FINITEPOINT2D: A MESHLESS GiD CODE FOR LINEAR ELASTICITY

Enrique Escolano and Franco Perazzo
International Center for Numerical Methods in Engineering
Universidad Politécnica de Cataluña, Edificio C1
Gran Capitán s/n, 08034 Barcelona, Spain

Key words: Finite Point Method, Meshless Method, Elasticity, GiD

Abstract

The Finite Point method, a meshless formulation, is applied to the classic linear elastic 2D problem. This new method is implemented as a GiD problem type to automatically generate the clouds and visualize the results.

1. INTRODUCTION

The meshless methods are developed to solve the difficulties to generate the mesh for finite element methods, especially in 3D cases. The finite point method is based on a approximation of the unknown function, on a cloud around a star node, by least weighted squares. The differential equation to solve is sampled in each point to construct a linear system of equations, similarly to finite difference techniques. This method is fully meshless (it not requires a local auxiliary mesh for integration as other methods).

2. FUNCTION APPROXIMATION

The unknown function *u* is locally approach near a star node with a linear combination of base functions [$f_1,f_2,...,f_m$]. For this 2D case, the program use the monomial base with *m*=6, [$1,x,y,x^2,xy,y^2$]

$$u(x,y) \cong \hat{u}(x,y) = \sum_{i=1}^{m} \alpha_i \cdot f_i(x,y) \qquad (1)$$

The coefficients α_i are determined minimizing a summatory of square errors on the local nodes {$p_1,..p_n$} of the cloud. The error is weighted by a function $\varphi(x,y)$ to enhance the precision near the star node.
The functional *J* to minimize respect α_i is

$$J = \sum_{j=1}^{n} \varphi(p_j) \cdot (u(p_j) - \hat{u}(p_j))^2 = \sum_{j=1}^{n} \varphi(p_j) \cdot (u(p_j) - \sum_{i=1}^{m} \alpha_i \cdot f_i(p_j))^2 \qquad (2)$$

Solving the normal equations linear system, the coefficients obtained are:
Notation: $\overline{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$, $\overline{f} = [f_1, \cdots, f_m]^T$, $\overline{u} = [u(p_1), \cdots, u(p_n)]^T$

$$\overline{\alpha} = \underline{\underline{C}}^{-1} \cdot \overline{u} = \underline{\underline{A}}^{-1} \cdot \underline{\underline{B}} \cdot \overline{u} \qquad (3)$$

with

$$\underline{\underline{A}} = \sum_{j=1}^{n} \varphi(p_j) \cdot \overline{f}(p_j) \cdot \overline{f}(p_j)^T$$

$$\underline{\underline{B}} = [\varphi(p_1) \cdot f(p_1), \cdots, \varphi(p_n) \cdot f(p_n)]$$

3. CLOUDS

For each node, the approximation is local, only depends of the values on the cloud nodes (collection of n near nodes).
A cloud requires a minimum number of nodes n≥m (dimension of the space base functions), and also a maximum for computational pourposes.
Another requirement is the non-singularity of the "A" matrix of the equation (3) (for example, a collection of aligned nodes is invalid).
A simple selection of the points inside a sphere centered at the star node is not generally a good choose.
Near the boundary, a cloud must not contain nodes physically unconnected with the star node. A strategy to construct the clouds from the list of all points can be consulted at [2].
In order to apply the boundary conditions for the EDP's it is also required the boundary normals.

The FPM is fully meshless, it only requires clouds and boundary normals, but in this problem type implementation, it can read a 2D mesh from GiD to build the clouds and normals.
To select nodes for the cloud, the edge-connected nodes are used: first the directly connected with the star node, and then the next level and so on until the desired number of nodes is reached.
For a proper version, GiD creates directly only nodes (and a boundary mesh for normals). It is a contradiction to develop a meshless method to bypass the effort of generate elements, and create the clouds/normals from an auxiliary mesh.

4. LINEAR ELASTICITY

In the Finite Point method, the differential equations are sampled by puntual collocation at each node, using for the displacement the approximation (1)

In this program a stabilized formulation is used to avoid ill-conditioned system of equations typical of puntual collocation. If the differential equations are:

$$A(u_j)=0 \quad \text{inside the domain } \Omega \qquad (4)$$

The stabilized formulation appends terms based in the real finite dimensions of the cloud. The stabilized form is:

$$A - \frac{1}{2} h_k \frac{\partial A}{\partial x_k} = 0 \qquad (5)$$

with h_k characteristic length parameter dimension (*k=1,2* for 2D case)
For elasticity problems the stabilized equilibrium equations are:

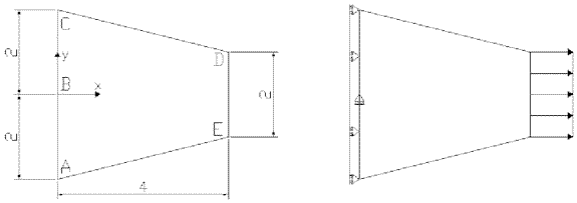
$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i - \frac{1}{2} h_k \frac{\partial}{\partial x_k}\left(\frac{\partial \sigma_{ij}}{\partial x_j} + b_i\right) = 0 \qquad \text{in } \Omega \qquad (6a)$$

$$u_j - \hat{u}_j = 0 \qquad \text{in } \Gamma_u \text{ (dirichlet)} \qquad (6b)$$

$$\sigma_{ij} n_j + t_i - \frac{1}{2} h_k n_k \left(\frac{\partial \sigma_{ij}}{\partial x_j} + b_i\right) = 0 \qquad \text{in } \Gamma_t \text{ (neumann)} \qquad (6c)$$

5. EXAMPLES

Source of the test: NAFEMS, linear statics benchmarks vol.1, October of 1987, test IC1.
This sample is a plane stress case, geometry dimensions are in meters and the boundary conditions are: AC locked in *x* direction and B locked in *y*.

