA has proven a formidable problem. It seems that fundamental changes must take place to fully integrate engineering design and analysis processes.

Recent trends taking place in engineering analysis and high-performance computing are also demanding greater precision and tighter integration of the overall modeling-analysis process. We note that a finite element mesh is only an approximation of the CAD geometry, which we view as "exact." This approximation can in many situations create errors in analytical results. The following examples may be mentioned: Shell buckling analysis is very sensitive to geometric imperfections (see Figure 1.3), boundary layer phenomena are sensitive to the precise geometry of aerodynamic and hydrodynamic configurations (see Figures 1.4 and 1.5), and sliding contact between bodies cannot be accurately represented without precise geometric descriptions (see Figure 1.6). The Babuška paradox (see Birkhoff and Lynch, 1987) is another example of the pitfalls of polygonal approximations to curved boundaries. Automatic adaptive mesh refinement has not been as widely adopted in industry as one might assume from the extensive academic literature, because mesh refinement requires access to the exact geometry and thus seamless and automatic communication with CAD, which simply does not exist. Without accurate geometry and mesh adaptivity, convergence and high-precision results are impossible.

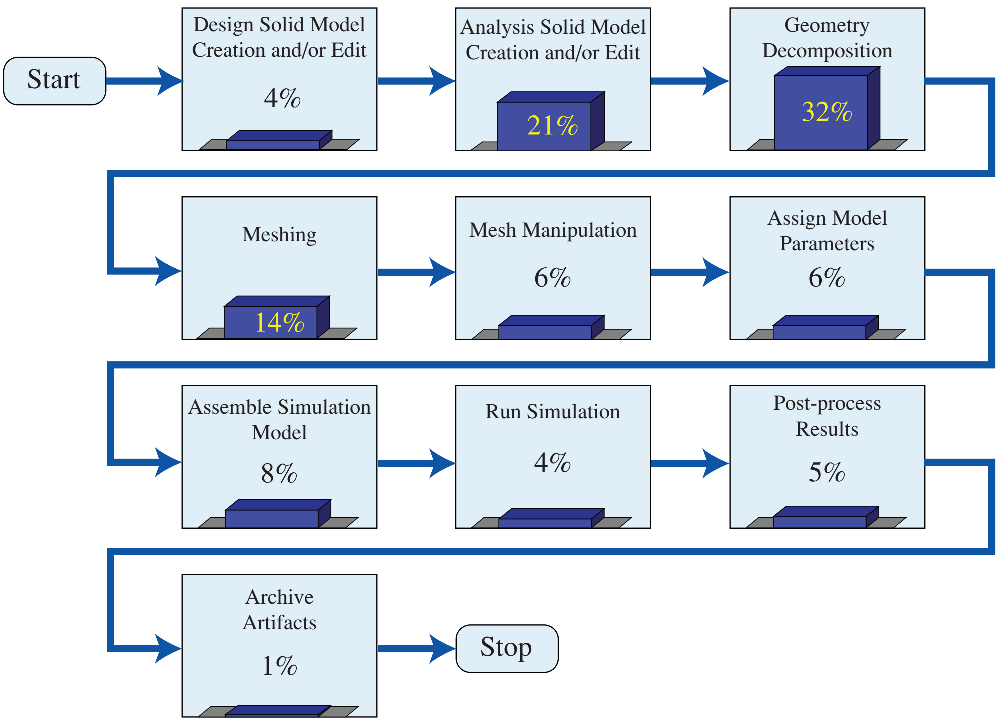


Figure 1.2 Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. Note that the process of building the model completely dominates the time spent performing analysis. (Courtesy of Michael Hardwick and Robert Clay, Sandia National Laboratories.).

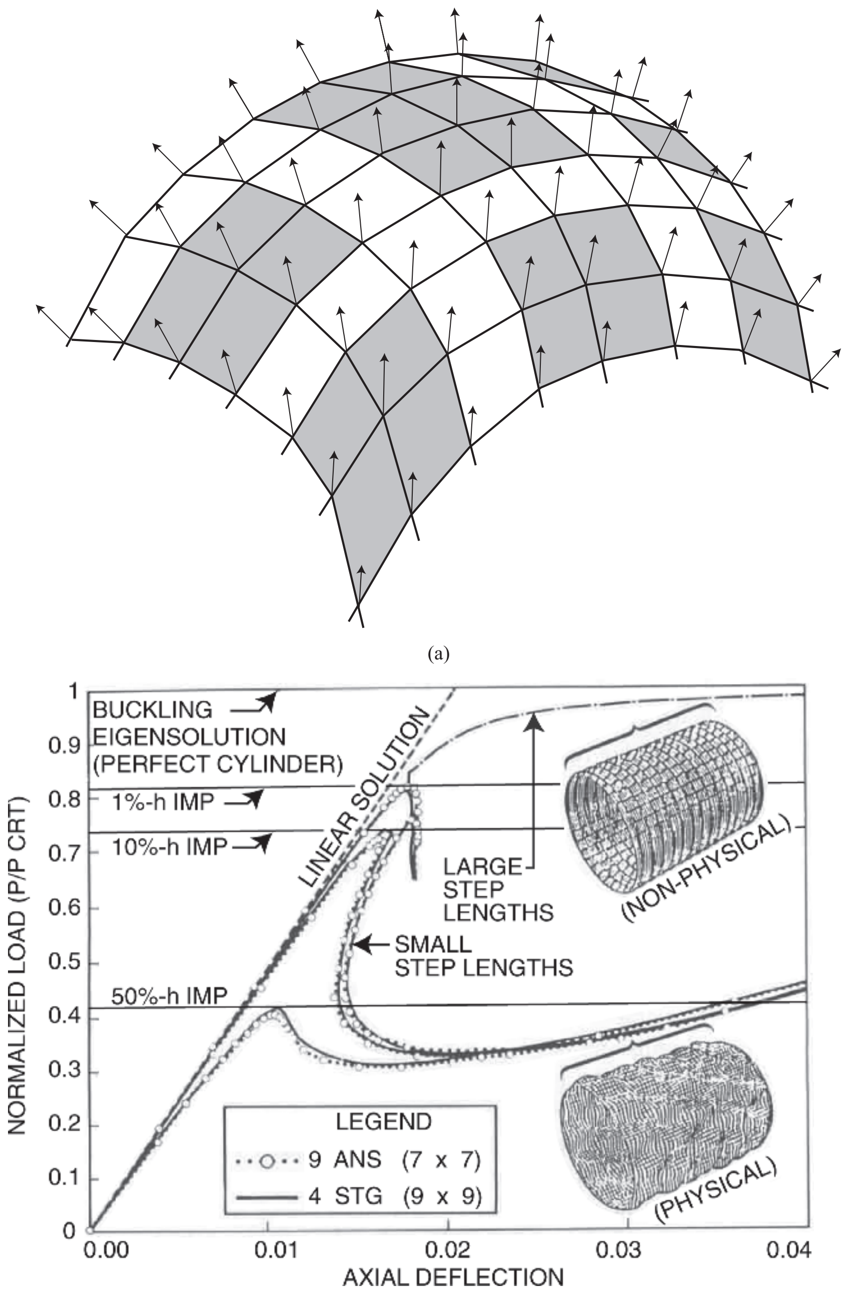
Deficiencies in current engineering analysis procedures also preclude successful application of important pace-setting technologies, such as design optimization, verification and validation (V&V), uncertainty quantification (UQ), and petascale computing.

