

Implementation of Meshless FEM for Engineering Applications

A. Seidl, and Th. Schmidt

Abstract—Meshless Finite Element Methods, namely element-free Galerkin and point-interpolation method were implemented and tested concerning their applicability to typical engineering problems like electrical fields and structural mechanics. A class-structure was developed which allows a consistent implementation of these methods together with classical FEM in a common framework. Strengths and weaknesses of the methods under investigation are discussed. As a result of this work joint usage of meshless methods together with classical Finite Elements are recommended.

Keywords—Finite Elements, meshless, element-free Galerkin, point-interpolation.

I. INTRODUCTION

THE Finite Element Method (FEM) is well established as the primary method for solution of partial differential equations in engineering science. One of its shortcomings still is the difficulty of automatic mesh-generation. Boundary Element Methods (BEM) completely overcome this difficulty; however exhibit serious problems in dealing with inhomogeneities. In addition BEM lead to an equation system with a full matrix thus giving up the advantages in treating sparse and positive matrices, as in using FEM. Meshless Finite Element Methods can potentially fill this gap, since they reduce the problem of mesh-generation to a mere point-set-generation but nonetheless maintain full ability to treat inhomogeneous and/or nonlinear problems. The resulting linear equation system to be solved also has a sparse structure like in the case of classical Finite Elements. In this work element-free Galerkin (Belytschko [1]) and point-interpolation method (Liu [2]) were implemented and tested concerning their applicability to typical engineering problems like electrical fields and structural mechanics.

Manuscript received July 31, 2007. This work was supported in part by the German Ministry of Research and Technology under Grant 1705503.

A. Seidl is with the Dept. for Engineering and Industrial Design, University of Applied Sciences, Magdeburg, Germany (phone: +49-391-886 4372; fax: +49-391-886 4372; e-mail: albert.seidl@hs-magdeburg.de).

Thomas Schmidt is with the Dept. for Civil Engineering University of Applied Sciences, Magdeburg, Germany (phone: +49-391-886 4372; fax: +49-391-886 4372; e-mail: thomas.schmidt@hs-magdeburg.de).

II. DESCRIPTION OF IMPLEMENTED METHODS

A. Equations

In this work electrical field problems and elasticity problems have been treated. As an example for a typical elliptic problem we introduce the Poisson Equation in two dimensions. This equation reads in its strong form:

$$\nabla(k\nabla u) = f \quad (1)$$

where k and f are given functions of x,y . The solution to be computed $u(x,y)$ denotes the electrostatic potential. Correspondingly the displacement would be computed in case of an elasticity problem. For all FEM addressed in this paper the solution will be based on the weak of Eq. (1):

$$I_B + I_P + I_r = \min \quad \text{and} \quad I_d = 0$$

with

$$I_B = \frac{1}{2} \iint_{\Omega} k(x, y)(u_x^2 + u_y^2) dx, dy \quad \text{bilinear form}$$

$$I_P = \int_{\Omega} f(x, y) u dx dy$$

$$I_r = - \int_{\Gamma_u} [\lambda u] ds \quad \text{and} \quad I_d = - \int_{\Gamma_u} [u - \bar{u}] ds$$

The EFG Method (Belytschko) [1] and PIM (Liu) [2] have been implemented together with classical Finite Elements within a fully consistent scheme. The concept of approximating the solution by a weighted set of shape-functions is common to these methods:

$$u^h(\underline{x}) = \underline{\Phi}^T(\underline{x}) \underline{u} \quad (2)$$

The component Φ_I of the Vector $\underline{\Phi}$ is called the shape-function attributed to node I . Usually these shape-functions take non-zero values in the influence domain of the respective node only. In classical FEM the shape-function $\Phi_I(x)$ is confined to the element that contains the node I . In meshless methods such rules apply in a more general way introducing the concept of “domain of influence” [2].

B. Element-Free Galerkin

EFG-shape-functions are constructed by interpolating the solution by polynomial functions with moving least squares. EFG uses a weight function attributed to node I $w_I := w_I(\underline{x})$ which is non-zero in the environment of node I. If a point \underline{x} is close enough to node I such that $w_I(\underline{x}) \neq 0$ then the node will be used to construct shape-function-values at point \underline{x} , otherwise not. The following summarizes characteristic properties of EFG:

- a weight-function must be attributed to any meshpoint defining its domain of influence and thereby the neighbourhood relationship.
- shape-function construction using moving least squares
- the number or neighbours of a point where the shape-function is being evaluated (typically a Gauss-point) must be greater than the number of coefficients of the basis polynomial.
- leads to fully conforming shape-functions.

