On the use of the Finite Element Method to Represent Real-World Phenomena in Spatiotemporal Databases

José Duarte
DETI/IEETA, University of Aveiro
Aveiro, Portugal
hfduarte@ua.pt

***Abstract*—The methods proposed in the spatiotemporal databases community to represent the continuous evolution of real-world phenomena from observations do not consider the physical characteristics of the phenomena and the external conditions with which they interact. As a result, the representation has no real physical meaning, and it is hard to establish error estimates and bounds. The finite element method approximates the behavior of a phenomenon using equations based on laws and principles of physics. It considers material properties and external conditions, can handle complex geometries, and provides error estimates and bounds. It requires some expertise to be used correctly, can be expensive, does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, and a structural model has to be defined for every problem. It can be used to predict unknown states, but its use is somewhat limited in the context being proposed.**

Keywords*—*finite element method, morphing, region interpolation problem, spatiotemporal databases

I. BACKGROUNG

Several technologies exist that can be used to collect data on the evolution of real-world phenomena (e.g., sequences of satellite images tracking the evolution of icebergs in the Antarctic, and video recording the evolution of biological tissue). Our goal is to represent the evolution of real-world phenomena in-between know observations using moving regions [1] (i.e., objects whose position, shape and extent change continuously over time) in spatiotemporal database management systems (STDBMSs).

In this context, creating moving regions from snapshots (observations) is called the region interpolation problem [2], and moving regions are represented using the *sliced representation* [3]. In the *sliced representation*, a moving region is an ordered collection of units. A unit represents the evolution of a geometry between a source and a target known geometries, during an interval of time. The evolution of a geometry within a unit is given by an interpolation function, F^+, that should have some properties of interest. In particular: it should have low complexity and allow the processing of large datasets, handle geometries with an arbitrary shape and complexity, generate only valid intermediate geometries, and provide a good approximation of the evolution of the phenomena, ideally with a known error (providing error estimates and bounds), that can be used in applied scientific work (e.g., to perform numerical analysis on the evolution of real-world phenomena).

In [4]–[6] the authors discuss the use of relational database management systems (RDBMSs) as a technology to support scientific computing and computer-based engineering, in particular, to simplify large scale finite element analysis (FEA). This approach differs from the objective (context) presented in this paper (e.g., in this paper, FEA is considered as a method that can potentially be used to create moving regions that will be used to implement operations to study the evolution of phenomena, and the relationships that they establish with other objects and phenomena, over time). The objective is not to use spatiotemporal databases to support FEA.

Morphing techniques are used successfully, for example, in animation packages and computer graphics. Their main goal is to obtain a natural continuous transformation of a geometry between two consecutive known geometries and can potentially be used to implement F^+. Several methods have been proposed in the literature (e.g., using: some type of decomposition [7], deformation transfer [8], and physical principles [9]). However, in general, the physical properties of the phenomena being represented and the external conditions with which they interact, that can have an impact on their evolution, are not considered, and no guarantees on the global and local distortions introduced by the methods are given. As a consequence, it is hard to evaluate the quality of the interpolation objectively and establish an approximation error w.r.t. the actual evolution of the phenomena.

Engineering analysis and computational science simulation are used successfully in many fields (e.g., in structural analysis and fluid flow prediction). They can simulate the physical properties of materials and their interaction with external conditions, predict how a phenomenon will evolve in the future, and provide error estimates and bounds. Therefore, this paper presents a critical overview of the use of the finite element method (FEM) in the context of the region interpolation problem. This should also serve as a reference when considering the use of other numerical methods in the same context.

This paper is organized as follows. Section II presents and characterizes the finite element method. Section III presents an overview on the use of the finite element method in the context of the region interpolation problem. Section IV presents a discussion on the main advantages, disadvantages, and challenges of using the finite element method in the new context, taking as a reference the properties defined in Section I for an ideal interpolation function F^+. Section V presents the conclusions and future work.

