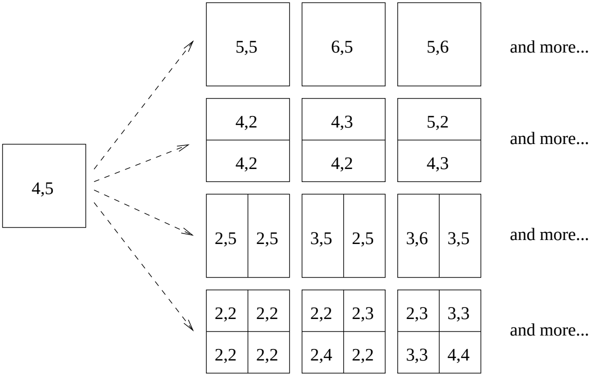
The Challenge

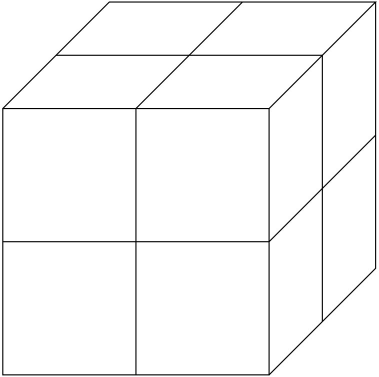
To refine a quad element:



$\approx$ 7000 candidates.

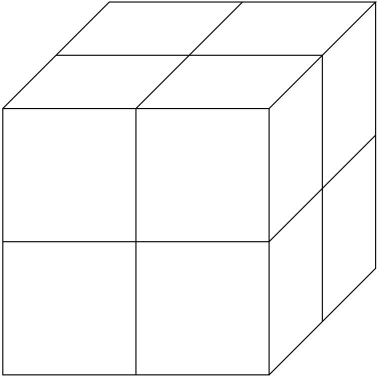
The Challenge

How many ways to refine a brick?



The Challenge

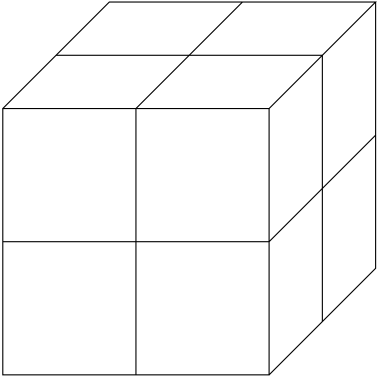
How many ways to refine a brick?



282 billion

The Challenge

How many ways to refine a brick?

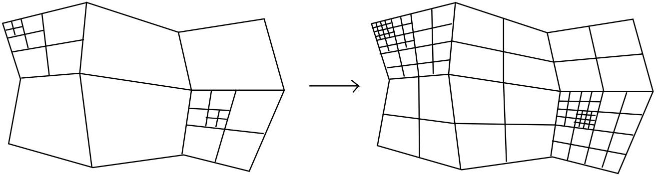


282 billion

$\Rightarrow$ estimating error magnitude in elements is not enough.

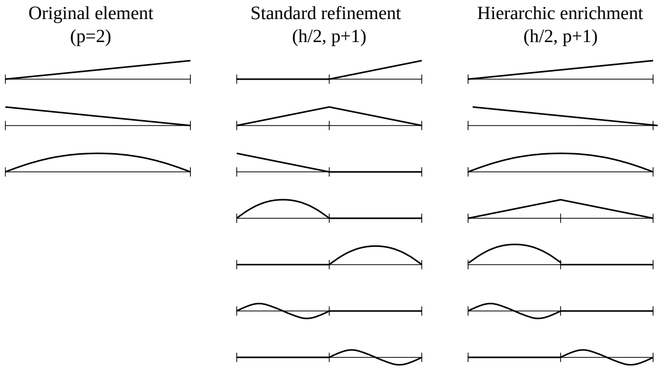
Estimating Shape of Error

- Use *approximation pairs* with different orders of accuracy in space
- Analogy to embedded ODE methods (Fehlberg, Hairer, Wanner et al.)



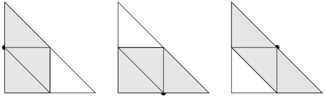
- Approximation pair maintained through the entire adaptivity process
- Global projection $\rightarrow$ global error
- Local projections $\rightarrow$ element refinements
- PDE-independent (as long as error is reasonably local)

Hierarchic *hp*-refinement: 1D case

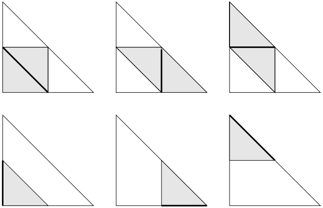


Quadrilateral elements: straightforward by product geometry

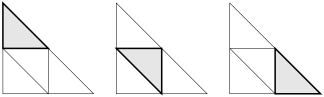
Hierarchic *hp*-refinement: triangles



Add three vertex functions.

