Order verification - facts and fiction

Pavel Bochev

Computational Mathematics and Algorithms
Sandia National Laboratories

Sandia CSRI Workshop on Math Methods for V&V
August 14-16, 2007

Supported in part by

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

Definitions

Fact: a thing that is indisputably the case: Theory $\Rightarrow$ Practice

Theory: under some assumptions (solution regularity, grid regularity) FE solutions should converge as $O(h^r)$ for some $r>0$

Practice: if these assumptions are satisfied, numerical solutions converge as $O(h^r)$ too.

Fiction: a belief or statement that is false but is often held to be true because it is expedient to do so: Practice $\Rightarrow$ Theory

Checking that numerical solutions converge as $O(h^r)$ for **smooth solutions** on a limited set of grids (**usually uniform**) is sufficient to declare that the code

- is implemented correctly
- is robust and will perform well on general grids
- will recover physical solutions that are less regular than required by convergence theory, i.e., the only side effect will be slower convergence, not reduced fidelity

I will illustrate how things can go wrong using 3 codes for the Poisson (!) equation implemented using FE on quadrilateral grids.

$$-\Delta\phi = 0 \ \text{ in } \ \Omega$$
$$\phi = 0 \ \text{ on } \Gamma$$

– Standard Galerkin method with Q1 elements

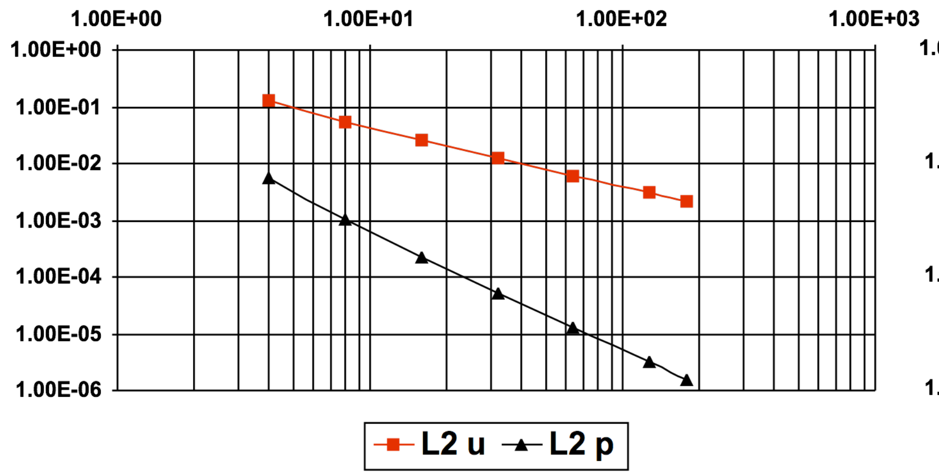
– Mixed Galerkin method with Q1-P0 elements

– Mixed least-squares method with Q1-Q1 and RT0-Q1 elements

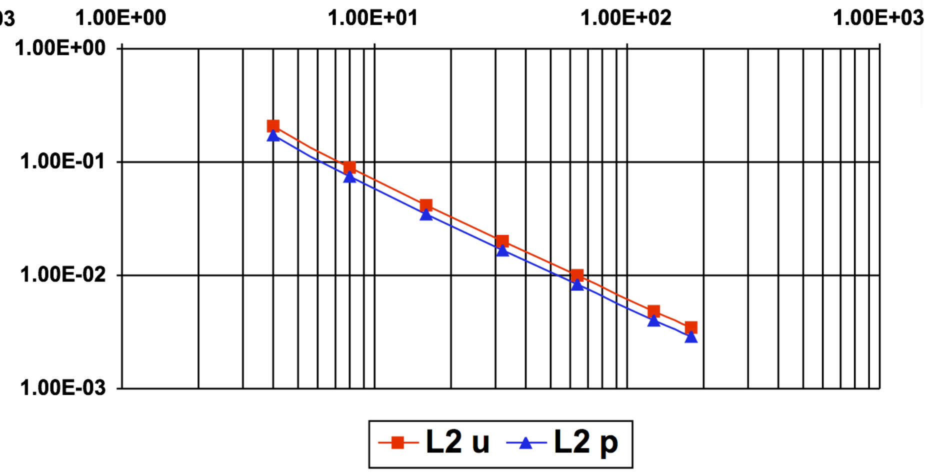
Order verification

This is what we see on a sequence of uniform grids:

