Can we do better than the standard FE error estimates? The answer is yes if you intelligently construct the mesh. It is clear that if you grade the mesh so there are smaller elements in area where the “action is” and large elements where there is “little to no action” you get a better answer for the same number of elements. But is it possible to even obtain better convergence rates?

There are two aspects to how you may actually get better convergence. One is the method you use to improve the mesh (typically referred to as refinement). The error estimates we have so far are for uniform h-refinement. However, we can do p-refinement in which we increase the polynomial order.

We can also be “intelligent” about how we refine all the way to the development of “optimally” refined meshes. It turns out that to get to an optimally refine mesh for general problems one has to employ *a posteriori* error estimates and associated mesh correction indicators within and adaptive analysis loop.

Common methods of mesh enrichment

- Uniform h-refinement – Subdivide element into smaller elements. The common method is to cut all edges in half.
- Adaptive h-refinement – employ an adaptive method to refine the mesh as needed to be optimal (e.g., to equilibrate the error per element which is optimal for linear self adjoint problems).
- Uniform p-refinement – increase the order of the elements in a fixed mesh topology.

- Adaptive p-refinement – selectively increase the order of elements in a fixed mesh topology as dictated by an adaptive method that drives the mesh to an optimal element order distribution.
- Adaptive hp-refinement – selectively refine elements and selectively increase polynomial order, as driven by an adaptive procedure, to create an optimal distribution of element sizes and order. Such a procedure needs both local error estimates and local correction indicators that can indicate which of the two methods to enrich the mesh should be applied locally.

Note to compare the different methods of mesh enrichment we need a common basis for defining convergence – we only have an equation that has h^{α} where α is a function of p. The ideal would be total solution time. However, this is a strong function of details of implementation of the numerical procedures used to solve the finite element problem.

The closest we can come-up with as a common basis for describing the convergence is the total number of unknowns which we will denote as N. The main problem with N is that it does not account for the fact that as you increase p the stiffness matrix becomes less sparse that when you decrease h. However, for a good selection of numerical methods used, the influence is not that noticeable.
The Szabo and Babuska text book provides information on the comparison of the methods:

For $n_{sd} = 2$

$p = k =$ order of complete polynomial of elements

$\lambda =$ a measure of the strength of a singularity if one exists

$\beta =$ convergence rate

note: p controls for problems with a smooth exact solution and λ controls if the exact solution is not smooth,

Convergence rates:

Uniform h-refinement

$$\|e\|_E \leq \frac{c}{N^{\beta}} \ , \ \beta = \tfrac{1}{2} \min(p, \lambda)$$

Where $\|e\|_E = a(e,e)^{1/2}$

Adaptive h-refinement

$$\|e\|_E \leq \frac{c}{N^{\beta}} \ , \ \beta = \tfrac{1}{2} p$$

basically eliminate the influence of the singularity

Uniform (or adaptive) p-refinement for smooth problems

$$\|e\|_E \leq \frac{c}{\exp(\gamma N^{\theta})} \ , \ \gamma \text{ is a constant, } \theta \geq \tfrac{1}{2}$$

Uniform p-refinement for singular problems

$$\|e\|_E \leq \frac{c}{N^{\beta}}$$

$\beta = \tfrac{1}{2} \lambda$ if the singularity in inside the element, $\beta = \lambda$ if the singularity in on element boundary.

Adaptive hp-refinement for singular problems.

$$\|e\|_E \le \frac{c}{\exp(\gamma N^{\theta})}\ ,\ \gamma \text{ is a constant, } \theta \ge 1/3$$

This case requires very carefully constructed meshes. For example for a 1D $1/\sqrt{r}$ singularity, what one has with a linear elastic crack tip, a geometrically graded mesh is needed with a factor of 0.15. That is the first element is 85% of the domain.