II. THE FINITE ELEMENT METHOD

A number of important problems found in nature can be described using partial differential equations (PDEs). Some have no known analytical solution, or if an analytical solution is known, it is not practical to use it. In these situations, numerical methods (e.g., the finite element method (FEM), the boundary element method (BEM), the finite difference method (FDM), the finite volume method

(FVM), and the meshless method) can be used to solve these problems. Because each method has different variations, choosing the best method to be used is application dependent. For example, FEM is a very general method, with a solid mathematical foundation, widely used in continuum mechanics and structural analysis. FVM is preferred in computational fluid dynamics (CFD). It is a conservative method (e.g., it ensures the conservation of mass, momentum and energy at each element of the discretization) widely used to solve models based on conservation laws. It is also becoming common to encounter situations in which different methods are combined to solve a particular problem.

Several types of analysis can be performed (e.g., static, dynamic, linear, and nonlinear). Most real-world phenomena are dynamic and nonlinear in nature. For example, time-dependent (transient) analysis can be used to determine the dynamic response of a structure at different time steps. Nonlinearities include geometric, material, and contact nonlinearities [10] (e.g., large deformations, plasticity, material damage or fracture, and hyper-elasticity). Analysis (e.g., FEA) can be used to understand and predict the behavior, and optimize and control the design and operation, of structures subjected to static or dynamic loads.

Numerical methods have potential applications in several areas [10], and various packages are available to perform numerical analysis on physical phenomena.

FEM [10], [11] is used in engineering as well as in pure and applied sciences (e.g., in continuum mechanics and structural engineering) and it has potential applications in several areas (e.g., geomechanics, biomechanics, and environmental engineering). It is a powerful procedure for the analysis of structures with arbitrary geometry and general material properties, subjected to different types of loads. It can approximate the behavior of a physical (real) system in space and time (e.g., compute the displacements, stresses and strains in a structure under a load). Its main goal is to simulate (predict) with a high degree of accuracy the evolution (behavior) of a phenomenon or structure under certain conditions, using equations that follow established laws and principles of physics, giving error estimates and bounds on the quality of the solution [10]. Various FEM methods and variations have been proposed in the literature (e.g., the extended (XFEM) and the smoothed (S-FEM) finite element methods).

In FEM, a system is divided into a finite number of individual well-defined elements or components whose behavior, specified by a finite number of parameters, can be understood. These simple elements may have physical properties. Then, the solution of the system is given by the local solutions, computed for each element. The quality, validity, and accuracy of the solution depends on the quality of the discretization. In general, geometries with finer elements improve the quality of the simulation (e.g., local displacements and stresses can be captured in greater detail). The precision of the solution, and the efficiency of the method can be improved by choosing an appropriate element type for the discretization. It can be advantageous to use more than one element type to discretize a problem.

In general, FEM solves an equation of the form $\boldsymbol{r} = \boldsymbol{K}\boldsymbol{u}$, where $\boldsymbol{r}$ is a vector of known values (e.g., loads) $\boldsymbol{K}$ is a matrix of known values representing, for example, stiffness, and $\boldsymbol{u}$ are the unknowns at the nodes of the discretization (e.g., the nodal displacements). Boundary conditions and constraints can also be specified (e.g., known displacements).

A. *Classification of a Problem*

In order to select an appropriate structural model and computational method for solving a specific problem [10] the following should be considered:

- Identify the relevant physical phenomena influencing the structure being studied, the nature of the problem, the material properties and the differential equations governing the phenomenon.
- Define the level of accuracy desired, and the variables being studied.

The choices made during this phase are extremely important and can have an impact on the accuracy and validity of the results obtained by the simulation [10]. In general, the following steps need to be defined when using FEM:

- Step 1. Select a structural model. This includes choosing an appropriate mathematical model representing the physical problem being studied (e.g., specifying material properties and constraints). Two important properties of an appropriate mathematical problem are effectiveness and reliability [10].
- Step 2. Select a discretization. Create nodes and elements and define boundary conditions and loads. As discussed previously, the accuracy of the analysis depends on the discretization.
- Step 3. Compute the stiffness matrix ($\boldsymbol{K_i}$) and the load vector for each element. The stiffness matrix represents the relationship between the loads and the displacements at each node in an element.
- Step 4. Assemble the global stiffness matrix ($\boldsymbol{K}$) and load vector, compute the unknown displacements, the reactions, and the strains and the stresses for each element. Direct and iterative solvers are available, and the choice on which one to use depends on the problem.
- Step 5. Analyze the results, also known as the postprocessing step (e.g., analyze the displacements, the stresses and the strains). This step is crucial. The results of a simulation should always be checked.