Galerkin



Mixed Galerkin



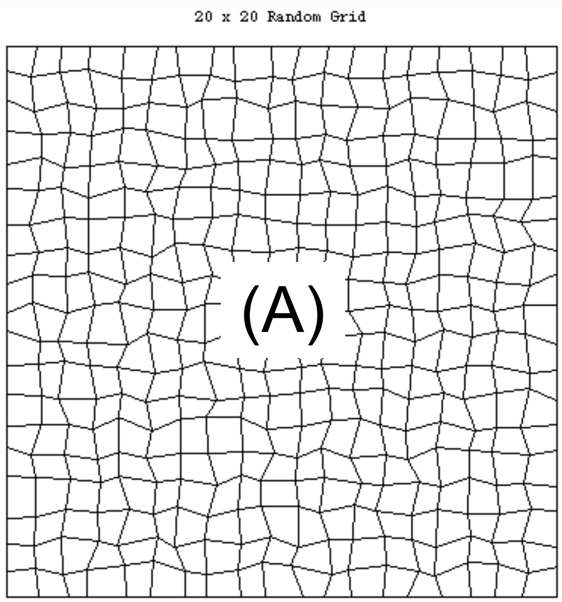
And it is exactly what the theory predicts:

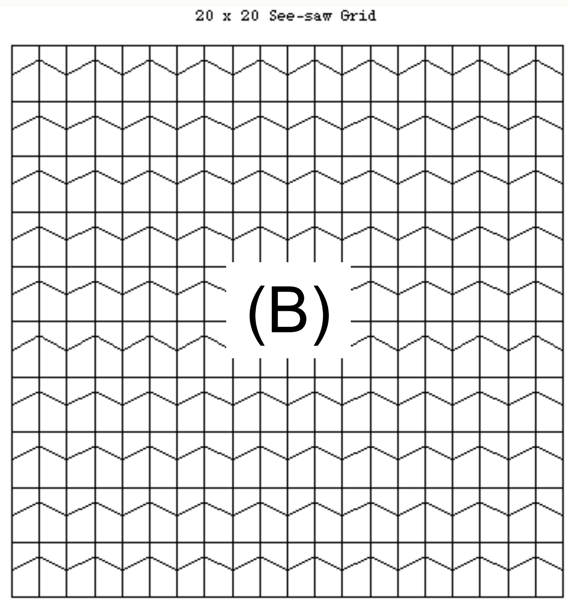
$\left\| p - p^h \right\| = O(h^2)$ and $\left\| u - u^h \right\| = O(h)$

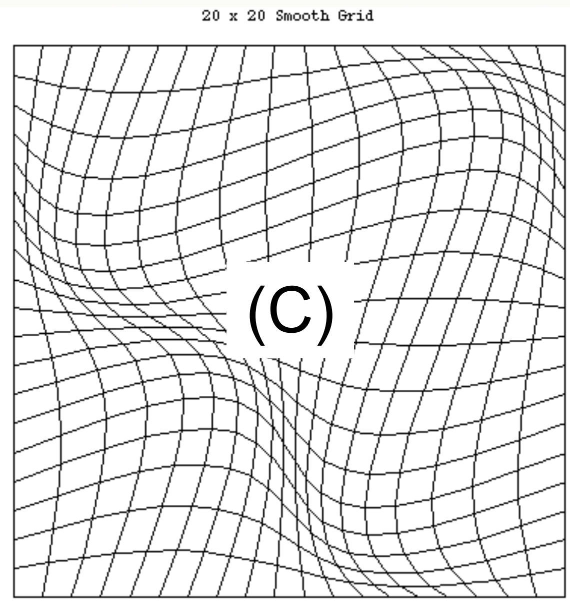
$\left\| p - p^h \right\| = O(h)$ and $\left\| u - u^h \right\| = O(h)$