The benefits of design optimization have been largely unavailable to industry. The bottleneck is that to do shape optimization the CAD geometry-to-mesh mapping needs to be automatic, differentiable, and tightly integrated with the solver and optimizer. This is simply not the case as meshes are disconnected from the CAD geometries from which they were generated.

V&V requires error estimation and adaptivity, which in turn requires tight integration of CAD, geometry, meshing, and analysis. UQ requires simulations with numerous samples of models needed to characterize probability distributions. Sampling puts a premium on the ability to rapidly generate geometry models, meshes, and analyses, which again leads to the need for tightly integrated geometry, meshing, and analysis.

The era of petaflop computing is on the horizon. Parallelism keeps increasing, but the largest unstructured mesh simulations have stalled, because no one truly knows how to generate and adapt massive meshes that keep up with increasing concurrency. To be able to capitalize on the era of $O(100{,}000)$ core parallel systems, CAD, geometry, meshing, analysis, adaptivity, and visualization all have to run in a tightly integrated way, in parallel, and in a scalable fashion.

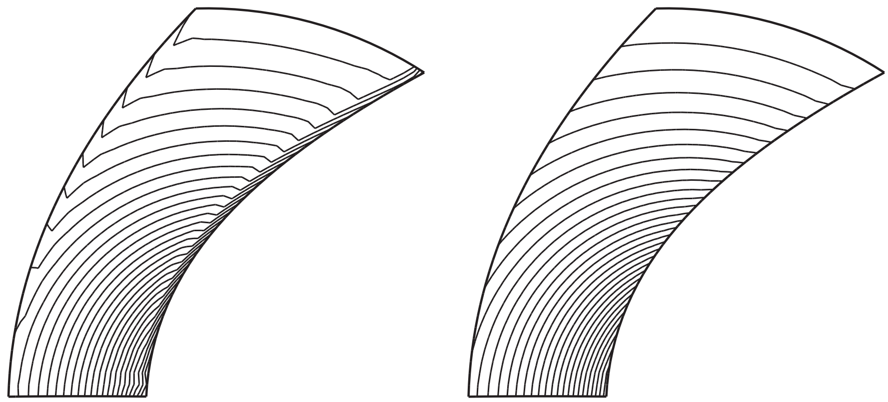
P1: ABC/ABC P2: c
c01 JWBK372-Cottrell June 12, 2009 15:11 Printer Name: Yet to Come



(a)

(b)

Figure 1.3 Thin shell structures exhibit significant imperfection sensitivity. (a) Faceted geometry of typical finite element meshes introduces geometric imperfections (adapted from Gee *et al.*, 2005). (b) Buckling of cylindrical shell with random geometric imperfections. The buckling load depends significantly upon the magnitude of the imperfections (from Stanley, 1985).



(a) (b)

Figure 1.4 Isodensity contours of Galerkin/least-squares (GLS) discretization of Ringleb flow. (a) Isoparametric linear triangular element approximation: both solution and geometry are represented by piecewise linear functions. (b) Super-isoparametric element approximation: solution is piecewise linear, while geometry is piecewise quadratic. Smooth geometry avoids spurious entropy layers associated with piecewise-linear geometric approximations (from Barth, 1998).

It is apparent that the way to break down the barriers between engineering design and analysis is to reconstitute the entire process, but at the same time maintain compatibility with existing practices. A fundamental step is to focus on one, and only one, geometric model, which can be utilized directly as an analysis model, or from which geometrically precise analysis models can be automatically built. This will require a change from classical finite