B. *Error Recovery and Estimates*

Computational methods are applied to conceptual models of reality, and therefore can only compute approximate solutions. The main sources of error [10] are the model and the discretization. Strategies to minimize the error include improving the conceptual and the structural models and using a finer discretization. Because the conceptual and the structural models are in general not perfect, the simulation cannot reproduce exactly the real phenomenon even in a situation where the error is zero. In some problems, round-off errors introduced by finite precision arithmetic in computers can be significant. Several methods to estimate and reduce the error of a solution have been proposed (e.g., a posteriori error estimators and adaptive analysis procedures [12]). FEM provides error estimates and bounds that allow the use of adaptative self-correcting procedures.

C. *Solving Simultaneous Algebraic Equations*

A system of simultaneous linear algebraic equations can be solved using direct (elimination methods) and iterative or approximate methods. Iterative methods (e.g., the Gauss-Seidel method) are best suited to solve very large systems of equations, in general, avoid round-off errors, and can have convergence problems. Elimination techniques (e.g., Gauss elimination and the Cholesky factorization) can have round-off errors, and difficulties handling ill-conditioned systems that can lead to bad solutions or singularity [13].

A comparison between the Gauss-Seidel and the Gauss elimination methods, commonly used in practice, can be made to have an idea about the algorithmic complexity of these methods. For solving a system of n linear equations, the Gauss-Seidel method performs n divisions, n^2 multiplications, and $n^2 - n$ additions in each iteration, the Gauss elimination method uses n divisions, $(1/3)n^3 + n^2$ multiplications, and $(1/3)n^3 + n$ additions [14]. Other elimination and iterative methods are available (e.g., see [13], [14]). The choice of the method to be used depends on the problem being solved.

D. *Time-Dependent Analysis*

When working with time-dependent problems in dynamic analysis, procedures are required to perform numerical integration in time. In the case of nonlinear dynamic analysis, time integration algorithms can be implicit or explicit [14], [10]. Implicit algorithms satisfy equilibrium conditions at each increment (time step) and are said to be unconditionally stable. Explicit algorithms do not satisfy equilibrium conditions at each time step and are said to be conditionally stable. As a consequence, errors may be amplified during analysis. In order to satisfy equilibrium conditions, implicit algorithms use iterative methods. This makes them more expensive and can cause convergence problems but can act as an error correction mechanism. Because explicit algorithms are conditionally stable, time steps must be small enough to guarantee the accuracy and validity of the solution and avoid numerical instability. Implicit algorithms impose no limit on the size of the time step used but it still has an impact on the accuracy of the solution.

The choice on the approach to be used depends on the problem being solved (e.g., explicit algorithms are generally used to solve highly nonlinear problems with many degrees of freedom [15]). If a suitable time step is chosen both techniques converge to an accurate solution. There are also situations in which it is advantageous to use both techniques for different time steps [15].

E. *Nonlinear Analysis*

Linear analysis assumes that the shape and the material properties of the structure being simulated do not change significantly during deformation, displacements are infinitesimally small, no gaps or overlaps occur, the nature of the boundary conditions remains unchanged, and there is no time-dependence (In accordance with the steady state assumption [10].). The structure maintains its initial stiffness independently of the amount of deformation, stress developed in response to the load, and on how the load is applied. This assumption simplifies the problem formulation and its solution.