For details of the derivation, implementation, especially the computation of partial derivatives of the shape-functions the reader is referred to [1,10]. Please note that the computational effort for evaluation of the shape-functions is considerably high.

C. Point-Interpolation Methods

A detailed description of the method is given in [2]. Notes on our implementation can be found in [10]. The main difference to EFG is the fact that the number of neighbor-nodes is equal to the number of basis functions. Therefore a weighting of nodes is not necessary. Instead of a regression polynomial we obtain an interpolation polynomial which leads us to a strongly simplified procedure. PIM is based on polynomial interpolation like classical FEM, however the choice of neighbor nodes of a point in the plane to be used for interpolation is not defined by any element. As a consequence neighbor-nodes and the degree of interpolation polynomial can be chosen freely without any changes in the mesh. The cost of this advantage is the loss of conformity or the additional measures (i.e. Penalty Methods [2]) to maintain conformity. We summarize PIM:

- no weighting of nodes
- no least squares but direct interpolation
- number of neighbour points used for interpolation must correspond to number of coefficients in the basis-polynomial
- conformity requires additional measures

D. Implementation of Classical FEM

Due to the parallels between PIM and classical FEM the latter was implemented as a special case of PIM. The difference consists only in a different definition of the neighbourhood relationship.

E. Combination-Elements

The full potential of meshless FEM can only be exploited by combination of multiple methods. As an alternative to a coupling of multiple domains divided by internal boundaries, the introduction of transition elements allows a flexible coupling of different methods within one domain. In a domain where mixed approaches have to be used, every meshpoint is attributed a shape-function type. If different type-indices are found within one element, shape-functions of all types in question have to be evaluated within the element. Consequently inside the element the shape-functions

$$\Phi_{i,j}(x, y)$$

have to be computed where i denotes the local node-number and j the shape-function type index. For superposition of the shape-function components weight-functions are used which constitute a partition of unity:

$$\sum_{j=1}^3 w_j(x, y) = 1$$

$w_j(x, y)$ here denotes the weight-function attributed to shape-function-type of node with local number j. The resulting combined shape-function can then be computed by:

$$\Phi_i(x, y) = \sum_{j=1}^3 \Phi_{i,j} \cdot w_j(x, y)$$

All of the above described shape-function types can be combined using this concept thus allowing flexible combination of all different methods including classical FEM. In contrast to coupling over domain boundaries the weight-function strategy realizes a smooth transition between regions where different shape-functions are used.

III. CLASS-STRUCTURE

The class structure proposed in this paper is shown in Fig. 2 as UML-class diagram. The diagram includes only the base classes. Classes derived from these classes, for example special element classes for different kinds of elements or special analysis classes to perform nonlinear static calculations are not included. Also utility classes that handle vectors, matrices etc. are not shown. Heart of the class structure is the CModel class. This is a container class for all parts/objects/classes a FEM model consists of. The base classes for the FEM model parts are CNode, CElement, CMaterial and CLoads/CLoadHistory. FE model and FE analysis are decoupled from each other. The CModel class "communicates" with the the CAnalysis class over a defined interface. The CAnalysis class represents the FEM processor by which the FEM model state is transformed from one model state to another. CAnalysis manages the whole algorithmic interactions of the CModel parts. The base classes for the FEM analysis are CSolutionAlgorithm, CSolutionIntegrator, CSolver. Here the design is similar to [12],[13]. A special class CGlobal_Data was designed to manage fast and simple

access to global FE model data for all classes that participates in the FEM calculation. CGlobal_Data is member of CModel. Every class in the whole class design can ask for a pointer to CGlobal_Data. For the pre- and post-processing of the FE model a graphical users interface (GUI) is developed that use the same base classes as the CModel class (CNode, CElement .. s. Fig. 3). The whole design is similar to a Model-View-Controll pattern which is a well know pattern in modern object orientated design.