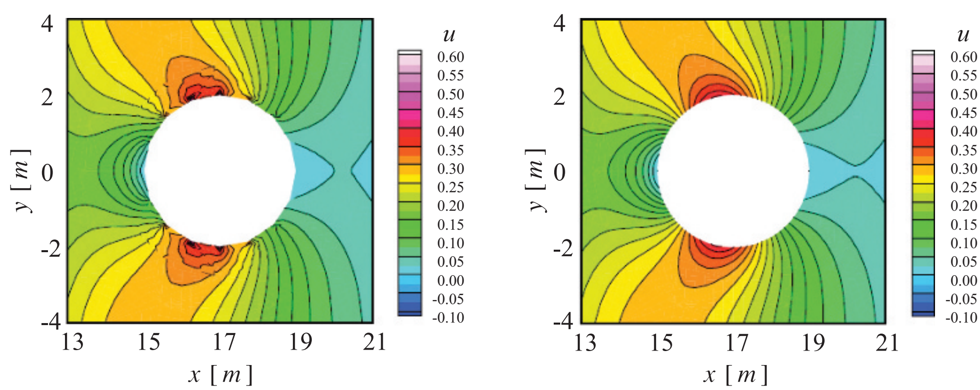
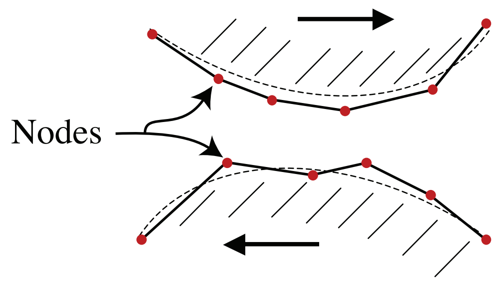
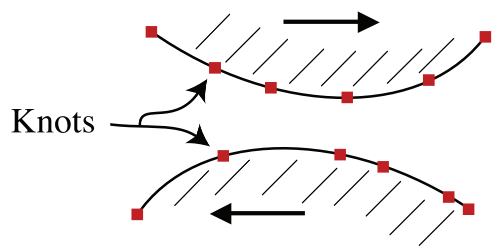


Figure 1.5 The two-dimensional Boussinesq equations. The x-component of velocity obtained using 552 triangles with fifth order polynomials on each triangle. On the left, the elements are straight-sided. The spurious oscillations in the solution on the left are due to the use of straight-sided elements for the geometric approximation. On the right, the cylinder is approximated by elements with curved edges, and the oscillations are eliminated (from Eskilsson and Sherwin, 2006).



(a) Finite element analysis



(a) Isogeometric analysis

Figure 1.6 Sliding contact. (a) Faceted polynomial finite elements create problems in sliding contact (see Laursen, 2002 and Wriggers, 2002). (b) NURBS geometries can attain the smoothness of real bodies.

element analysis to an analysis procedure based on CAD representations. This concept is referred to as ***Isogeometric Analysis***, and it was introduced in Hughes *et al.*, 2005. Since then a number of additional papers have appeared (Bazilevs *et al.*, 2006a, 2006b; Cottrell *et al.*, 2006, 2007; Zhang *et al.*, 2007; Gomez *et al.*, 2008).

Here are the reasons why the time may be right to transform design and analysis technologies: Initiatory investigations of the isogeometric concept have proven very successful. Backward compatibility with existing design and analysis technologies is attainable. There is interest in both the computational geometry and analysis communities to embark on isogeometric research. Several mini-symposia and workshops at international meetings have been held and several very large multi-institutional research projects have begun in Europe. In particular, EXCITING – exact geometry simulation for optimized design of vehicles and vessels – is a three year, six million dollar project focused on developing computational tools for the optimized design of functional free-form surfaces, and the Integrated Computer Aided Design and Analysis (ICADA) project is a five year, five million dollar initiative focused on bridging the gap between design and analysis in industry through isogeometric analysis.

There is an inexorable march toward higher precision and greater reality. New technologies are being introduced and adopted rapidly in design software to gain competitive advantage. New and better analysis technologies can be built upon and influence these new CAD technologies. Engineering analysis can leverage these developments as a basis for the isogeometric concept.

P1: ABC/ABC P2: c
c01 JWBK372-Cottrell June 12, 2009 15:11 Printer Name: Yet to Come

Anyone who has lived the last 60 years is acutely aware of the profound changes that have occurred due to the emergence of new technologies. History has demonstrated repeatedly that statements to the effect that "people will not change" are false. An interesting example of a paradigm shift concerns the slide rule, a mechanical device that dominated computing for approximately 350 years. In the 20th century alone nearly 40 million slide rules were produced throughout the world. The first transistorized electronic calculators emerged in the early 1960s, with portable four-function models available by the end of the decade. The first hand-held scientific calculator, Hewlett-Packard's HP35, became commercially available in 1972. Keuffel and Esser Co., the world's largest producer of slide rules, manufactured its last slide rule in 1975, just 3 years later (see Stoll, 2006).

1.1.2 Computational geometry

There are a number of candidate computational geometry technologies that may be used in isogeometric analysis. The most widely used in engineering design are NURBS (non-uniform rational B-splines), the industry standard (see, Piegl and Tiller, 1997; Farin, 1999a, 1999b; Cohen *et al.*, 2001; Rogers, 2001). The major strengths of NURBS are that they are convenient for free-form surface modeling, can exactly represent all conic sections, and therefore circles, cylinders, spheres, ellipsoids, etc., and that there exist many efficient and numerically stable algorithms to generate NURBS objects. They also possess useful mathematical properties, such as the ability to be refined through knot insertion, C^{p-1}-continuity for pth-order NURBS, and the variation diminishing and convex hull properties. NURBS are ubiquitous in CAD systems, representing billions of dollars in development investment. One may argue the merits of NURBS versus other computational geometry technologies, but their preeminence in engineering design is indisputable. As such, they were the natural starting point for isogeometric analysis and their use in an analysis setting is the focus of this book.

T-splines (Sederberg *et al.*, 2003; Sederberg *et al.*, 2004) are a recently developed forward and backward generalization of NURBS technology. T-splines extend NURBS to permit local refinement and coarsening, and are very robust in their ability to efficiently sew together adjacent patches. Commercial T-spline plug-ins have been introduced in Maya and Rhino, two NURBS-based design systems (see references T-Splines, Inc., 2008a and T-Splines, Inc., 2008b). Initiatory investigations of T-splines in an isogeometric analysis context have been undertaken by Bazilevs *et al.*, 2009 and Dorfel *et al.*, 2008. These works point to a promising future for T-splines as an isogeometric technology.