In nonlinear analysis a time-dependent non-steady state is assumed, and equilibrium must be achieved at all time steps. For example, assuming large displacements, rotations, and strains, for a body in motion, its volume, surface area, mass density, stresses, and strains can change continuously over time. Nonlinear problems are solved using iterative methods. This type of analysis does not always converge, and it is sensitive to small variations (perturbations) in the data. This makes it more complex and expensive. Some phenomena can only be simulated using nonlinear analysis, and some expertise is required to ensure the accuracy and the validity of the results. Nonlinear analysis allows the study of, for example, structural response to extreme events, performance under limit conditions and failure, impacts and large deformations, and phenomena that evolve dynamically. Sources of nonlinearities include [10]:

- Nonlinear geometry. Stiffness changes only due to changes in the shape of the geometry.
- Nonlinear material. Stiffness changes due to changes in the material properties during the analysis. Linear material models assume that stress is proportional to strain and that the model will return to its original shape once the load has been removed (i.e., no permanent deformations occur).
- Loss of elastic stability (buckling). Stiffness changes due to the applied loads. Nonlinear analysis can explain the post-buckling behavior of the structure (e.g., if it collapses or is still able to support the load after buckling).
- Contact stresses and nonlinear supports. Support conditions and contact stresses change during the application of the loads.

If large displacements, rotations, and strains occur, nonlinear analysis should be used. A problem can exhibit more than one type of nonlinear behavior [11]. Nonlinear analysis can be used if the nonlinear material properties of the structure being studied are known.

III. Using the Finite Element Method in the Context of Spatiotemporal Databases

Several types of analysis can be performed, each with its own advantages, disadvantages, and limitations, and there are situations in which it is advantageous to use more than one type of analysis to solve a problem. For example, when analyzing the evolution of icebergs, situations with: a) large displacements, rotations, and strains, b) small displacements and strains, and large rotations, and c) small displacements, rotations, and strains may be encountered. That is, we can potentially use different types of analysis and methods to study the evolution of a phenomenon. It is impractical to analyze the use of all possible types of analysis and FEM methods proposed in the literature. Therefore, the use of a general formulation, called the displacement-based finite element method [10] (based on the principle of virtual work), for the analysis of solids and structures is considered in the remainder of this section. The following is also considered:

- We are not interested in a full analysis (We are interested, in particular, in the displacements: translation and rotation, at the nodes.). This can simplify the analysis.

- FEM cannot be directly applied to the region interpolation problem because it cannot interpolate a geometry between two known geometries. It can however predict unknown states.
- Problems solved using FEM can have millions of degrees of freedom. This is not expected to occur in the context of spatiotemporal databases, for most problems.
- Meshes, matrices, vectors, and functions can be stored in a spatiotemporal database extension (e.g., for PostgreSQL) using abstract data types (ADTs). This includes, for example, the discretization, the stiffness matrix, the loads and the boundary conditions.
- In FEM, some boundary conditions must be set so that the system of equations to be solved has a unique solution (e.g., some displacements must be known).
- Each node in the discretization has at most three degrees of freedom: rotation, and translation in x and y.
- The use of optimized procedures is not considered, and in FEM terms, a few seconds can be considered a considerable amount of time.

In the simplest case, the governing equilibrium equations (corresponding to the nodal point displacements) for the static analysis of structures and solids, assuming linearity and n degrees of freedom, are given by [10]:

$$\boldsymbol{Ku} = \boldsymbol{r}, \tag{1}$$

$$\boldsymbol{Ku}(t) = \boldsymbol{r}(t), \tag{2}$$

where $\boldsymbol{r}$ is a vector of known loads or forces, $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{u}$ are the unknown nodal point displacements, and t represents time. $\boldsymbol{r}$, $\boldsymbol{u}$, and $\boldsymbol{K}$ are assembled from individual $\boldsymbol{r}^e_i$, $\boldsymbol{u}^e_i$, and $\boldsymbol{K}^e_i$ for each element i of the discretization. In (2) the displacements can be evaluated at any time t independently of the displacement and loading history. This is not the case in dynamic analysis [10]. Equations (1) and (2) can be solved using direct and iterative methods [10]. Iterative methods are usually used to solve very large systems of equations. In our context we assume n is much smaller than 1 million, therefore, we can use direct methods in most cases. Under the linear analysis assumption, $\boldsymbol{K}$ is constant. Therefore, $\boldsymbol{r}$ or $\boldsymbol{r}(t)$, and $\boldsymbol{K}$ or a factorization of $\boldsymbol{K}$ can be stored in the database and retrieved when necessary to compute $\boldsymbol{u}$ or $\boldsymbol{u}(t)$ at time t.