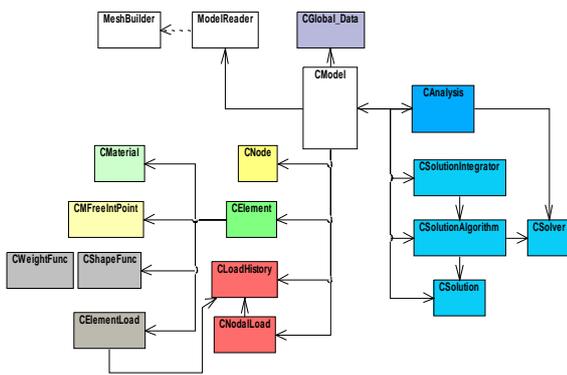


Fig. 2 UML-diagram of FEM and meshless FEM class design

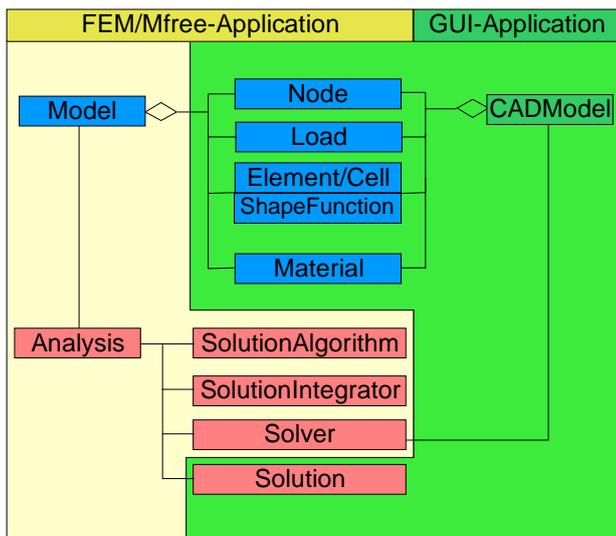


Fig. 3 Class-Coupling: FEM/Mfree-Application (processing) with GUI-Application (Pre-,Post-processing)

In classical FEM elements consist of a fixed number of nodes with a fixed integration area. In meshless methods the node number associated with a certain integration area is variable and the integration can vary. The design handles this by treating meshless methods as FE with variable nodes, variable integration point number and variable integration area. By this classical and meshless FEM can be handled consistently. Mixing classical and meshless methods within on FE model is easily possible. The major differences between

the two approaches are reflected in the classes to build the shape functions and to assemble the integration points for an integration area.

Special attention was paid to constructing the linear solver such that full modularity of the software is maintained. The current algorithms and data-structures for indexed storage of sparse linear equation systems require the sparsity-pattern to be known before the equation-system is set up. This is also true in the case of band- profile- and other sparse storage modes. But as soon as the linear solver needs mesh-related information to set up the sparse matrix structure full modularity is no longer maintained. This problem was solved by supplying a fully dynamic sparse matrix structure as an interface between Analysis- and Solver-class. Only after all coefficients of the linear system have been collected the solver creates the data structures specific for its particular solution method. This process is "hidden" within the solver-class and does not affect the interface. This implies that also a node-ordering e.g. by a Cuthill Mc Kee Algorithm is no longer executed on basis of the geometric mesh-structure but solely on basis on the connectivity of the DOF within the equation system. To represent this connectivity a graph-class was developed and set up completely independent on the geometric mesh. This algebraically motivated node-ordering has clearly greater flexibility since in many applications geometric connectivity (allocation of nodes to elements) does not correctly reflect the connectivity of the equations. For example in magnetostatics different components of the vector potential belonging to the same node or same element are not coupled within the equation system. The additional graph which is needed to carry out the node ordering such as Cuthill McKee is kept in memory only temporarily and does not affect the limits on problem-size to be treated.

IV. COMPUTATIONAL RESULTS

A. Test-Problems

In this section tests performed on electrical field problems are described. Different variants of point-interpolation methods (PIM), element-free galerkin (EFG) and classical Finite Elements were implemented a tested on different mesh-structures. As far as irregularly shaped structures were concerned a quadtree strategy was implemented for point-set generation. Point-sets were generated alternatively on basis of the center-points and on basis of the corners of the quadtree-boxes. In order to allow better classification a set of test-meshes was derived from a rectangular basis-structure. This was done to test the sensitivity of the respective method against the quality of the node-set used. The original equidistant rectangular structure was modified by applying linear-anisotropic, quadratic and random-distorsion.

B. Test of Implemented Methods

The mesh was varied between a completely regular mesh and a strongly degenerate one. The degree of degeneracy could be varied using a degeneracy-parameter between 0 (regular) and 1.2 (degenerated). For test purposes a Dirichlet

Problem was solved with a known solution: EFG was tested against classical FEM in order to investigate the sensitivity against degenerate node-sets. The errors on the nodes were evaluated; their arithmetic mean and maximum norm were computed. The results for the mean error norm are shown in Fig. 3. It can be seen that EFG method exhibits stable behaviour for degenerate meshes. Fig. 3 shows errors depending on the mesh-size. Fig. 4 shows errors depending on the above-mentioned mesh degeneracy-parameter.