There are other computational geometry technologies that also warrant investigation as a basis of isogeometric analysis. One is subdivision surfaces which use a limiting process to define a smooth surface from a mesh of triangles or quadrilaterals (see, *e.g.*, Warren and Weimer, 2002; Peters and Reif, 2008). They have already been used in analysis of shell structures by Cirak *et al.*, 2000; Cirak and Ortiz, 2001, 2002. The appeal of subdivision surfaces is there is no restriction on the topology of the control grid. Like T-splines, they also create gap-free models. Most of the characters in Pixar animations are modeled using subdivision surfaces. The CAD industry has not adopted subdivision surfaces very widely because they are not compatible with NURBS. With billions of dollars of infrastructure invested in NURBS, the financial cost would be prohibitive. Nevertheless, subdivision surfaces should play an important role in isogeometric technology. Subdivision solids have been studied by Bajaj *et al.*, 2002.

Other geometric technologies that may play a role in the future of isogeometric analysis include Gordon patches (Gordon, 1969), Gregory patches (Gregory, 1983), S-patches (Loop and DeRose, 1989), and A-patches (Bajaj *et al.*, 1995). Provatidis has recently solved a number of problems using Coons patches (see Provatidis, 2009, and references therein). Others may be invented specifically with the intent of fostering the isogeometric concept, namely, to use the surface design model directly in analysis. This would only suffice if analysis only requires the surface geometry, such as in the stress or buckling analysis of a shell. In many cases, the surface will enclose a volume and an analysis model will need to be created for the volume. The basic problem is to develop a three-dimensional (trivariate) representation of the solid in such a way that the surface representation is preserved. This is far from a trivial problem. Surface differential and computational geometry and topology are now fairly well understood, but the three-dimensional problem is still open (the Thurston conjecture characterizing its solution remains to be proven, see Thurston, 1982, 1997). The hope is that through the use of new technologies, such as, for example, Ricci flows and polycube splines (see Gu and Yau, 2008), progress will be forthcoming.

1.2 The evolution of FEA basis functions

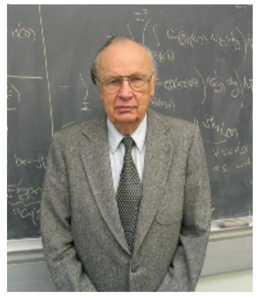
Solution of partial differential equations by the finite element method consists, roughly speaking, of a variational formulation and trial and weighting function spaces defined by their respective basis functions. These basis functions are defined in turn by finite elements, local representations of the spaces. The elements are a non-overlapping decomposition of the problem domain into simple shapes (*e.g.*, triangles, quadrilaterals, tetrahedra, hexahedra, etc.). In the most widely used variational methods, the trial and weighting functions are essentially the same. Specifically, the same elements are used in their construction. There are three ways to improve a finite element method:

1 Improve the variational method. Sometimes this can be done in such a way as to correct a shortcoming in the finite elements for the problem under consideration, such as, for example, through the use of selective integration (see Hughes, 2000). Another way is to use an alternative variational formulation with improved properties, an example being "stabilized methods." See Brooks and Hughes, 1982; Hughes *et al.*, 2004.
2 Improve the finite element spaces, that is, the elements themselves.
3 Improve both, that is, the variational method *and* the elements.

Our focus here is on finite element spaces and ultimately how they perform in comparison to spaces of functions built from NURBS, T-splines, etc. Consequently, we will give a brief review of the historical milestones in finite elements.

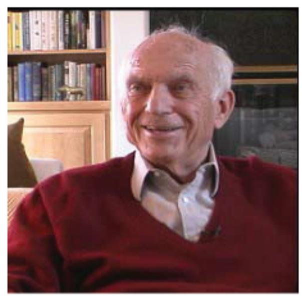
Typically, finite elements are defined in terms of interpolatory polynomials. The classical families of polynomials, especially the Lagrange and Hermite polynomials, are widely utilized (see Hughes, 2000). These may be considered the historical antecedents of finite elements.


John Argyris


Ivo Babuška

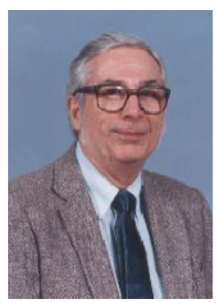

Ted Belytschko


Franco Brezzi

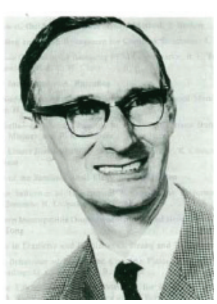

Ray Clough


Richard Courant


Leszek Demkowicz


Jim Douglas, Jr.


Richard Gallagher


Bruce Irons


Wing Kam Liu


Jean-Claude Nédélec

Figure 1.7 Finite element picture gallery.

J. Tinsley Oden D. R. J. Owen Pierre Arnaud Raviart

Robert L. Taylor Olgierd C. Zienkiewicz

Figure 1.7 (*continued*)

Early publications in the engineering literature describing what is now known as the finite element method were Argyris and Kelsey, 1960, which is a collection of articles by those authors dating from 1954 and 1955, and Turner *et al.*, 1956. The term “finite elements” was coined by Clough, 1960. However, the first finite element, the linear triangle, can be traced all the way back to Courant, 1943. It is perhaps the simplest element and is still widely used today. It is interesting to note that the engineering finite element literature was unaware of this reference until sometime in the late 1960s by which time the essential features of the finite element method were well established. The linear tetrahedron appeared in Gallagher *et al.*, 1962. Through the use of triangular and tetrahedral coordinates (*i.e.*, barycentric coordinates) and the Pascal triangle and tetrahedron, it became a simple matter to generate C^0-continuous finite elements for straight-edged triangles and flat-surfaced tetrahedra. The bilinear quadrilateral was developed by Taig, 1961, and it presaged the development of ***isoparametric elements*** (Irons, 1966; Zienkiewicz and Cheung, 1968), perhaps the most important concept in the history of element technology.