So the theory seems to check out just fine…

Let’s try some other grids





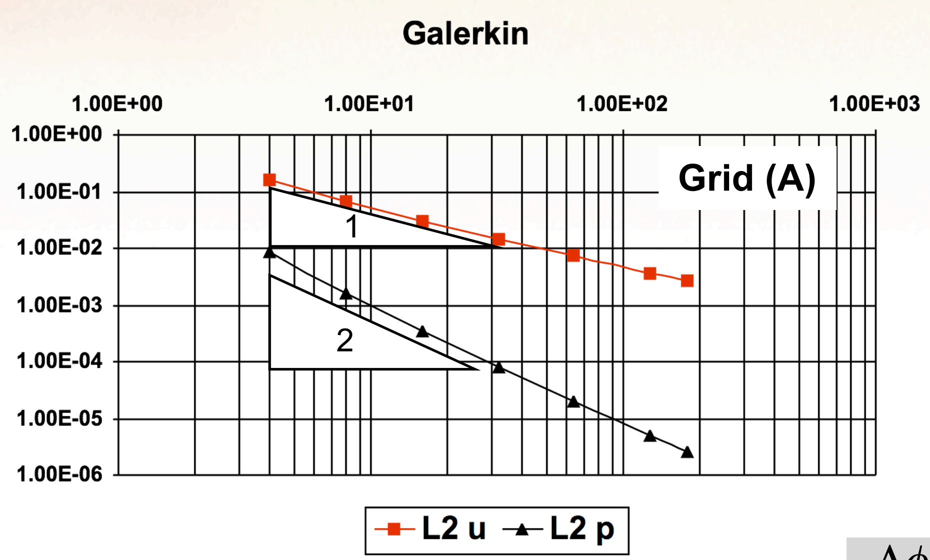


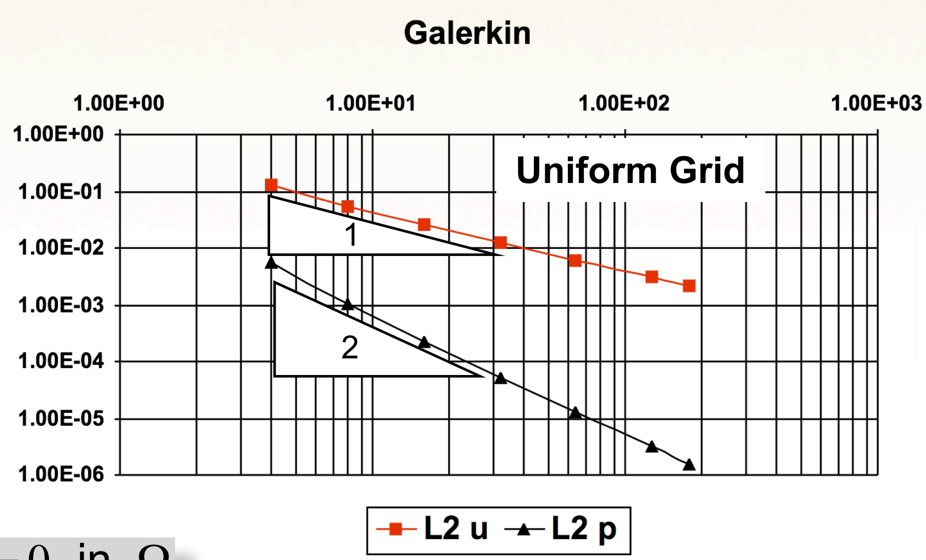
(A) & (B) are **mild perturbations** of a uniform mesh

(C) has the **worst** cell aspect ratio.