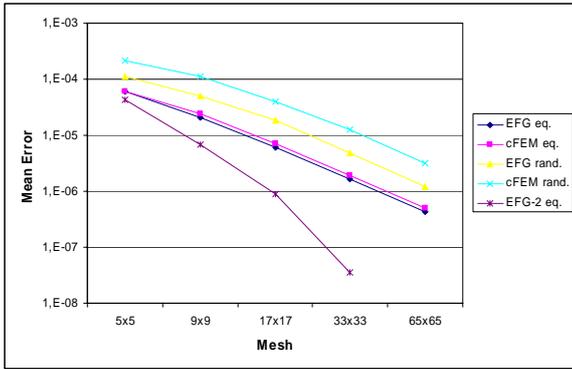


Fig. 3 Mean error of the test-problem versus mesh size for classical FEM (cFEM) and EFG. EFG-2 refers to shape-functions obtained with a 2nd degree basis function

The test of PIM was done in a similar way. Fig. 5 summarizes the test-results. Results obtained with classical FEM on different node-sets have been computed and are shown for comparison. Nonconforming PIM was found to exhibit poor solution quality and convergence. Only on rectangular meshes the accuracy could be obtained as to be expected for the high order shape-functions used. Our experiments have shown that in this case PIM becomes conforming for symmetry reasons, even if no measures are taken to enforce conformity. For meshes lacking these symmetry properties a penalty function strategy has been proposed by Liu [2] (CPIM). Consequently, results obtained using CPIM also are superior to those obtained with classical FEM. However, the penalty strategy requires the usage of an additional parameter, the proper choice of which also turned out to be difficult. A variation of PIM will therefore be shown in section V.

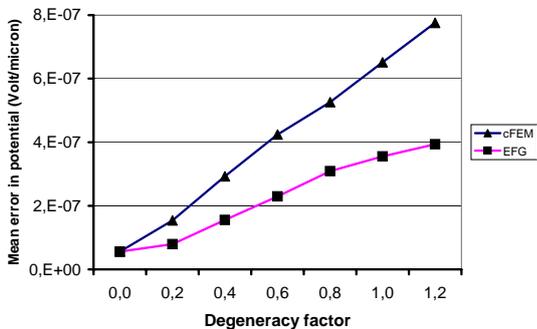


Fig. 4 Mean error of the test-problem versus mesh-degeneracy for classical FEM (cFEM) and EFG

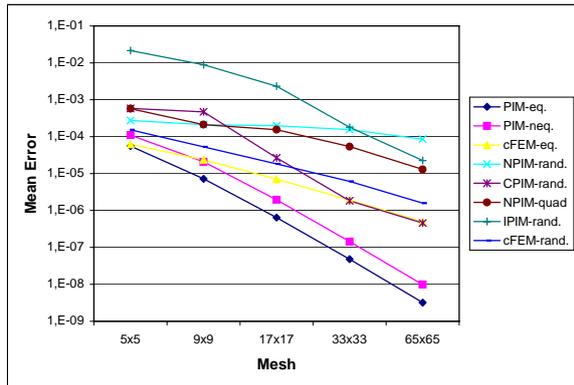


Fig. 5 Test of different variants of PIM depending on the mesh in comparison with cFEM

V. CONSTRUCTION OF QUADTREE-ELEMENTS

The meshless shape-function generation algorithm can be used to flexibly construct additional element-types for the classical method. This section describes a new element-type as is immediately created automatically arising from our implementation of the point-interpolation method. If a Quadtree approach is used for point-set generation it is convenient to immediately use the boxes as elements. As shown in Fig. 7 additional mid-side nodes appear in areas of mesh-refinement. As a result we have to deal with an irregular pattern of 4-, 5, 6-node elements. In this work the default elements were attributed a bilinear basis-polynomial with the option to add one or two quadratic terms depending on mid-side-nodes. The PIM shape-function generator (see section II. C.) constructs the shape-functions arising from the basis-polynomials in a straight-forward manner. Along the boundary between fine and coarse discretization linear and quadratic terms appear in directly adjacent elements. Since this approach leads to non-conforming elements tests were necessary to verify convergence. A near-singular problem with a known logarithmic function as the exact solution was used as test-case. The quadtree-boxes were divided depending on a