In the case of dynamic analysis, assuming linearity, the dynamic equilibrium equation has a characteristic form [10], [12] as in:

$$\boldsymbol{M\ddot{u}} + \boldsymbol{C\dot{u}} + \boldsymbol{Ku} = \boldsymbol{r}, \tag{3}$$

where $\boldsymbol{u} = \boldsymbol{u}(t)$ are the unknown nodal displacements, t represents time, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the damping matrix, $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{r}$ is a vector of known loads or forces, and $\boldsymbol{\ddot{u}}$ and $\boldsymbol{\dot{u}}$ are the nodal acceleration and velocity vectors, respectively. $\boldsymbol{C}$ is neglected in some types of dynamic analysis. Equation (3) can be solved using direct integration and mode superposition methods [10].

For example, if using an implicit integration method to solve (3) (e.g., the Newmark integration method that is unconditionally stable), with constant mass, time step, and material properties, and no damping, the displacements in the next time step ($t + \Delta t$) are computed using information about previous time steps. $\boldsymbol{M}$, $\boldsymbol{K}$, $\boldsymbol{r}$, the initial conditions ${}^0\boldsymbol{\dot{u}}$, and ${}^0\boldsymbol{\ddot{u}}$, the integration constants, and a factorization $\boldsymbol{\check{K}} = \boldsymbol{K} + a_0\boldsymbol{M} + a_1\boldsymbol{C}$ (where a_0 and a_1 are integration constants) can be stored in the database. Then, at each time step 1) the effective loads (${}^{t+\Delta t}\boldsymbol{r}^+$) are computed, 2) the system of equations $\boldsymbol{\check{K}}\,{}^{t+\Delta t}\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r}^+$ is solved for ${}^{t+\Delta t}\boldsymbol{u}$, and 3) ${}^{t+\Delta t}\boldsymbol{\ddot{u}}$ is computed. Whether or not some components can be computed once and stored in the database depends on the method used to solve (3), and how the problem is defined.

In the case of nonlinear analysis, using an updated Lagrangian formulation based on the principle of virtual work for general nonlinear analysis, assuming large displacements, rotations and strains (the area and the volume of the geometry change continuously), no nonlinearities in the boundary conditions, a negligible damping effect, displacement degrees of freedom only, and deformation-independent loads, the governing equilibrium equations are given by [10]:

$$({}^t\boldsymbol{K}_L + {}^t\boldsymbol{K}_{LN})\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r} - {}^t\boldsymbol{f}, \tag{4}$$

$$\boldsymbol{M}\,{}^{t+\Delta t}\boldsymbol{\ddot{u}} + ({}^t\boldsymbol{K}_L + {}^t\boldsymbol{K}_{LN})\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r} - {}^t\boldsymbol{f}, \tag{5}$$

$$\boldsymbol{M}\,{}^t\boldsymbol{\ddot{u}} = {}^t\boldsymbol{r} - {}^t\boldsymbol{f}, \tag{6}$$

for a static analysis (4), a dynamic analysis using implicit time integration (5), and a dynamic analysis using explicit time integration (6). Where ${}^t\boldsymbol{K}_L$ and ${}^t\boldsymbol{K}_{LN}$ are the linear and nonlinear strain incremental stiffness matrices at time t, ${}^t\boldsymbol{r}$ and ${}^{t+\Delta t}\boldsymbol{r}$ are the vectors of the external applied point loads at times t and $t + \Delta t$, ${}^t\boldsymbol{f}$ is a vector of nodal point forces equivalent to the element stresses at time t, $\boldsymbol{M}$ is a time-dependent mass matrix, $\boldsymbol{u}$ is a vector of increments in the nodal point displacements, and ${}^t\boldsymbol{\ddot{u}}$ and ${}^{t+\Delta t}\boldsymbol{\ddot{u}}$ are vectors of nodal point accelerations at times t and $t + \Delta t$.