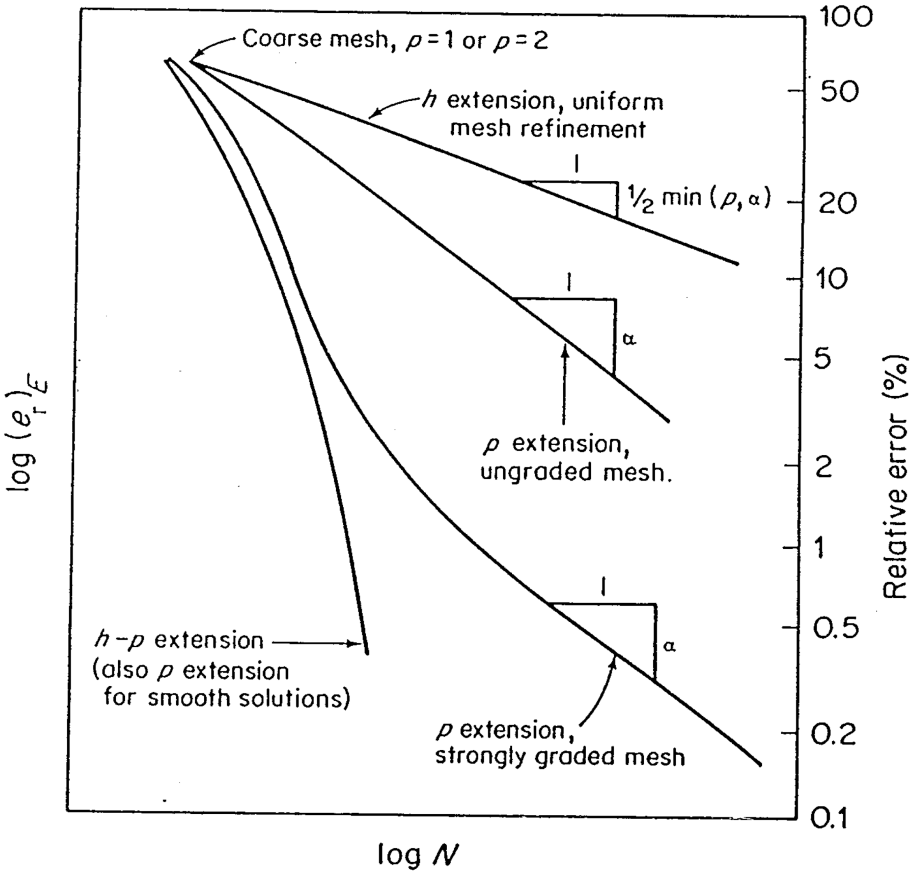
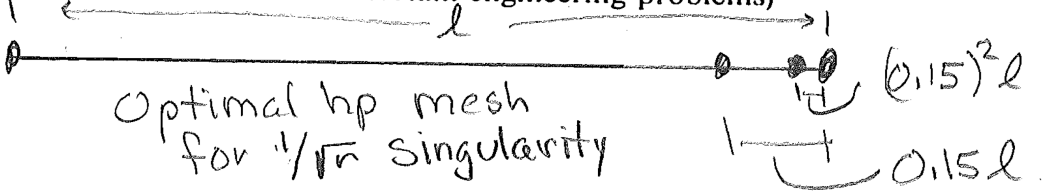


Figure 3.2 Performance of the *h*-, and *p*-, and *h*–*p*-extension processes. (The relative error values shown are typical for certain engineering problems)



We are often interested in the most accurate values of point wise quantities. In general some locations provide more accurate values than others. For example, we saw in 1D that the nodal values were exact and the derivative value was exact at some location in the element. What about 2D and 3D? The nodal values are not exact, but they tend to be more accurate with the values elsewhere typically not being dramatically less accurate.

However, the derivative values are more problematic. We care about this since there are “engineering” quantities of great interested like the values of stresses. As with the function itself, we do not know of any exact value existing. The more problematic part is that: (A) Although the stresses may be quite accurate at some specific points, they can be highly inaccurate at other points in the element. (B) Even if there are points which are more accurate, they are not the nodes, and worse than that, we do not know exactly where they are.

There are optimal sampling points for derivative values. There are even what are referred to as super-convergent points where they converge at a higher rate than they do elsewhere. Determination of super-convergence points is really hard even for simple problems on regular meshes. There has been work to try to generalize the question to find regions of “better convergence” done by Babuska and Strouboulis. However, even if one can follow what they talk about, the method is very complex to apply. On the more positive side, we do have some useful guidelines and some methods that work pretty well.

Lets start with the basic – Optimal 1D sampling points

The best approximation theory says the FE soln., u^h, is the best fit to the exact solution, u, WRT to the energy norm, $a(u^h,u^h)$. We can therefore think of u^h as a weighted (due to material parameters) least squares fit to the m^{th} derivatives for the simple cases of $a(u^h,u^h)$ we have seen in the text. The theory of least squares says that a polynomial approximation, $g^h(\xi)$, of order l that is a least squares fit to the function $g(\xi)$ will agree at the $l+1$ points corresponding to the zero's of the Legendre polynomial $P_l(\xi)$. Yes, these are the same locations we were finding for the Gauss quadrature points.

