Discretization Error Control In the Precise Solids Method (PSM)

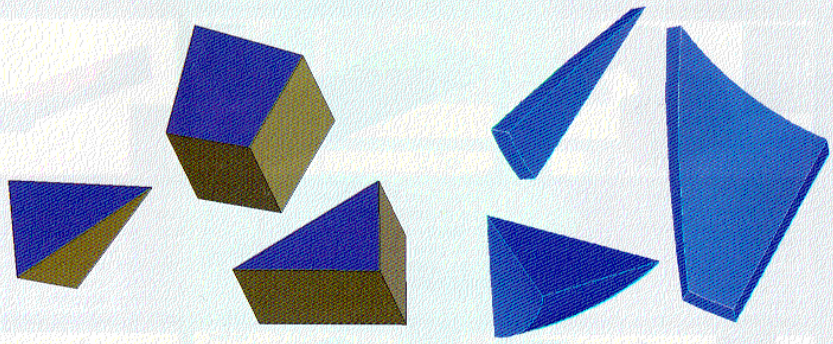
Dr. Victor Apanovitch & Dr. Paul Kurowski

Introduction

To assure that a numerical analysis produces results with required accuracy, we need tools to analyze and control discretization errors. In FEA, the discretization error analysis is commonly called convergence analysis. It examines the impact of different discretization choices on the data of interest such as displacements, stresses, temperatures, etc. Convergence analysis involves several iterations with a systematic change in the choices of discretization: FEA h method requires several mesh refinements with elements of the same order, FEA p method requires several runs with the same mesh but different element order. In either case, the convergence analysis is expected to demonstrate that the data of interest are sufficiently independent of the further refinement or further element order upgrade. Results are typically presented in the form of convergence curves representing the data of interest as functions of the discretization choice. We like to see those curves tending to finite, asymptotic values, hence demonstrating a "well converged solution", generally accepted as an assurance of good quality of results.

Typical convergence analysis indeed provides very valuable insight into the quality of results, but it indicates only **relative** change of data of interest during the iterative process. Cases of false convergence may slip through unnoticed. The convergence analysis can not calculate the **absolute** discretization error.

Calculation of absolute discretization error is made possible by the emerging analysis technology, the Precise Solids Method (PSM),which is based on the External Finite Element Approximation Method [1] [2] as implemented in commercial software Procision. The PSM takes error analysis to the next level. It offers the analysis of relative changes of data of interest, just like traditional FEA, as well as the analysis of absolute errors. Even though Procision best handles error analysis when left on "auto-pilot", it can be made to behave either as an h-code or as a p-code. So before we present its approach to the error analysis, we will investigate Procision capabilities of performing traditional h and p convergence. We will also take this opportunity to review major similarities and differences between FEA and PSM.



h - elements *p - elements*

Figure 1 Elements are restricted to simple shapes in FEA

Procision performing h convergence

h convergence requires that a model be meshed with several increasingly refined meshes. However, Procision is a meshless technique and does not use elements, it uses subparts instead. For that reason, before we demonstrate Procision performing h convergence, we need to review essential differences between an element and a subpart and how a subpart can be made to behave as an element.

The objective of discretization, both in FEA and in PSM, is to allow reasonably simple approximation functions to describe field variables in each sub-domain,an element in FEA and a subpart in PSM. In FEA, those approximation functions are exclusively of polynomial types and must conform (show continuity of displacements) on elements' boundaries. The conformity requirement restricts element shapes to simple primitives like tetras, wedges and bricks. Even though the higher order polynomials (up to 9th) available in p codes do allow for some freedom in mapping, elements are still limited to those basic shapes (fig. 1).

Procision also uses polynomial type functions, called here volume functions, for approximating field variables in subparts. These volume functions are designed to fulfill equilibrium conditions in the volume

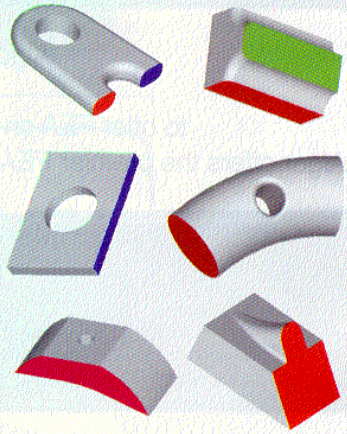


Figure 2
Subparts can be of any shape in PSM

and, at the same time, not to restrict subpart shape (fig. 2).