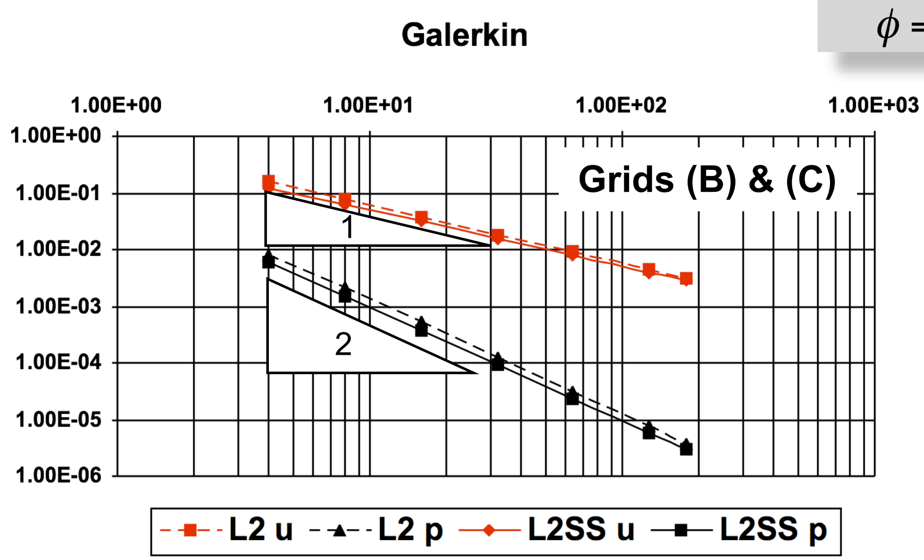
All grid sequences are uniformly regular

The Galerkin code seems to work fine





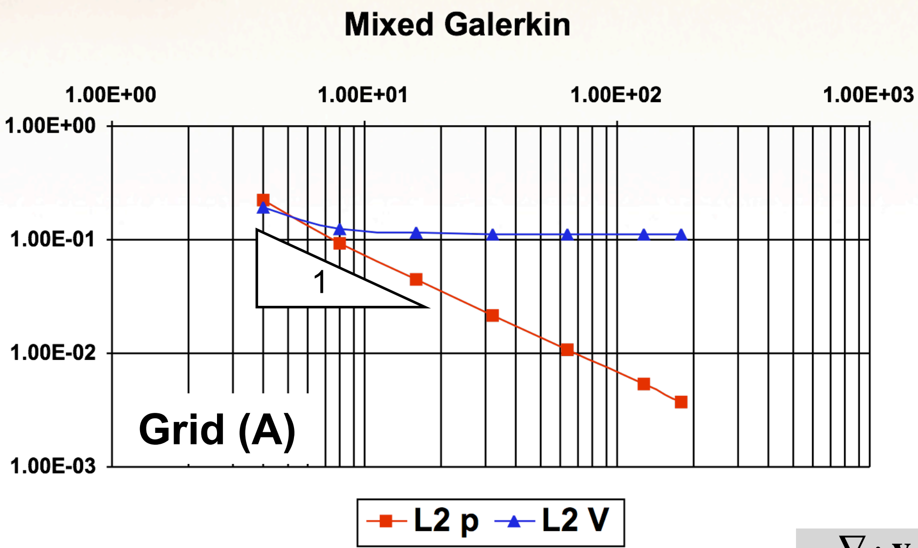
$$-\Delta\phi = 0 \ \text{ in } \ \Omega$$
$$\phi = 0 \ \text{ on } \Gamma$$