Recall our simplest case $a(u,u)=\int_0^1 u_x u_x\,dx$ we have what looks like a least squares case. With material parameters, it is now "weighted" $a(u,u)=\int_0^1 u_x \kappa u_x\,dx$.

Extrapolating this idea to the derivative terms in $a(u^h,u^h)$ we say u^h is a best least squared fit polynomial of order $k-m$ to the m^{th} derivative of the exact solution u. Thus the m^{th} derivation of u^h agrees with the m^{th} derivative of the exact solution u at the $k-m+1$ points corresponding to the zeros of the polynomial $P_{(k-m+1)}(\xi)$.

Note – this is only 1D but the 2D and 3D cases tend to have better values at these locations also.

Consider some elements:
4 noded, bi-linear quadrilateral – $k-m+1=1$ says the zero of $P_1(\vec{\xi})$ which is at $\vec{\xi}=0$. (Which is not the integration rule we typically want to use.)

8 noded quadratic quadrilateral – $k-m+1=2$ says the zero of $P_2(\vec{\xi})$ which correspond to the locations of the 2x2 Guass quadrature rule (which we like to use).

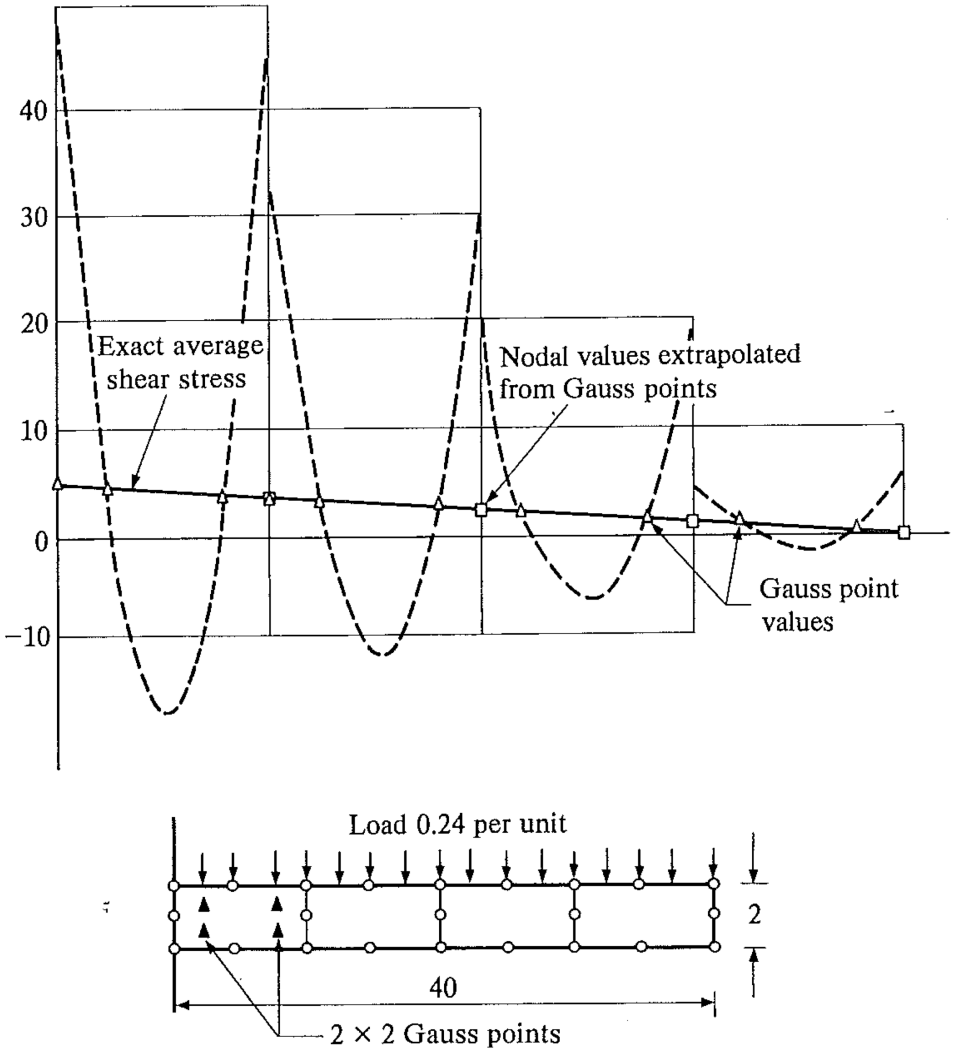


Fig. 12.14 A cantilever beam with four quadratic (Q8) elements. Stress sampling at cubic order (2 × 2) Gauss points with extrapolation to nodes

Interpretation of derivative values

Recall:

$$\vec{\xi} = Bd^e$$

$$\vec{\sigma} = DBd^e$$

Although these quantities are close to best integral least squares fits, we have just seen the direct evaluation of these expressions at any point in the element can give really poor values. We are looking for methods to define better (more accurate) distributions of these quantities over the elements in the mesh. Ideally we can base that on some pointwise evaluations of the above equations.