The idea of isoparametric elements immediately generalized elements which could be developed on a regular parent domain, such as a square, or a cube, to an element which could take on a smoothly curved shape in physical space. Furthermore, it was applicable to any element topology, including triangles, tetrahedra, etc. An essential feature was that the spaces so constructed satisfied basic mathematical convergence criteria, as well as physical attributes in problems of mechanics, namely, the ability to represent all affine motions (*i.e.*, rigid translations and rotations, uniform stretchings and shearings) exactly. Curved quadrilateral and hexahedral elements became popular in structural and solid mechanics applications. The classical isoparametric elements were developed using tensor-product constructs. Subsequently, procedures were developed to circumvent the necessity of the tensor-product format. The eight-node serendipity quadrilateral was an early noteworthy example (Zienkiewicz *et al.*, 1971). This eventually led to the variable-number-of-nodes concept (Zienkiewicz *et al.*, 1970; Taylor, 1972; see Hughes, 2000, chapter 3, for examples).

In practical applications computational efficiency is critical. In nonlinear dynamic applications low-order elements have played a dominant role. The constant pressure bilinear quadrilateral element (Hughes and Allik, 1969; Nagtegaal *et al.*, 1974; Hughes, 1977, 1980; Malkus and Hughes, 1978) and its three-dimensional generalization, the constant pressure trilinear hexahedral element, have dominated nonlinear solid mechanical calculations. Effective one-point integration quadrilateral bending elements (Hughes et al., 1977, 1978; Hughes and Liu, 1981a, 1981b; Flanagan and Belytschko, 1981; Belytschko and Tsay, 1983; Belytschko *et al.*, 1984) with scaled lumped rotatory inertia mass matrices (Hughes *et al.*, 1978; Hughes, 2000) enabled automobile crash analysis to become a standard design tool. The cost of analysis prior to these developments precluded its practical use.

A limitation of the isoparametric concept was that while it worked for C^0-continuous interpolation, it did not for C^1 or higher. There was a strong interest in the development of C^1-continuous interpolation schemes primarily because of the desire to construct thin plate and shell elements for structural analysis. Thin bending elements require square-integrability of generalized second derivatives and so C^1-continuous elements constitute a suitable subspace. Many researchers sought solutions to this problem. Noteworthy successes were due to Clough and Tocher, 1965; Argyris *et al.*, 1968; Cowper *et al.*, 1968; de Veubeke, 1968; Bell, 1969. However, these elements were complicated to use and expensive, and interest moved to different variational formulations to circumvent the need for C^1-continuous basis functions. This is an example where it was more convenient to adopt a different variational formulation than construct appropriate discrete approximation subspaces for the original one. It should be said, however, that the development of effective Reissner-Mindlin bending elements, requiring only C^0-continuity, was not without its own difficulties.

Mathematicians have played a prominent role in devising discrete approximation spaces for certain classes of variational formulations. Noteworthy examples are due to Raviart and Thomas, 1977, and Brezzi *et al.*, 1985; see also Brezzi and Fortin, 1991, for Darcy flow (these are referred to as $H(\text{div})$ elements) and Nedelec, 1980, Demkowicz, 2007, and Demkowicz *et al.*, 2008 for Maxwell's equations (these are referred to as $H(\text{curl})$ and $H(\text{div}) \oplus H(\text{curl})$ elements). The engineering and mathematics literatures are also replete with various alternative variational formulations that enhance the performance of simple elements.

Another recent trend in basis function construction has been away from the classical concept of an element decomposition. These approaches have come to be known as meshless methods (Nayroles *et al.*, 1992) and they have generated considerable interest. Noteworthy contributions to meshless methods are the element-free Galerkin method of Belytschko *et al.*, 1994, the reproducing kernel particle method of Liu *et al.*, 1995, the partition of unity method of Melenk and Babuska, 1996, and the *hp*-clouds of Duarte and Oden, 1996. This is another subject entirely, but we note that, as in the case of the finite element method, the link to CAD geometry, at best, is tenuous (see, *e.g.*, Sakurai, 2006). A timeline of FEA and meshless basis function development is presented in Table 1.1

Table 1.1 Timeline: Milestones in FEA and meshless basis function development

1779	Lagrange polynomials
1864	Hermite polynomials
1943	Linear triangle
1960	Clough coins the name "finite elements"
1961	Bilinear quadrilateral
1962	Linear tetrahedron
1965–1968	C^1-continuous triangles and quadrilaterals
1966	Isoparametric elements
1968–1971	Variable-number-of-nodes elements
1977–1986	H(div), H(curl), and $H(\text{div}) \oplus H(\text{curl})$ elements
1992–1996	Meshless methods

Another class of meshless methods that has enjoyed recent popularity is that of particle methods. An early variant is so-called smoothed particle hydrodynamics (Gingold and Monaghan, 1977). The particle finite element method of Oñate *et al.*, 1996 utilizes geometric reconstruction from particles combined with finite element remeshing strategies and thus has features in common with meshless methods and classical finite element discretizations. The discrete element method of Cundall and Strack, 1979 (see also Munjiza *et al.*, 1995) likewise combines ideas of particles and finite elements. These procedures have opened the way to solution of very complex engineering problems that are beyond the scope of classical finite element procedures.

It needs to be mentioned again that finite elements never faithfully replicate the CAD geometry. It is always a piecewise polynomial approximation. In most cases involving complex engineering designs, it has now become a much more formidable task to generate a finite element model from the CAD geometry than to perform the analysis. This is the primary motivation behind the development of the isogeometric concept.