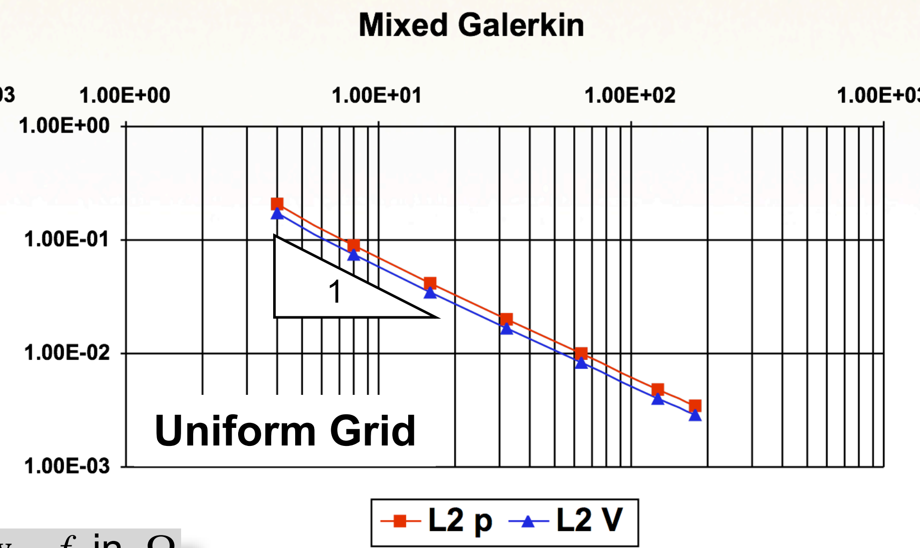


Nodal elements have comparable performance and are optimally accurate on all 3 grids

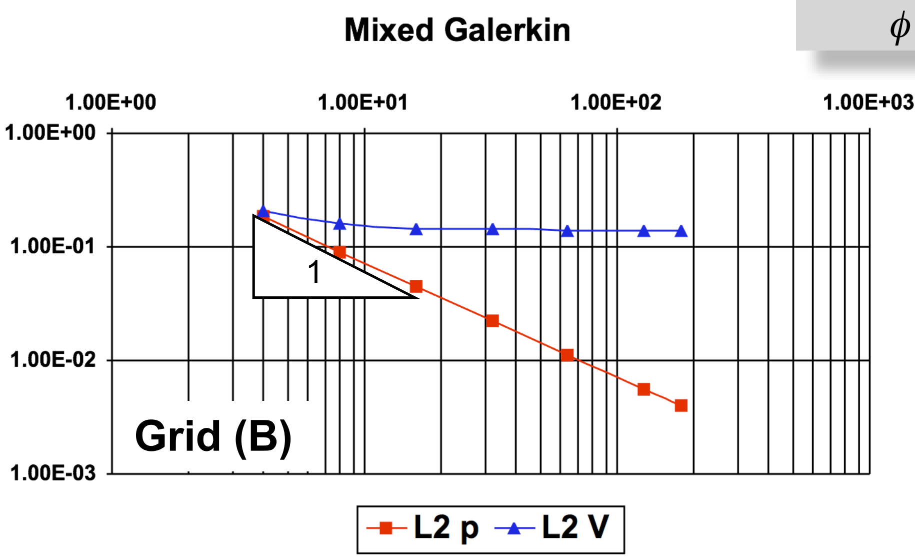
2x2 Gauss points used in all cases

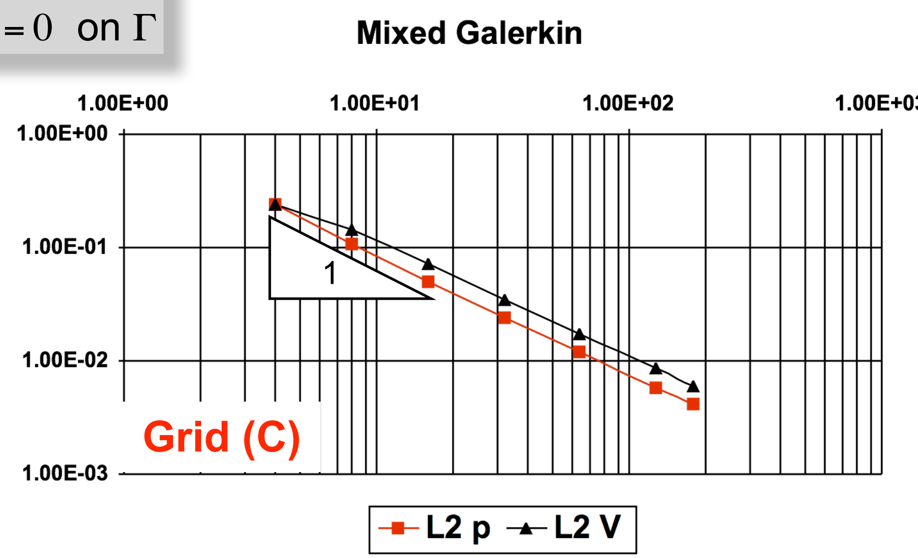
But look what happens to the Mixed Galerkin!





$$
\begin{aligned}
\nabla \cdot \mathbf{v} &= f \ \text{ in } \ \Omega \\
\mathbf{v} + \nabla \phi &= 0 \ \text{ in } \ \Omega \\
\phi &= 0 \ \text{ on } \Gamma
\end{aligned}
$$





What is going on here?