Subparts can be of any shape whatsoever and subpart degeneration does not apply since no mapping is performed. As we will explain later, the exact fulfillment of equilibrium makes it possible to calculate absolute boundary condition (BC) errors, which is the essence of discretization error analysis in the native Procision mode. However, satisfying equilibrium in the volume and freedom to use any subpart shape, comes at a price of not satisfying displacement boundary conditions and displacement continuity across subpart dividers exactly. Therefore, displacement BC's, and displacement continuity across subpart dividers are subjected to the control of specialized boundary DOF (degree of freedom) functions. As opposed to conventional FEA, in PSM DOF's are not nodal displacements, and in PSM specific boundary DOF's have no physical meaning. They assure the exact compatibility of displacements and tractions between subparts only in limit, when DOF tend to infinity (p type becomes infinitely high).

For any finite polynomial order of boundary DOF functions, the compatibility conditions are satisfied only approximately and the iterations continue until the user specified accuracy is achieved. Also, this approximate fulfillment of displacement BC's and displacement continuity, as controlled by boundary DOF functions, allows for freedom of defining subparts without matching adjacent subparts faces. We will take advantage of this when constructing models for this h convergence exercise.

Note that while a discontinuous displacement field is considered a "variational crime" in FEA, it is permissible in PSM because those discontinuities are under control exercised by boundary DOF functions.

So far we have identified two types of functions used in Procision: volume functions and boundary DOF functions, both of polynomial type, but Procision is not limited to only polynomial type representations of field variables. If it is necessary to keep BC both displacement and traction errors low, Procision can also use non-algebraic functions to represent stress concentrations. The discretization in PSM is still necessary, except for very simple models, because we are obviously limited by the finite p order of volume and boundary DOF functions and by the finite complexity of stress concentration functions.

Lets now go back to our exercise in h convergence. If we wish to run Procision in h code mode, we have to make subpart behave as an element, and, an assembly of subparts, the split set behave as a mesh. This is done in three steps. First we split the model so that the split set resembles an h mesh. Next, we freeze the p-order of all subparts at the lowest available level, which is p = 2. Finally, we instruct Procision not to use stress concentration functions. This way we roughly represent a mesh of the second order elements. The same has to be repeated for several increasingly fine splits sets, a rather tedious but straightforward process.

We choose simple model: a tensile strip with a single central hole (200x100x10mm, hole 40mm diam.), known to have a smooth solution. The material is steel (E = 200,000 MPa, n = 0.3); the model is rigidly supported on one side and loaded with the uniformly distributed tensile load of 100,000 N applied to the opposite side.

For completeness we start with no splits. The results (fig. 3b) show that only the global stress is captured and hole is ignored. The refinement process is illustrated in fig. 4, ...5. It is conducted in six steps, but only three are shown for conciseness.