1.3 The evolution of CAD representations

It is generally agreed that present day CAD had its origins in the work of two French automotive engineers, Pierre Bézier of Renault and Paul de Faget de Casteljau of Citroën. Bézier, 1966, 1967, and 1972 utilized the Bernstein polynomial basis (Bernstein, 1912) to generate curves and surfaces. De Casteljau, 1959, developed similar ideas, but his work was never published in the open literature. Although there seem to be earlier instances of work utilizing splines, the term "spline" was introduced in the mathematical literature by Schoenberg, 1946, whose work drew attention to the possibilities of spline approximations, but the subject did not become active until the 1960s (see Curry and Schoenberg, 1966). During the early years, the role of the Coons patch (Coons, 1967), based on the idea of generalized Hermite interpolation (http://en.wikipedia.org/wiki/Hermite_interpolation), predominated but its influence faded subsequently in favor of the methods of Bézier and de Casteljau. A number of fundamental contributions occurred during the 1970s beginning with Reisenfeld's Ph.D. dissertation on B-splines (Riesenfeld, 1972). This was followed shortly thereafter by Versprille's Ph.D. dissertation on rational B-splines, which have become known as NURBS (Versprille, 1975).

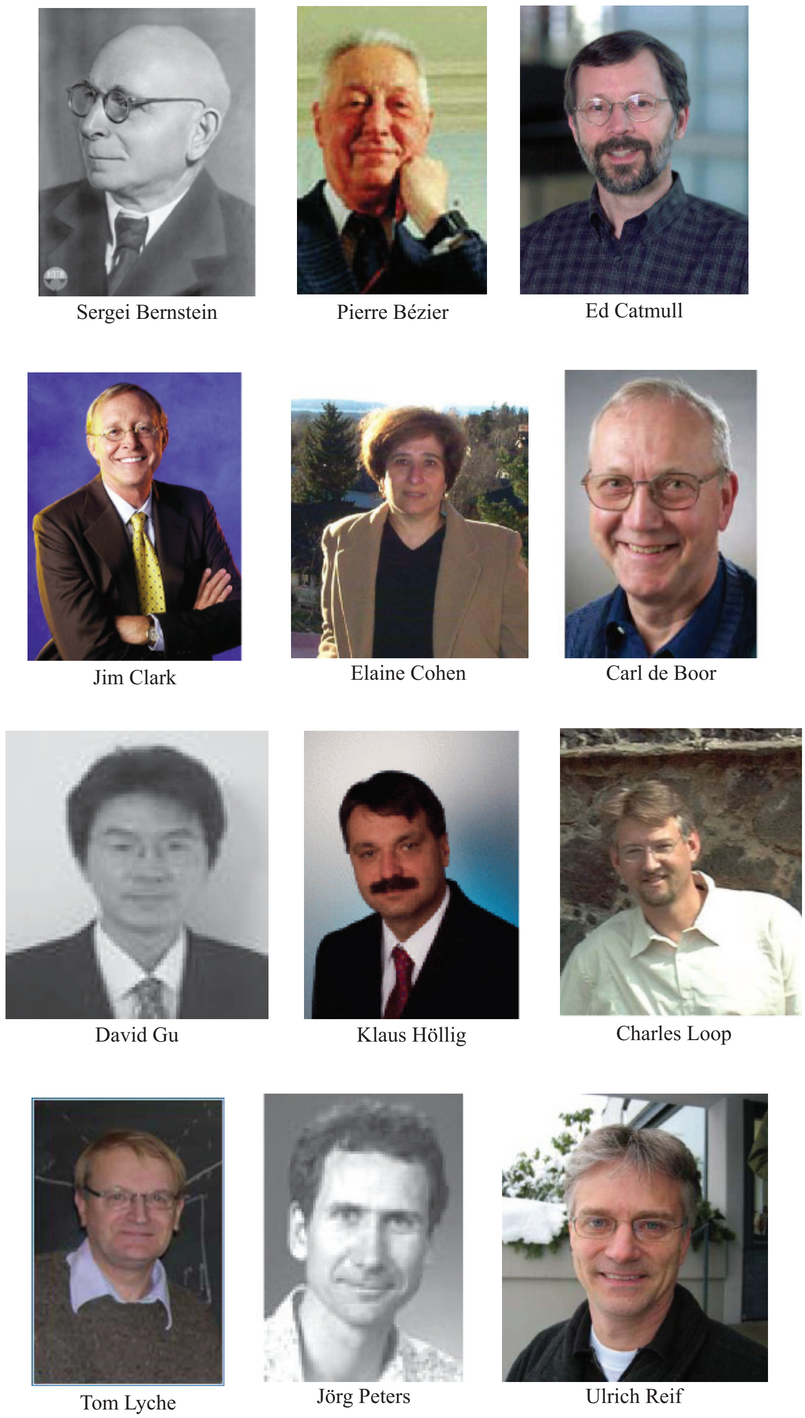


Figure 1.8 Computational Geometry Picture Gallery.

Rich Riesenfeld Malcolm Sabin Isaac Schoenberg

Peter Schröder Larry Schumaker Tom Sederberg

Figure 1.8 (*continued*)

There are many efficient and numerically stable algorithms that have been developed to manipulate B-splines, for example, the Cox–de Boor recursion (Cox, 1971; de Boor, 1972), the de Boor algorithm (de Boor, 1978), the Oslo algorithm (Cohen *et al.*, 1980), polar forms and blossoms (Ramshaw, 1987a; Ramshaw, 1989), etc.