We were too quick to conclude that all is fine by checking the error bound

$$\left\| p - p^h \right\| = O(h) \text{ and } \left\| u - u^h \right\| = O(h)$$

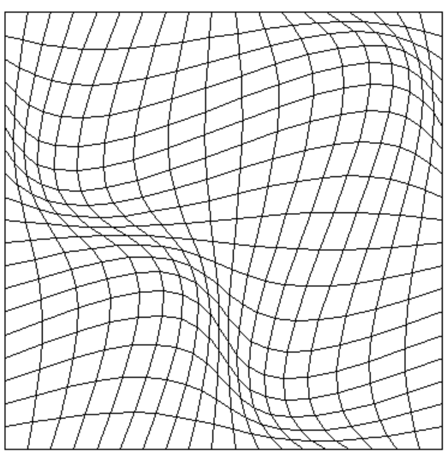
only on uniform grids. It turns out that this bound **does not hold on non-affine grids**!

Why? The RT0 space does not contain constants ➡ $W(\mathbf{x}) = \underbrace{\frac{1}{\det DF} DF}_{\text{Piola Transform}} \hat{W} \circ F^{-1}(\mathbf{x})$

This explains our results:

In the eyeball norm, grid **(C)** appears worse than **(A)** and **(B)**! However, the cells of **(C)** are, on the average closer to **affine** quads, than the cells of **(A)** and **(B)**!

20 x 20 Smooth Grid



- ✓ Face elements can behave **differently** from nodal elements
- ✓ Convergence studies must be carefully **designed**
- ✓ Using only uniform grids **can be inconclusive**!
- ✓ Tests must include **strongly** and **weakly** non-affine grids
- ✓ Choice of quadrature can change convergence too!

Now let's look at a different scenario

We take 2 different implementations of a mixed least-squares method:

- Mixed least-squares method with Q1-Q1 and RT0-Q1 elements

This is what we see on a sequence of uniform grids for a smooth solution:

LS vs BA	scalar		vector	
	H^1	***H^1***	***H(div)***	***H(div)***
	Q1-RT0	**Q1-Q1**	**Q1-RT0**	**Q1-Q1**
LS	**1.01**	**1.00**	**1.00**	**0.99**
BA	**1.00**	**1.00**	**1.00**	**1.00**

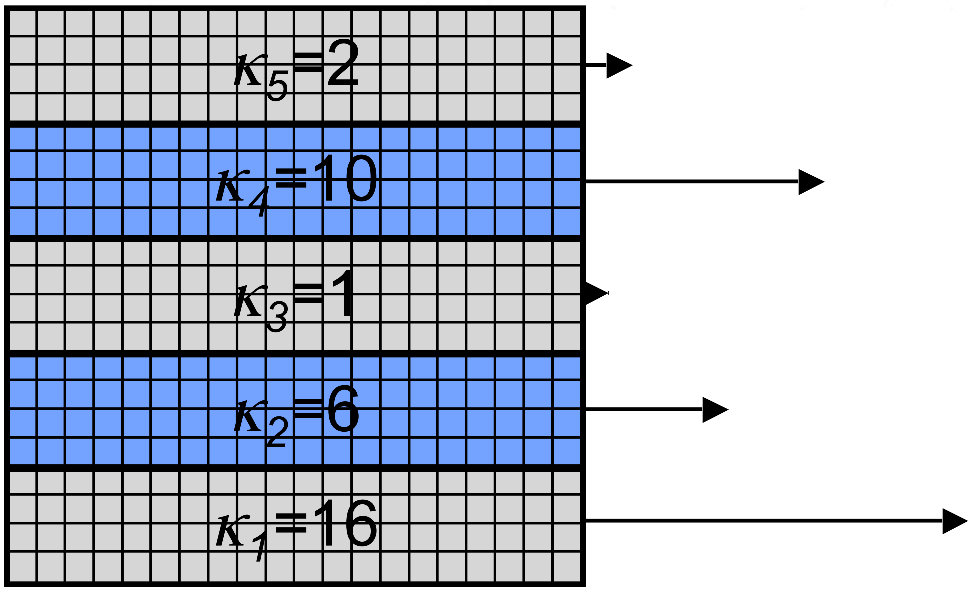
And it is exactly what the theory predicts for this method:

$$\left\| p - p^h \right\|_1 + \left\| \nabla \bullet (u - u^h) \right\| = O(h)$$