Nonlinear problems are solved iteratively for each time step. The iteration process starts with some initial known values from a previous time step. Which components can be stored in the database depends on the method used to solve the problem (e.g., implicit or explicit integration) and the characteristics of the problem (e.g., are the external loads deformation-independent?). Since the solution at a time step t depends on the solution of previous time steps, some precomputed time steps can be stored in the database to accelerate computation.

IV. Discussion

This section discusses the advantages and disadvantages of using FEM, and numerical methods in general, in the context of spatiotemporal databases, having as a reference the properties defined in Section I for an ideal interpolation function F^+.

The main advantages of using numerical methods include the following. Numerical methods:

- Can handle a variety of problems (e.g., fluids, and systems with complex geometries and interconnected components), provide useful error estimates and bounds, and error recovery strategies are known and can be used.
- Solve equations based on established laws and principles of physics (i.e., consider the physical properties of materials and the external conditions with which they interact, that can have an impact on their evolution).
- Can approximate the behavior of real-world phenomena with a high accuracy and predict unknown states.

The main disadvantages of using numerical methods include the following:

- Parameter values may have to be provided by the user (i.e., the process in general is not automatic), and the values chosen can have a significant impact in the accuracy and validity of the results.
- The most appropriate type of element to be used depends on the problem, and hybrid meshes can obtain better results in some situations. The ideal discretization depends on the problem and on what is being analyzed.
- Some problems can only be solved using iterative methods (e.g., nonlinear problems) which makes them more expensive. Nonlinearity is abundant in the physical world.
- In time-dependent problems the integration time step chosen can have a significant impact on the accuracy and validity of the analysis. Guidelines exist to find an optimum time step. In general, the shorter the time step the greater the accuracy. In some types of analysis, the solution for an arbitrary time step t depends on solutions from previous steps. For example, given a phenomenon evolving for 20 seconds. If we want to know its state at time step $t = 15s$, assuming that the state at $t = 0s$ is known, we can compute for $t = 15s$ directly from $t = 0s$ with more or less impact on the accuracy and validity of the solution. If, for example, the optimum time step for the problem is $1s$, then we would have to compute for $t = 1s, 2s, 3s, \ldots, 15s$. It seems reasonable that the optimum time step should be used. A possible solution for this limitation is to precompute and store intermediate states in the database.
- An improper choice of a structural model, using an inappropriate numerical procedure, or type of analysis, for example, can lead to "improperly posed", inaccurate or invalid solutions, that may be so subtle that cannot be perceived by a nonexpert. Therefore, some level of expertise is required. For exemple, nonlinear analysis requires a significant amount of expertise.
- The time spent in the pre-processing and post-processing steps of complex problems can largely exceed the time needed to compute a solution. These steps, in general, require user intervention. In the context of spatiotemporal databases, automatic processes are preferred.
- FEM cannot interpolate a geometry between two known states (i.e., it cannot be used directly in the context of the region interpolation problem).
- Unless a mathematical model is known for a problem and material being analyzed, one has to be constructed which is not a trivial task. This can limit the use of FEM to specific problems and types of materials. In general, the analysis is problem-dependent.
- Overall, FEM does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, possibly involving nonlinearities.

V. Conclusion and Future Work

The finite element method is a powerful tool, and care and some level of expertise are needed to use it properly. For every problem a structural model must be defined, which is not a trivial task. It provides error estimates and bounds, and error recovery strategies can be used. It considers the material properties of the phenomena and the external conditions with which they interact and that can have an impact on their evolution. In some situations, the time step used for analysis can have an impact on the accuracy and validity of the solution, and arbitrary time step displacements are computed using information from previous known (or computed) time steps displacements.

It is important to note however that in the context of spatiotemporal databases, we are interested on the evolution (changes) of the nodal displacements (i.e., the translation and rotation of the nodes of the geometry representing the phenomenon) over time, not on a full analysis, and a relatively small number of degrees of freedom (much less than 1 million) are expected to be found in most of the problems being solved. This can simplify or make a finite element analysis less expensive. On the other hand, situations with large displacements, rotations, and strains, and nonlinearities are expected to be encountered, and most problems are time-dependent. The goal is not to use spatiotemporal databases as a data management technology to support the finite element method.