Consider some possibilities:

Direct evaluation at specific locations:

1. Node points: Would be the most convenient. First issue is each element sharing the node is a different value. More importantly, is that the node is one of the least accurate places to evaluate derivative values.
2. Other points on element boundaries: pretty much the same story as the nodal values.
3. Element Centroids: Often better, often at the location of one of the “optimal” (based on the simplified analysis) locations. Also typically within the portion of the element the Babuska and Strouboulis method would indicate is “good”. However, not always ideal. See the “cantilever” beam example above – the average shear stress is not even of the correct sign at the center of the elements.

4. Gauss integration points: What is most commonly used. As we saw the points often used for the reduced integration for elements are optimal in terms of what the simple analysis says.
5. Optimal stress evaluation points: Would be great if we knew them, but in reality we only have guidance on their locations.

Averaging the nodal and or other boundary values:
That is evaluate the values for each element that has the point of interest on its boundary. Then do a straight or weighted (i.e., based on element size) average. In some cases that may not be the bad. In others it is poor, look the beam example above. It clearly demonstrates that:

(bad value A + bad value B)/2 = bad value C

Finite difference relations:
Apply difference equations using the nodal values of the continuous FE field to define derivatives of that solution field. Not hard to apply to a uniform quad or hex mesh and may give reasonable values. However, we are typically using much less structured meshes, thus easy to use difference equations that are accurate are not available.

Projection Methods:
Build off the fact that the finite element is a best approximation in an integral sense of derivative quantities. One way to look at these methods is can we define another distribution that will end up being a more accurate point wise representation that we match to the finite element solution in an integral least squares sense.

Looking at the beam example what do we see as “issues” with the direct evaluation of the FE stresses:

- They are discontinuous between elements.
- They tend to be noise (higher order than desired) in the element.

We would like a method that addresses these things is some manner.

Three approaches:

1. Global projections onto a C^0 continuous derivative field This would likely be the most accurate method. However, it is quite expensive. As far as I know, no one uses it in widely distribute code.
2. Elemental projections using a lower order set of derivative shape functions. Will still be discontinuous between elements in general. From there one applies nodal averaging to make it C^0 continuous on the domain. Unlike straight nodal evaluation and averaging, the assumption is you are averaging much more accurate values. Has been used heavily in the past to be able to make the nice continuous color stress plots one gets from commercial codes.
3. Patchwise projections to define unique nodal values, the apply the original shape functions to define the C^0 derivative field. When a “lower order” fit is used on the patch level, this method is addressing, in a manner, both of the “issues” indicated above.

Global Projection:

$$\hat{\sigma} = \sum_{A=1}^{n_{np}} \hat{N}_A \sigma_A^* \quad \text{eq. 1}$$

$\hat{\sigma}$ is a C^0 derivative or stress field

$\hat{N}_A$ are C^0 nodal shape functions (assume interpolating)

σ_A^* are the nodal values of the C^0 derivative or stress field

σ^h is the finite element stress field (used below)

We need to select the nodal shape functions and define the criteria by which we determine the values of σ_A^*. Following the approach of Oden and Brauchi, we will use the same shape functions as used for the original field (since we are doing things globally this is not a problem in terms of being too high an order) and define an integral least squares fit.