So the theory seems to check out just fine for both methods…

Let's try another (rough) solution

From: *T. Hughes, A. Masud and J. Wan, A stabilized mixed DG method for Darcy flow*



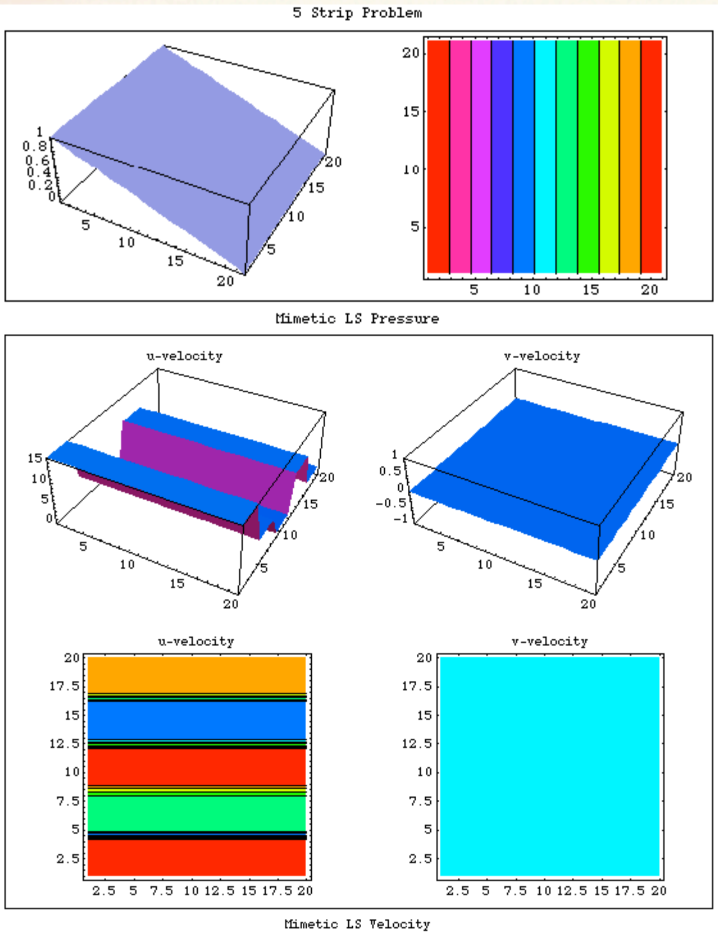
Exact solution

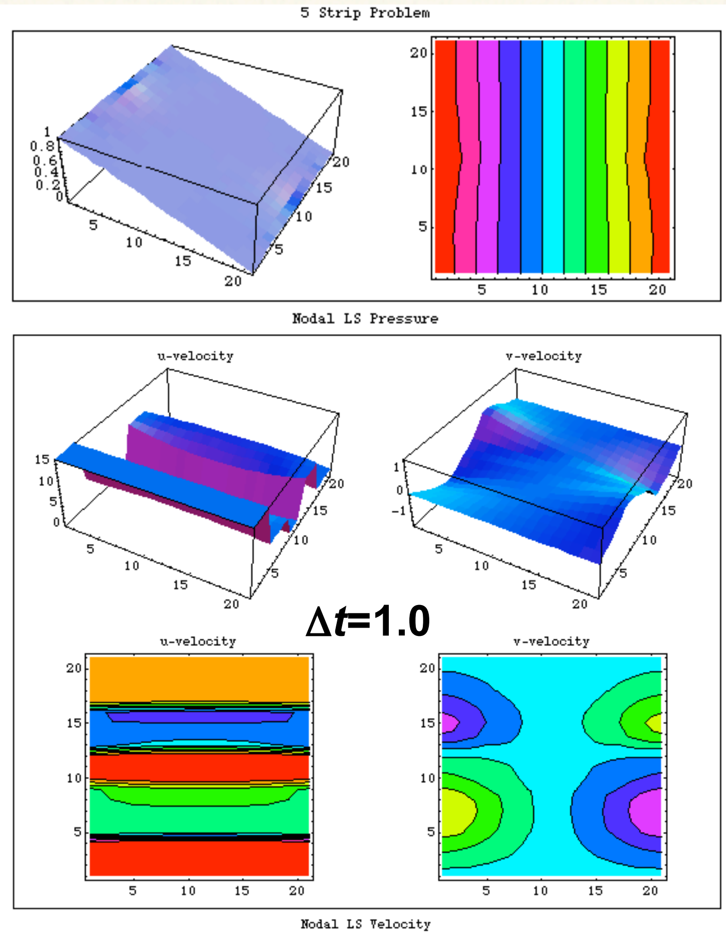
$$\phi = 1 - x;$$

$$\mathbf{u} = \begin{pmatrix} \kappa_i \\ 0 \end{pmatrix} \text{ in strip } i$$

- Problem is driven by Neumann boundary condition (normal flux)
- Solution is physical, i.e., can be realized in an experiment
- Grid has 400 uniform elements aligned with the interfaces between the strips

Q1-RT0 **???** Q1-Q1



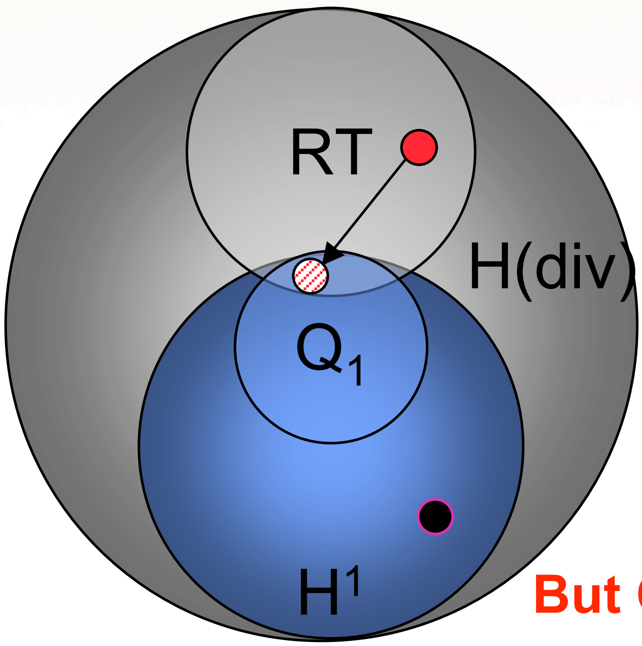


Recovers the physical solution! **Recovers a spurious solution!**

The trouble is, the spurious solution looks completely reasonable. Order verification did not signal any trouble for the Q1-Q1 code. Imagine this scenario in a complex multiphysics code…

This is what happens in this test:

Solution belongs to Q1-RT0 $\Rightarrow$ **recovered by Q1-RT0 implementation**



Least-Squares solution is a projection onto the discrete space

$\Downarrow$

gives the best possible approximation out of that space with respect to the energy norm

Q1-Q1 Least-Squares: gives the **best energy norm** approximation of that solution out of Q1

But $Q_1 \subset H^1$ and H^1 has infinite co-dimension in H(div)!

$\Rightarrow$ Error > const. $\Rightarrow$ Q1-Q1 LS will never converge!

- ✓ Convergence for smooth solutions **does not guarantee approximation** of other solutions!
- ✓ Testing on rough but physically meaningful solutions **should be part of the verification**
- ✓ This is **accepted practice in hydro codes** which are expected to work for rough solutions
- ✓ This is **seldom practiced** in codes for elliptic problems despite the fact that in many cases we can only expect, e.g., H(curl) or H(div) solution (discontinuous materials, etc)