The finite element method can be used in the context of spatiotemporal databases to predict unknown states of real-world phenomena. However, it cannot be used directly to interpolate a geometry between a source and a target known geometries. It can handle complex geometries, but its use is limited to problems and materials for which a mathematical model is known. It provides error estimates and bounds, can simulate the behavior of a phenomenon with high precision, and the level of accuracy can be adapted to the needs of the user. Overall, it does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, possibly involving nonlinearities. It requires input from the user, some level of expertise, and the results should always be interpreted and analyzed with care.

As is, the finite element method can be used in specific situations, but it does not provide a solution for the problem that we want to solve in the context of spatiotemporal databases. A possible line for research is to study how it could be combined with morphing techniques to improve the interpolation quality of the latter and how it could be used to create a ground truth. Some interesting questions are raised for future work and investigation on this subject:

- Study how morphing and numerical methods can be used together (e.g., to improve the quality of morphing techniques).
- Study the use of meshless methods. These methods avoid some of the problems associated with the use of a discretization.
- Create moving regions for a use case using the finite element method and study its performance and the quality of the representation.

ACKNOWLEDGMENTS

I thank the reviewers for their comments and suggestions. José Duarte has a research grant awarded by FCT with reference PD/BD/142879/2018. This work was performed in the context of the project POCI-010145-FEDER-032636.

REFERENCES

[1] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. Lorentzos, M. Schneider and M. Vazirgiannis, "A Foundation for Representing and Querying Moving Objects," *ACM Trans. Database Syst.*, 2000, vol. 25, no. 1, pp. 1–42.

[2] M. McKenney and J. Webb, "Extracting Moving Regions from Spatial Data," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 438–441.

[3] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider, "A Data Model and Data Structures for Moving Objects Databases," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 319–330.

[4] G. Heber and J. Gray, "Supporting Finite Element Analysis with a Relational Database Backend Part I: There is Life beyond Files,", Microsoft Research Technical Report, MSR-TR-2005-49, April, 2005.

[5] G. Heber and J. Gray, "Supporting Finite Element Analysis with a Relational Database Backend Part II: Database Design and Access,", Microsoft Research Technical Report, MSR-TR-2006-21, March, 2006.

[6] G. Heber, C. Pelkie, A. Dolgert, J. Gray, and D. Thompson, "Supporting Finite Element Analysis with a Relational Database Backend Part III: OpenDX Where the Numbers Come Alive,", Microsoft Research Technical Report, MSR-TR-2005-151, December, 2005.

[7] M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible shape interpolation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000, pp. 157–164.

[8] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM SIGGRAPH*, 2004, p. 399.

[9] H. B. Yan, S. M. Hu, and R. Martin, "Morphing based on strain field interpolation," *Comput. Animat. Virtual Worlds*, 2004, vol. 15, no. 3–4, pp. 443–452.

[10] K. J. Bathe, *Finite Element Procedures*, 2nd Ed. Klaus-Jürgen Bathe, 2016.

[11] S. S. Bhavikatti, *Finite Element Analysis*. New Age International, 2005.

[12] R. L. Taylor, O. C. Zienkiewicz, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 6th Ed. Elsevier, 2005.

[13] S. Chapra, C. and R. Canale, P., *Numerical Methods for Engineers*, 7th Ed. McGraw-Hill Education, 2014.

[14] D. Logan, *A First Course in the Finite Element Method*, 5th ed. Cengage Learning, Inc, 2011.

[15] L. Noels, L. Stainier, and J.-P. Ponthot, "Combined implicit/explicit time-integration algorithms for the numerical simulation of sheet metal forming," *J. Comput. Appl. Math.*, 2004, vol. 168, no. 1, pp. 331–339.

[16] J. M. Gere, *Mechanics of Materials*, 6th Ed. Bill Stenquist, 2004.