Define

$$\mathrm{X}(\sigma_A^*) = \int_\Omega \left(\sigma^h - \hat{\sigma}\right)^2 d\Omega \quad \text{eq. 2}$$

as the function we want to minimize. Note that the only terms that can vary are the σ_A^* terms. Thus the minimum is found by setting.

$$\frac{\partial \mathrm{X}(\sigma_A^*)}{\partial \sigma_A^*} = 0 \text{ for } A = 1(1)n_{np}$$

Employing eq 2 with eq 1 substituted in we have:

$$0 = \frac{\partial}{\partial \sigma_A^*}\left[\int_\Omega \left(\sigma^h - \hat{N}_B \sigma_B^*\right)^2 d\Omega\right] \text{ for } A = 1(1)n_{np}$$

$$0 = 2\int_\Omega N_A \left(\sigma^h - \hat{N}_B \sigma_B^*\right) d\Omega \text{ for } A = 1(1)n_{np}$$

$$0 = \int_\Omega \hat{N}_A \sigma^h \, d\Omega - \int_\Omega \hat{N}_A \hat{N}_B \sigma_B^* \, d\Omega \text{ for } A = 1(1)n_{np} \text{ eq. 3}$$

taking a bit of liberty on notation the vector can be written as

$\sigma^h = [B]\{d\}$ for derivatives or $\sigma^h = [D][B]\{d\}$ for stresses where $[B]$ is the required derivative operators, $[D]$ is the material matrix and $\{d\}$ is the vector of nodal values of the solved for finite element field. Putting this in eq 3 we get

$$\int_\Omega \hat{N}_A \hat{N}_B \sigma_B^* \, d\Omega = \int_\Omega \hat{N}_A [D][B] d\Omega \{d\} \text{ for } A = 1(1)n_{np}$$

collecting for all A, and putting things in vectors and matrices we have

$$\int_\Omega [\hat{N}]^T [\hat{N}] d\Omega \{\sigma^*\} = \int_\Omega [\hat{N}]^T [D][B] d\Omega \{d\}$$

defining $[M_\sigma] = \int_\Omega [\hat{N}]^T [\hat{N}] d\Omega$ we have

$$\{\sigma^*\} = [M_\sigma]^{-1} \int_\Omega [\hat{N}]^T [D][B] d\Omega \{d\}$$

Note that the cost of solving this is on the order of the solution to the original finite element problems. Unlike the finite element stress field, σ^h, that correspond to a field in overall equilibrium, the stress field solved for here, $\hat{\sigma}$, is not in overall equilibrium. Thus an improvement that has been applied to this is to supplement it with “equilibrium iterations”. The same strategy of “equilibrium iterations” can be applied in the other two methods.

Elemental projections with lower order shape functions followed by averaging:

$$\hat{\sigma}^e = \sum_{a=1}^{n_{en}} \hat{N}_a^e \sigma_a^{e*} \text{ eq. 4}$$

$\hat{\sigma}^e$ is a element level derivative or stress field

$\hat{N}_a^e$ are element level shape functions, typically selected to be one order lower that finite element field shape functions (assume interpolating)

σ_a^{e*} are element level nodal values of the derivative or stress field

σ^h is the finite element stress field

The process will be same as before, but just at the element level. Define

$$\mathrm{X}(\sigma_a^*) = \int_{\Omega^e} \left(\sigma^h - \hat{\sigma}^e\right)^2 d\Omega^e \quad \text{eq. 5}$$

as the function we want to minimize. Note that the only terms that can vary are the σ_a^* terms. Thus the minimum is found by setting.

$$\frac{\partial \mathrm{X}(\sigma_a^*)}{\partial \sigma_a^*} = 0 \text{ for } a = 1(1)n_{en}$$

Employing eq 5 with eq 4 substituted in we have:

$$0 = \frac{\partial}{\partial \sigma_a^{e*}} \left[\int_{\Omega^e} \left(\sigma^h - \hat{N}_b^e \sigma_b^{e*}\right)^2 d\Omega^e \right] \text{ for } a = 1(1)n_{en}$$

$$0 = 2\int_{\Omega^e} \hat{N}_a \left(\sigma^h - \hat{N}_b \sigma_b^{e*}\right) d\Omega^e \text{ for } a = 1(1)n_{en}$$

$$0 = \int_{\Omega^e} \hat{N}_a \sigma^h \, d\Omega^e - \int_{\Omega^e} \hat{N}_a \hat{N}_b \sigma_b^{e*} \, d\Omega^e \text{ for } a = 1(1)n_{en} \quad \text{eq. 6}$$

taking a bit of liberty on notation the vector can be written as

$\sigma^h = [B]\{d^e\}$ for derivatives or $\sigma^h = [D][B]\{d^e\}$ for stresses

where $[B]$ is the required derivative operators, $[D]$ is the material matrix and $\{d^e\}$ is the vector of element nodal values of the solved for finite element field. Putting this in eq 6 we get

$$\int_{\Omega^e} \hat{N}_a^e \hat{N}_b^e \sigma_b^{e*} d\Omega^e = \int_{\Omega^e} \hat{N}_a^e [D][B] d\Omega^e \{d^e\} \text{ for } a = 1(1)n_{en}$$

collecting for all a, and putting things in vectors and matrices we have

$$\int_{\Omega^e} [\hat{N}]^T [\hat{N}] d\Omega^e \{\sigma^{e*}\} = \int_{\Omega^e} [\hat{N}]^T [D][B] d\Omega^e \{d^e\}$$