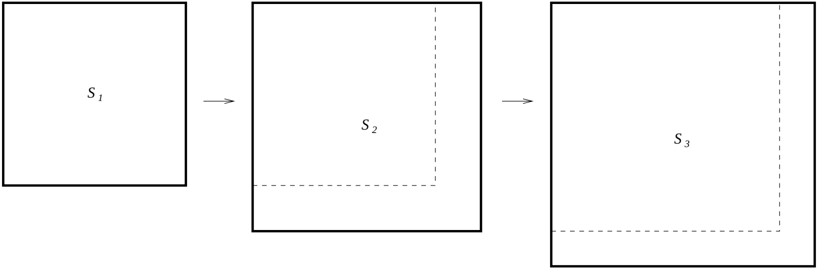


Add $p-1$ edge functions of degrees $2, 3, \ldots, p$ per highlighted edge.
Add one edge function of degree $p+1$ to each of the 9 edges.



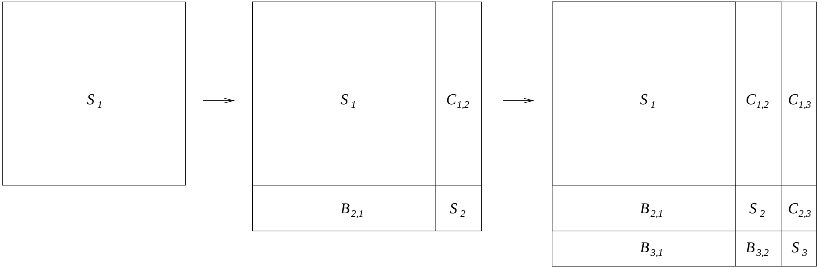
Add bubble functions of degrees $3, 4, \ldots, p$ into highlighted subelements.
Add $p-1$ bubble functions of degree $p+1$ to each of the four subelements.

Embedded stiffness matrices



Nonsymmetric, indefinite, ill-conditioned $\rightarrow$ need for a sparse direct solver

Block Jacobi method



Global discrete problem in enriched space:

$$\begin{pmatrix} S_1 & C_{1,2} \\ B_{2,1} & S_2 \end{pmatrix} \begin{pmatrix} Y_1 + \Delta Y_1 \\ Y_2 + \Delta Y_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

Solve $S_1 Y_1 = F_1, \ S_2 Y_2 = F_2.$

Block Jacobi method

Global discrete problem in enriched space:

$$\begin{pmatrix} S_1 & C_{1,2} \\ B_{2,1} & S_2 \end{pmatrix} \begin{pmatrix} Y_1 + \Delta Y_1 \\ Y_2 + \Delta Y_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

Iterative method for ΔY_1 and ΔY_2:

$$L_1 U_1 \Delta Y_1^{(k+1)} = -C_{1,2} \left(Y_2 + \Delta Y_2^{(k)} \right),$$
$$L_2 U_2 \Delta Y_2^{(k+1)} = -B_{2,1} \left(Y_1 + \Delta Y_1^{(k)} \right).$$

Works extremely well for a wide range of problem types.

How is your *hp*-adaptivity?

Benchmarks are available:

- W. Mitchell: A Collection of 2D Elliptic Problems for Testing Adaptive Algorithms, NISTIR 7668, 2010.
- P. Solin, O. Certik, L. Korous: Three Anisotropic Benchmarks for Adaptive Finite Element Methods, Appl. Math. Comput., doi:10.1016/j.amc.2010.12.080.

Response to Mitchell's paper:

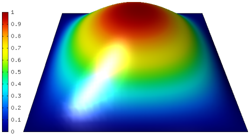
- Z. Ma, L. Korous, E. Santiago: Solving a Suite of NIST Benchmark Problems for Adaptive FEM with the Hermes Library, Journal CAM.

Example: Smooth-Iso (Elliptic)

Equation: $-\Delta u = f \quad \text{in } \Omega = (0, \pi)^2$.

Solution: $u(x, y) = \sin(x)\sin(y)$.

Benchmark Smooth-Iso:



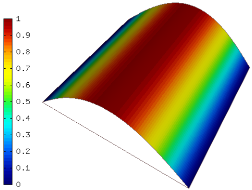
Start with a single bilinear element.
Goal: Rel. error in H^1-norm less than 10^{-4} % with at most 49 DOF.

Example: Smooth-Aniso (Elliptic)

Equation: $-\Delta u = f \quad \text{in } \Omega = (0, \pi)^2$.

Solution: $u(x, y) = \sin(x)$.

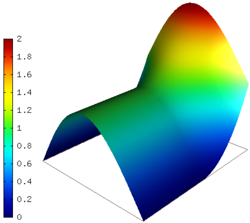
Benchmark No. 1 from Solin-Certik-Korous:



Start with a single bilinear element.
Goal: Rel. error in H^1-norm less than 10^{-4} % with 16 DOF.

Example: Line Singularity (Elliptic)

Benchmark No. 10 from W. Mitchell's NIST Report NISTIR 7668:



Start with a single bilinear element.
Goal: Rel. error in H^1-norm less than 10^{-4} % with at most 150 DOF.