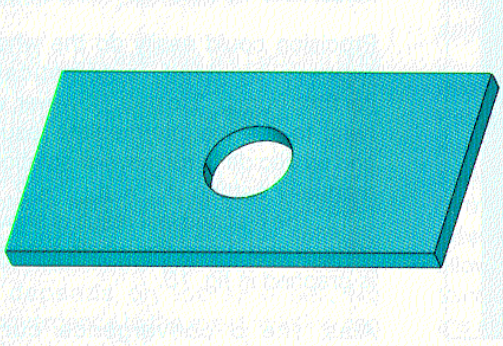


Figure 3a: No division into subparts

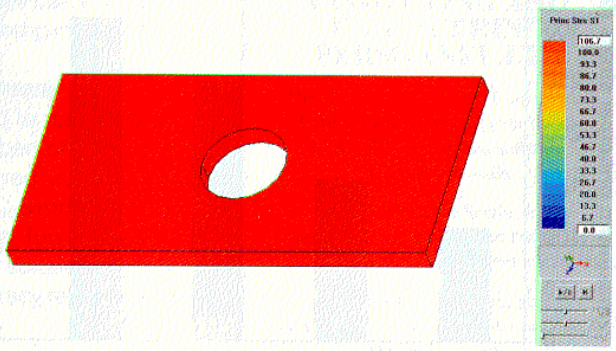


Figure 3b: Max. principal stress = 107 MPa

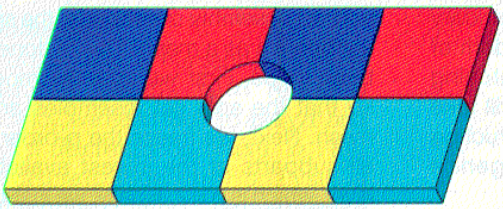


Figure 4a: 8 subparts

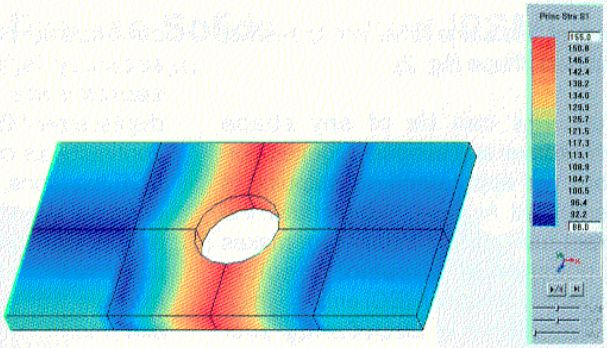


Figure 4b: 8 Subparts Max. principal stress = 155 MPa

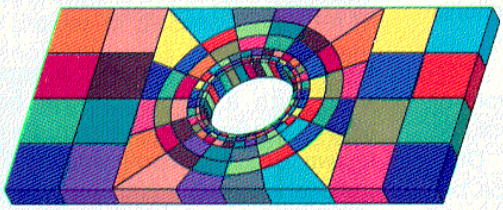


Figure 5a: 80 subparts

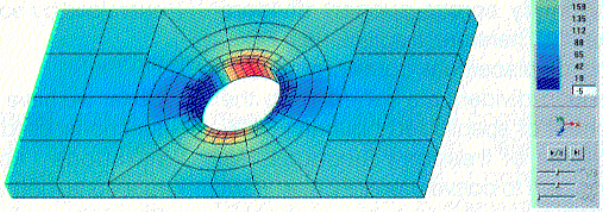


Figure 5b: 80 Subparts Max. principal stress = 370 MPa

Note, that even though we ask Procision to pretend to be an h code, we still take advantage of freedom to use arbitrary subparts shapes like seven sided "chunks" (fig. 4a.) and do not match faces of adjacent subparts (fig. 5a).

We summarize the results of the h convergence study in fig. 6 demonstrating a nice, "well converged" solution and note the ability of Procision to perform a correct h convergence analysis.

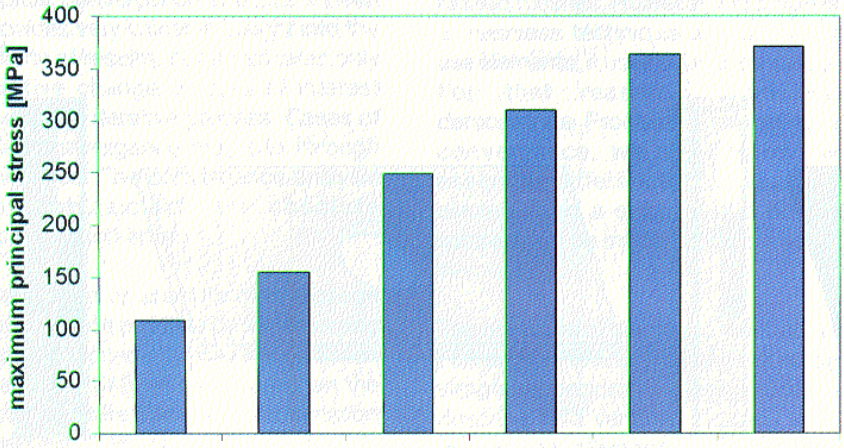


Figure 6 Results of h convergence study

Procision performing p convergence

Let us now see if Procision can equally successfully replicate p convergence process. Using the same model, we start with a split set intended to resemble a manually created p element mesh (fig. 7).

This time we want to use the whole range of p orders available in Procision: p = 2, ...12. Again, we do not want to use stress concentration functions, thus limiting Procision to polynomial only representation of displacement fields, as is the case in p method FEA[4].

Procision could easily do the whole process by itself, automatically iterating from p=3 to p=9. However, we need to see the intermediate results and so decide to run several solutions manually setting $p_{min} = p_{max} = 2, 3, ...12$. The first and last result is shown in fig. 8 and 9 and the whole process is graphed in fig. 10.

Alas, the p convergence curve depicting the change in the maximum principal stress, shows no signs of flattening; let us explain why?