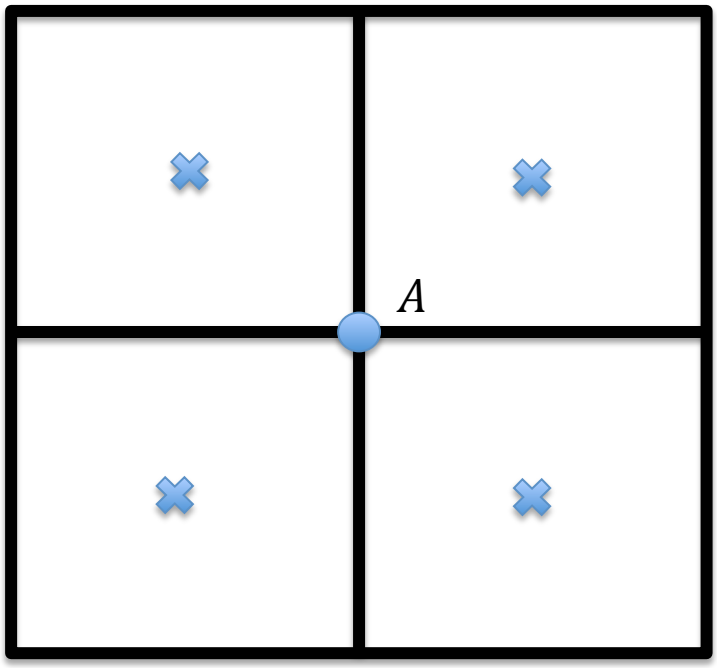
defining $[M_\sigma^e] = \int_{\Omega^e} [\hat{N}]^T [\hat{N}] d\Omega^e$ we have

$$\{\sigma^{e*}\} = [M_\sigma^e]^{-1} \int_{\Omega^e} [\hat{N}]^T [D][B] d\Omega^e \{d^e\}$$

Note that now the systems to solve are very small so the cost is a function of the number of elements and much less than solving the global system. Still have discontinuous values between elements, thus this step is followed by nodal averaging. Since the element values are more accurate (at least that is what they typically are) the averaged values can be accurate.

Patch wise least squares fit:

Developed by Zienkiewicz and Zhu (actually for use in *a posteriori* error estimation). Instead of defining the local problem over the element, the focus is on the individual node. Every node is surrounded by a patch of elements. A least squares fit to the values over the patch to some selected polynomial function defined over the patch is carried out and the nodal value obtained by evaluating this fitted function at the node point.



Patch of elements around node A

The polynomial selected to be fit to can be as simple as a tailor series

$$\hat{\sigma}^p = C_0 + C_1 x + C_2 y + C_3 x^2 + C_4 xy + C_5 y^2 + \ldots$$

In this case the problem is to minimize

$$\mathrm{X}(C_i) = \int_{\Omega^e} \left(\sigma^h - \hat{\sigma}^p \right)^2 d\Omega^e$$

and the process follows the same steps as we saw in the previous cases. Like the element level one, this one yields a one small system to solve per node to produce a unique nodal value. The stress field over the domain is then defined by using these nodal values with a set of C^0 nodal shape functions.

Note that since these methods employ numerical integration it is easy to see the statement of the local methods, particularly the patch wise one, as a discrete

least squares fit. This is useful in determining what order you may want for $\hat{\sigma}^p$. For example if we are to sample at one point per element in the above, then we most likely want to use a linear $\hat{\sigma}^p$ since the patch would only have four points of evaluation. If we used quadratic $\hat{\sigma}^p$ we do not have sufficient conditions for a discrete least squares fit.

Extraction Techniques

These methods employ appropriate Green's functions in the original FE integral forms to extract point wise quantities of interest that, in specific cases, are shown to converge at the rate of convergence of the energy norm. Some specific point wise stress extractions techniques have been defined. The problem is that there are no generalized Greens functions to extract everything desired. The methods are complex to implement and reasonably expensive. However, they do yield highly accurate results. Not likely these methods will ever be more than an academic exercise.