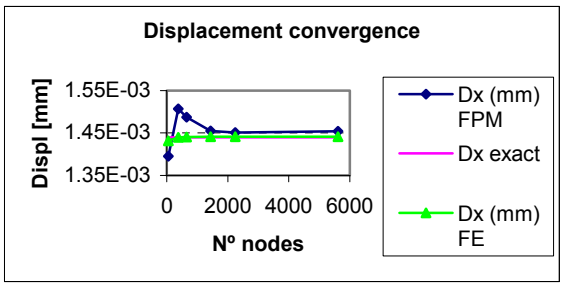


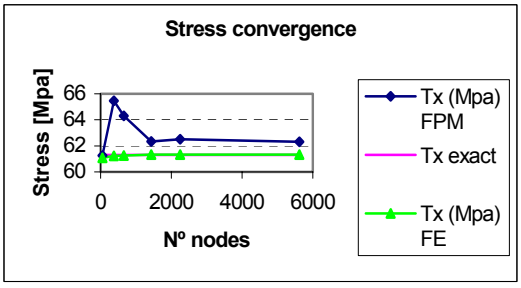
Material: Isotropic, E=210103MPa, v=0.3, thickness=0.1m
Aplicated charge: load of 10MN/m applied over DE border

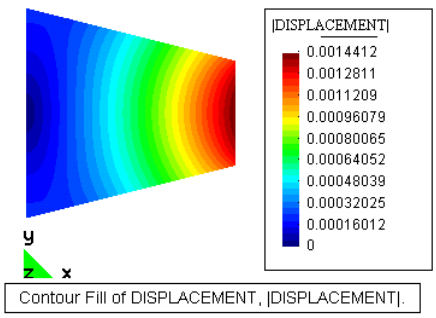
Wished result: U=1,44 mm in the mid of DE border - Sxx=61,3MPa in B

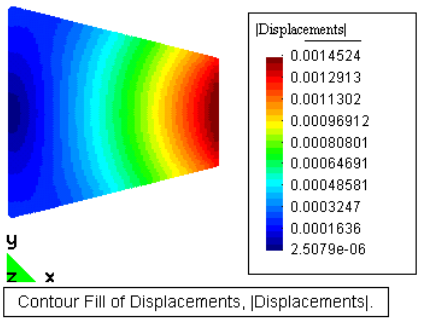
Obtained results:

Nº nodes	Dx (mm) FPM	Dx (mm) FE	Tx (Mpa) FPM	Tx (Mpa) FE
54	0.001395	0.0014316	61.266	61.070
377	0.001507	0.0014398	65.471	61.240
652	0.001487	0.0014406	64.307	61.262
1432	0.001455	0.0014411	62.336	61.318
2246	0.001451	0.0014410	62.513	61.320
5614	0.001452	0.0014412	62.306	61.333









6. CONCLUSIONS

To evaluate the FPM, it has to be compared with the classical Finite Element Method (FEM), not only in accuracy of results but also in the practical and computational aspects.

The disadvantages of the FPM are:
- The generation of clouds from the nodes is complicated, and the “standard” preprocess programs actually are not ready for this requirement.
- Unsymmetrical stiffness matrix (the memory occupation increase, and some classical algorithms are disabled, for example, a direct cholesky decomposition can not be used to solve the linear equations system)
- ill-conditioned linear systems (convergence problems for iterative solvers as biconjuqate gradients and bad precision of some results)
- Construction of “Shape functions” more expensive (inversion of a $_{mxm}$ matrix).
- Cannot apply punctual forces (the formulation works directly with stresses) and cannot obtain reactions (the domain is diffuse without a mesh).
- Problems to sample the EDP if exist more than one material.
- Undefinition of required boundary normals in some edges or vertex (bad results near this boundary).
- The graphical postprocess is more limited without elements (cannot represent well defined contour fill of colours, integration on cuts, etc)

The main advantages are:
- Savings in the mesh generation (the main advantage of the method)
- Easy increment of the interpolation degree (only increasing the number of monomials of the base functions)
- Can work with high order derivatives of the unknown function (of interest when solving EDP’s involving these derivatives)

Concluding, for a 2D elastic case the cost of the mesh generation is low, and the finite FEM is actually more competitive than the FPM, but it is of interest for a complicated 3D volume, or evolutive shape (big displacements, shape optimization, damage models, fluid dynamics, etc.).

7. REFERENCES

[1] E. Oñate, F. Perazzo and J. Miquel. A finite point method for elasticity problems. Computer and Structures 79 (2001).
[2] R. Löhner, C. Sacco, E. Oñate and S. Idelsohn - A Finite Point Method for Compressible Flow. *ECCOMAS 2000*, Barcelona, September (2000).
[3] M. Vericat. Análisis de estructuras en el campo elástico mediante métodos sin malla. Tesina de especialidad. UPC. Septiembre 2000.
[4] A. Huerta and S. Fernández. Coupling element free Galerkin and finite element methods. *ECCOMAS 2000*, Barcelona, September (2000).
[5] GiD Reference Manual. International Center for Numerical Methods in Engineering (CIMNE).