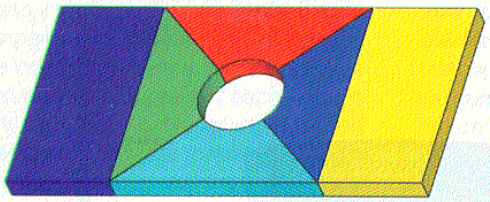


Figure 7: Split set resembling a p element mesh

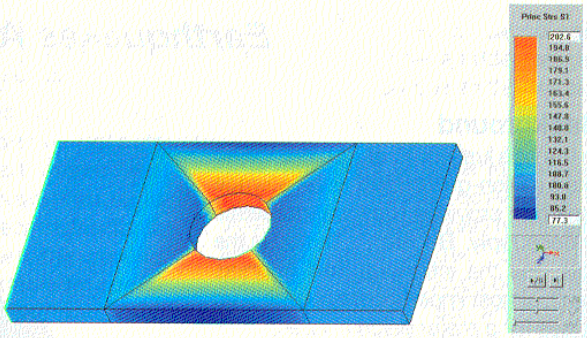


Figure 8: P=2 Max. principal stress = 203 MPa

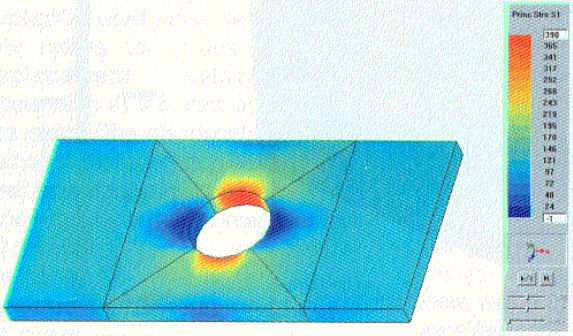


Figure 9 : P=12 Max. principal stress = 390 MPa

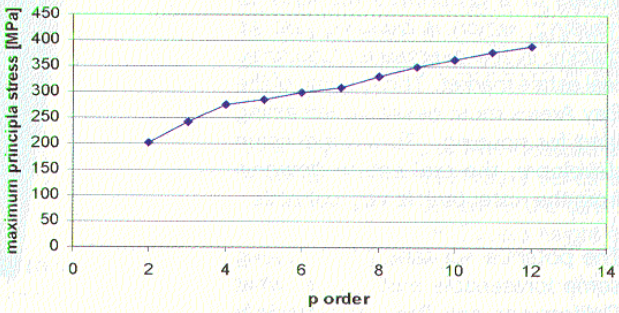


Figure 10 : Results of p convergence study

At the beginning of every step of the manually executed p convergence process, Procision initializes boundary DOF functions and volume functions. Procision user's interface allows for control of p-levels of volume functions by setting the desired $p_{min} = p_{max} = 2,3,$ ...(in h convergence exercise we set p =2), but it does not allow control of the boundary DOF functions. For example, two runs with $p_{min} = p_{max} = 2$ and $p_{min} = p_{max} = 3$, both have the same set of boundary DOF functions, because in both cases this is the first iteration, no matter what p is equal to.

Since in our exercise we perform only one iteration, the boundary DOF functions remain the same set, no matter what we set $p_{min} = p_{max}$ to be equal to. The run with higher p-level volume functions but with boundary DOF functions still at initial settings, can yield more or less accurate results, it depends on the split set. No convergence to the exact solution can take place in our exercise preventing us from using Procision to replicate a manual p convergence process.

p convergence could be replicated if we allowed automatic iterations and did not use stress concentrations functions. Having gone through several iteration, the boundary DOF function would be able to adapt the required p level and gain control of displacement BC errors. However, we are using h and p convergence study only as a convenient way to introduce the reader to error analysis concepts in Procision.

In the next article we will discuss how Procision should really be run! Meanwhile perhaps we can whet your appetite with the following introduction.

Procision in native mode

Procision is designed to run using the least possible number of subparts, specifying the desired accuracy in terms of absolute BC errors, enabling stress concentration functions and allowing the iterative solution process to continue until accuracy requirements are satisfied.

For Further Info Contact:
pkurowski@odyssey.on.ca

References

1. Apanovitch Victor; The Method of External Finite Element Approximation, Minsk, (1991) ISBN 5-339-00597-6
2. Colin Andre, Endersby Stephen; Procision – A Brief Technical Background, Benchmark, July 1999