Another major development in the 1970s was the pioneering work on subdivision surfaces (Catmull and Clark, 1978; Doo and Sabin, 1978). Ed Catmull is the CEO of Pixar and Walt Disney Animation Studios and Jim Clark was the founder of Silicon Graphics and Netscape. The seminal ideas of subdivision are generally attributed to de Rham, 1956 and Chaikin, 1974. Other works of note are Lane and Riesenfeld, 1980, which is intimately linked to Bézier and B-spline surfaces, and Loop, 1987, which is box spline based. Subdivision surfaces have become popular in the field of animation. They generate smooth surfaces from quadrilateral or triangular (Loop, 1987) surface meshes. For engineering design, NURBS are still the dominant technology. Recent generalizations of NURBS-based technology that allow some unstructuredness are T-splines (Sederberg *et al.*, 2003, 2004). T-splines constitute a superset of NURBS (*i.e.*, every NURBS is a T-spline) and the local refinement properties of T-splines facilitate solution of the gap/overlap problem of intersecting NURBS surfaces. A recent work

Table 1.2 Timeline: Milestones in CAD representations

1912	Bernstein polynomials
1946	Schoenberg coins the name "spline"
1959	de Casteljau algorithm
1966–1972	Bézier curves and surfaces
1971, 1972	Cox-de Boor recursion
1972	B-splines
1975	NURBS
1978	Catmull–Clark and Doo–Sabin subdivision surfaces
1980	Oslo knot insertion algorithm
1987	Loop subdivision
1987, 1989	Polar forms, blossoms
1996–present	Triangular and tetrahedral B-splines
2003	T-splines

shows how to replace trimmed NURBS surfaces with untrimmed T-splines (Sederberg *et al.*, 2008). Table 1.2 presents a timeline of important developments in CAD.

Other technologies of note include triangular and tetrahedral generalizations of B-splines (see Lai and Schumaker, 2007).

Splines have also been used as a basis for solving variational problems (see, *e.g.*, Schultz, 1973; Prenter 1975; Höllig 2003; Kwok *et al.*, 2001), but these efforts have been dwarfed by activity in finite element analysis. Spline finite elements were also developed in the (second) Ph.D. thesis of Malcolm Sabin (Sabin, 1997).

It is interesting to note that isoparametric elements developed in the 1960s are still the most widely utilized elements in commercial FEA codes, and even in research activities in FEA. This is in contrast to CAD in which fundamentally new technologies, such as T-splines, have only recently been introduced. It seems very likely that this trend may continue, presenting new opportunities to unify CAD and FEA.

Earlier attempts to integrate finite element analysis and computational geometry were referred to as "physically-based modeling." Several researchers developed tools for free-form geometric design based on mechanical principles (see, *e.g.*, Celniker and Gossard, 1991; Terzopoulos and Qin, 1994; Kagan *et al.*, 1998; Volpin *et al.*, 1999; Bronstein *et al.*, 2008). For example, rather than manipulating a B-spline surface by explicitly moving the control points, the material properties of a thin metal shell are ascribed to the surface so that the geometry may be deformed by applying fictitious forces wherever desired by the designer to "mold" the surface into the desired configuration. This mechanical approach to geometrical modeling is appealing in that the geometries respond in very intuitive ways. The difficulty is that it requires solving differential equations, frequently using an FEA-based approach, each time the designer modifies its shape. Many approaches to such modeling are inherently isogeometric. Those who develop physically-based design systems and those who develop isogeometric analysis capabilities have many goals in common. The futures of these technologies are probably linked.

1.4 Things you need to get used to in order to understand NURBS-based isogeometric analysis

In FEA there is one notion of a mesh and one notion of an element, but an element has two representations, one in the parent domain and one in the physical space. Elements are usually defined by their nodal coordinates and the degrees-of-freedom are usually the values of the basis functions at the nodes. Finite element basis functions are typically interpolatory and may take on positive and negative values. Finite element basis functions are often referred to as "interpolation functions," of "shape functions." See Hughes, 2000 for a discussion of the basic concepts.

In NURBS, the basis functions are usually not interpolatory. There are two notions of meshes, the control mesh[2] and the physical mesh. The control points define the control mesh, and the control mesh interpolates the control points. The control mesh consists of multilinear elements, in two dimensions they are bilinear quadrilateral elements, and in three dimensions they are trilinear hexahedra. The control mesh does not conform to the actual geometry. Rather, it is like a scaffold that controls the geometry. The control mesh has the look of a typical finite element mesh of multilinear elements. The control variables are the degrees-of-freedom and they are located at the control points. They may be thought of as "generalized coordinates." Control elements may be degenerated to more primitive shapes, such as triangles and tetrahedra. The control mesh may also be severely distorted and even inverted to an extent, while at the same time, for sufficiently smooth NURBS, the physical geometry may still remain valid (in contrast with finite elements).

The physical mesh is a decomposition of the actual geometry. There are two notions of elements in the physical mesh, the patch and the knot span. The patch may be thought of as a macro-element or subdomain. Most geometries utilized for academic test cases can be modeled with a single patch. Each patch has two representations, one in a parent domain and one in physical space. In two-dimensional topologies, a patch is a rectangle in the parent domain representation. In three dimensions it is a cuboid.

Each patch can be decomposed into knot spans. Knots are points, lines, and surfaces in one-, two-, and three-dimensional topologies, respectively. Knot spans are bounded by knots. These define element domains where basis functions are smooth (*i.e.*, C^{∞}). Across knots, basis functions will be C^{p-m} where p is the degree[3] of the polynomial and m is the multiplicity of the knot in question. Knot spans are convenient for numerical quadrature. They may be thought of as micro-elements because they are the smallest entities we deal with. They also have representations in both a parent domain and physical space. When we speak of "elements" without further description, we usually mean knot spans.

There is one other very important notion that is a key to understanding NURBS, the ***index space*** of a patch. It uniquely identifies each knot and discriminates among knots having multiplicity greater than one.

See Table 1.3 for a summary of NURBS paraphernalia employed in isogeometric analysis. A schematic illustration of the ideas is presented in Figure 1.9 for a NURBS surface in $\mathbb{R}^3$. Detailed examples will be provided in subsequent chapters.