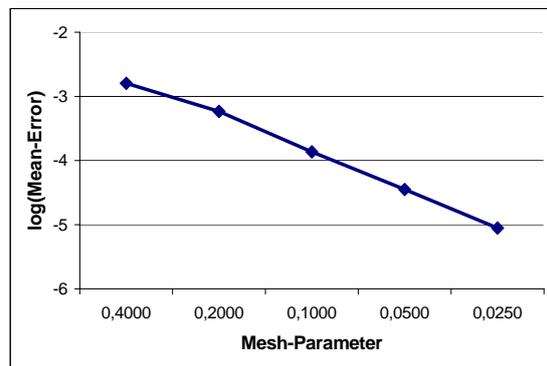


Fig. 6 Dependence of disc retization error from mesh-parameter

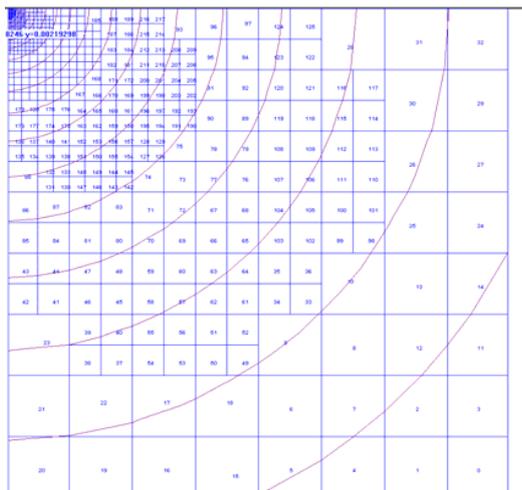


Fig. 7 Mesh directly generated from Quadtree. The singularity of the solution-function is near the upper-right corner

criterion which contains the gradient of the exact solution together with a mesh-parameter. Coarser and finer meshes could be constructed in dependence of the parameter. A division of the mesh-parameter by two means a four-fold increase of the number of boxes. Or in other words: At a given position the element-size will be approximately proportional to the mesh-parameter. Fig. 7 clearly shows extreme refinement close to the singularity. The deviation of the solution at the mesh-points (mean-value over all mesh-points) was evaluated as a measure for the achieved accuracy. Results are summarized in Fig. 6. Quadratic convergence is clearly shown.

VI. CONCLUSION

Meshless methods have been implemented in combination with classical FEM. By empirical tests EFG methods have been shown to be particularly resistant against the effects of odd geometries and the effects of extreme local mesh-refinement such as the appearance of single degenerate elements. However EFG that have been tested to be the most robust are considerably less efficient than classical Finite Elements. Point-Interpolation methods exhibit good efficiency, but are sensitive against variation of the point-set. They show excellent performance as long as the choice of the basis polynomial and the neighbourhood relationship can be adapted in an optimal way to the local node-pattern. Here the authors see much potential for possible future developments which was tried to demonstrate by the experimental 4,5,6-node quadtree-element setup. At the present state of the art a combination of meshless and classical FEM (e.g. with local use of EFG) is proposed as a result of this work. An object oriented flexible class structure to meet this requirement has been presented in this paper.

REFERENCES

- [1] Belytschko, T.; Lu, Y.Y. and Gu, L.: „Element-Free Galerkin Methods“, International Journal for Numerical Methods in Engineering, vol. 37, 1994, pp.229-256.
- [2] G.R. Liu: *Meshfree Methods - Moving beyond the Finite Element Method*, 712 pages, 2002, CRC Press. ISBN: 0849312388.
- [3] Beissel, S.; Belytschko T.: „Nodal Integration of the element-free Galerkin Method“, Computer Methods in Applied Mechanics and Engineering, 1996, 139: 49-74.
- [4] Martin Larcher private communication
- [5] Zienkiewicz, O.C.: „Methode der finiten Elemente“, Carl-Hanser Verlag, München 1984.
- [6] Chen, J.S.; Wu, C.T.; Yoon, S.; Yu, J.: „A stabilized conforming nodal integration for Galerkin mesh-free methods“, Int. J. Numer. Meth. Engng. 2001; 50:435-466.
- [7] Schmidt, Th.; Abschlussbericht: Effektive Implementierung gitterloser Finite Elemente Methoden, Forschungs-semester, Abschlußbericht.
- [8] Seidl, A; Schmidt, Th.: "Gitterlose FEM", Abschlußbericht des Forschungsvorhabens FKZ 1705503.
- [9] Seidl, A.; Schnattinger, Th.; Erdmann, A.; Hartmann, H; Petraschenko, A.: "Accurate extraction of maximum current densities from the layout", presented at IWCE 11, Vienna, 2006.
- [10] Seidl, A., Schmidt Th., Orlando 2005.
- [11] Seidl, A., Schnattinger Th., Erdmann A., Hartmann H., Petraschenko A.: "" Journal of computational electronics.
- [12] McKenna, Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing, PhD, University of California at Berkeley, 1997.
- [13] Archer Graham Charles, Object-Oriented Finite Element Analysis, PhD, University of California at Berkeley, 1996.