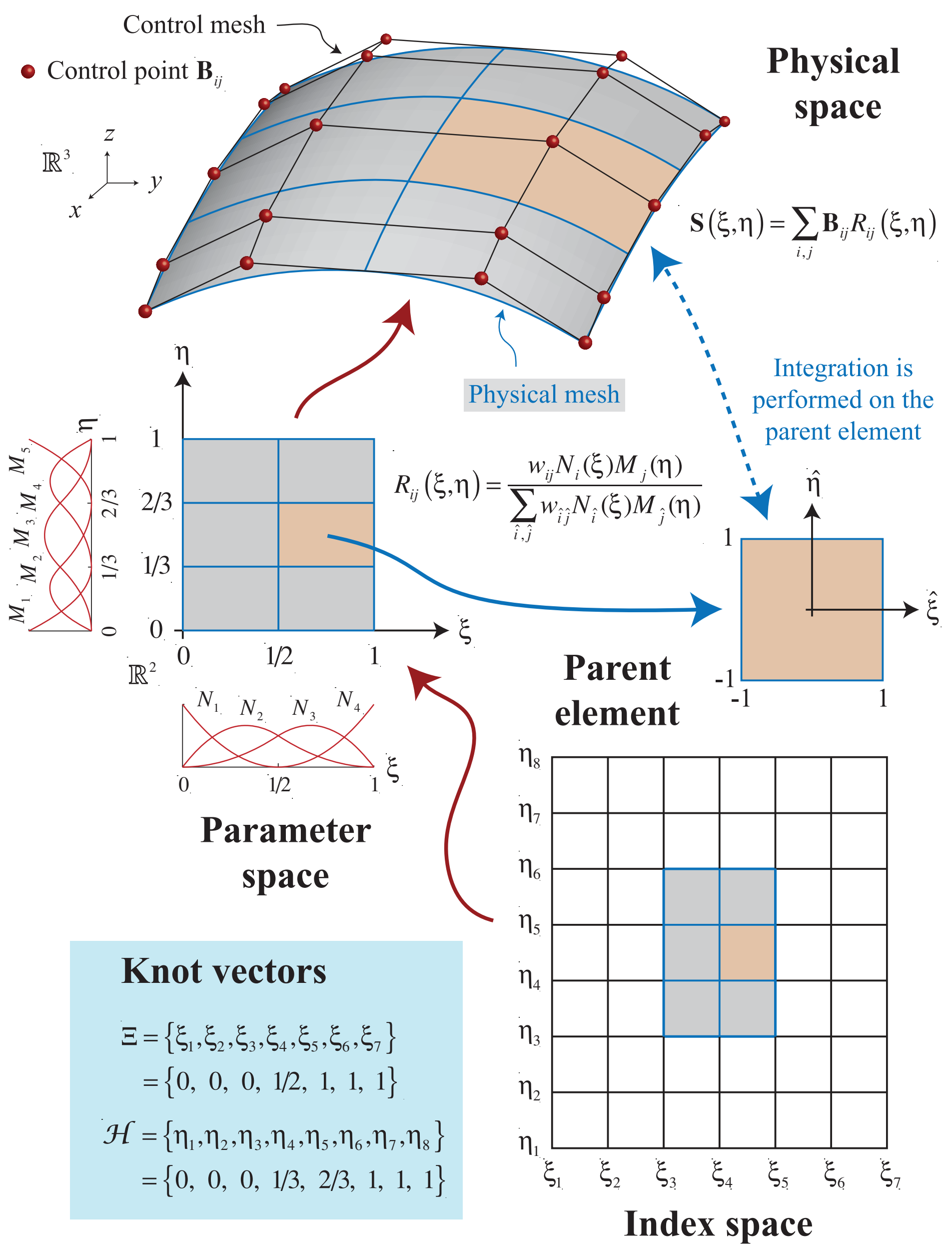


Figure 1.9 Schematic illustration of NURBS paraphernalia for a one-patch surface model. Open knot vectors and quadratic C^1-continuous basis functions are used. Complex multi-patch geometries may be constructed by assembling control meshes as in standard finite element analysis. Also depicted are C^1-quadratic ($p = 2$) basis functions determined by the knot vectors. Basis functions are multiplied by control points and summed to construct geometrical objects, in this case a surface in $\mathbb{R}^3$. The procedure used to define basis functions from knot vectors will be described in detail in Chapter 2.

Table 1.3 NURBS paraphernalia in isogeometric analysis

Index Space		
Control Mesh	Physical Mesh	
Multilinear Control Elements	Patches	Knot Spans
Topology: **1D:** Straight lines defined by two consecutive control points	Patches: Images of rectangular meshes in the parent domain mapped into the actual geometry. Patches may be thought of as macro-elements or subdomains.	Topology of knots in the parent domain: **1D:** Points **2D:** Lines **3D:** Planes
2D: Bilinear quadrilaterals defined by four control points	Topology: **1D:** Curves **2D:** Surfaces **3D:** Volumes	Topology of knots in the physical space: **1D:** Points **2D:** Curves **3D:** Surfaces
3D: Trilinear hexahedra defined by eight control points	Patches are decomposed into knot spans, the smallest notion of an element.	Topology of knots spans, *i.e.*, "elements": **1D:** Curved segments connecting consecutive knots **2D:** Curved quadrilaterals bounded by four curves **3D:** Curved hexahedra bounded by six curved surfaces

Notes

1. Young engineers may not know what vellum and Mylar are. Vellum is a translucent drafting material made from cotton fiber. Mylar is the trade name of a translucent polyester film used for drafting.
2. The control mesh is also known as the "control net," the "control lattice," and curiously the "control polygon" in the univariate case.
3. There is a terminology conflict between the geometry and analysis communities. Geometers will say a cubic polynomial has degree 3 and order 4. In geometry, order equals degree plus one. Analysts will say a cubic polynomial is order three, and use the terms order and degree synonymously. This is the convention we adhere to.