Bringing Serendipity Methods to Computational Practice in Firedrake

CYRUS CHENG, Imperial College, United Kingdom
JUSTIN CRUM, University of Arizona
ANDREW GILLETTE, University of Arizona
DAVID HAM, Imperial College, United Kingdom
ROBERT KIRBY, Baylor University
JOSHUA A. LEVINE, University of Arizona
LAWRENCE MITCHELL, Durham University, United Kingdom

An abstract about Firedrake and FEM here.

ACM Reference Format:
Cyrus Cheng, Justin Crum, Andrew Gillette, David Ham, Robert Kirby, Joshua A. Levine, and Lawrence Mitchell. 2018. Bringing Serendipity Methods to Computational Practice in Firedrake. 1, 1 (December 2018), **??** pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

2 BACKGROUND ON SERENDIPITY AND TRIMMED SERENDIPITY ELEMENTS

2.1 2D Elements

(1) Scalar (classical = Arnold-Awanou = $\mathcal{S}_r\Lambda^0(\mathbb{R}^2)$)
(2) Vector Serendipity (BDM = Arnold-Awanou = $\mathcal{S}_r\Lambda^1(\mathbb{R}^2)$)
(3) Vector Trimmed Serendipity (Arbogast-Correa = Gillette-Kloefkorn = $\mathcal{S}_r^-\Lambda^1(\mathbb{R}^2)$)
(4) Direct (Arbogast-Tao / Arbogast-Correa)

2.1.1 *Scalar (classical = Arnold-Awanou = $\mathcal{S}_r\Lambda^0(\mathbb{R}^2)$).*

2.1.2 *Vector Serendipity (BDM = Arnold-Awanou = $\mathcal{S}_r\Lambda^1(\mathbb{R}^2)$).*

2.1.3 *Vector Trimmed Serendipity (Arbogast-Correa = Gillette-Kloefkorn = $\mathcal{S}_r^-\Lambda^1(\mathbb{R}^2)$).*

2.1.4 *Direct (Arbogast-Tao / Arbogast-Correa).*

Authors' addresses: Cyrus Cheng, cyrus.cheng15@imperial.ac.uk, Imperial College, London, United Kingdom; Justin Crum, jcrum@math.arizona.edu, University of Arizona, Tucson, Arizona; Andrew Gillette, University of Arizona, Tucson, Arizona, agillette@math.arizona.edu; David Ham, Imperial College, London, United Kingdom, david.ham@imperial.ac.uk; Robert Kirby, Baylor University, Waco, Texas, robert_kirby@baylor.edu; Joshua A. Levine, University of Arizona, josh@email.arizona.edu; Lawrence Mitchell, Durham University, Durham, United Kingdom, wence@gmx.li.

Manuscript submitted to ACM

2.2 3D Elements

(1) Scalar (classical = Arnold-Awanou = $\mathcal{S}_r\Lambda^0(\mathbb{R}^3)$)
(2) Vector serendipity (Arnold-Awanou = $\mathcal{S}_r\Lambda^1(\mathbb{R}^3)$ and $\mathcal{S}_r\Lambda^2(\mathbb{R}^3)$)
(3) Vector trimmed serendipity (Gillette-Kloefkorn = $\mathcal{S}_r^-\Lambda^1(\mathbb{R}^3)$ and $\mathcal{S}_r^-\Lambda^2(\mathbb{R}^3)$)

3 BUILDING CAPACITY FOR SERENDIPITY ELEMENT TYPES IN FIREDRAKE

Description of which elements are now available in Firedrake and how to call them.

4 TIMING EXPERIMENTS

Compare tensor product to serendipity and direct serendipity as applicable

(1) Wall clock time
(2) Static condensation time
(3) matvec time
(4) KSP / solver time

4.1 Timing: 2D Elements

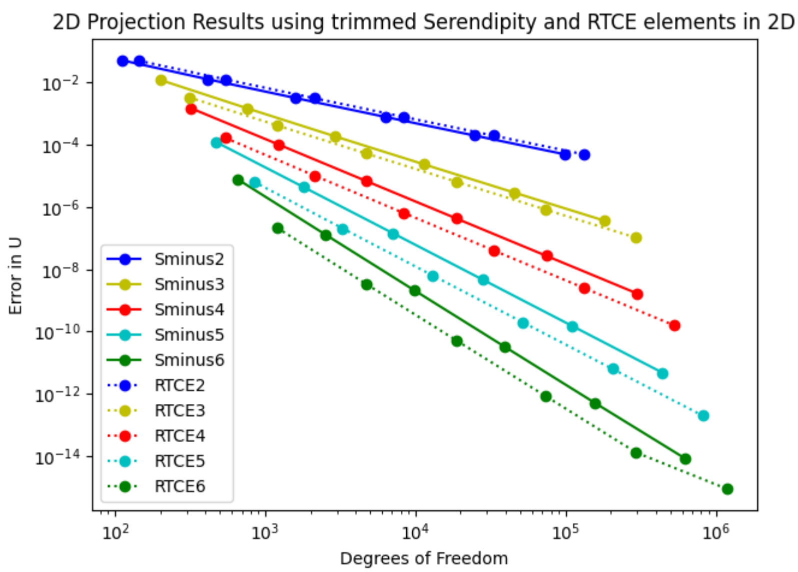
4.2 Timing: 3D Elements

5 DEGREES OF FREEDOM AND MEMORY EXPERIMENTS

5.1 DOFs and Memory: 2D Elements

The 2D elements are tested using a few test problems. The first is a basic projection check to make sure that the elements are behaving with the expected convergence rate. Then for checking 0-forms, we use the Primal Poisson problem, while for 1-forms, we use the mixed Poisson problem. The results of these experiments are shown below in the various figures.

Fig. 1. Degrees of Freedom vs Error analysis of projection using S^{-}(Curl) and RTCE.



First, we graphed the errors from computing projections using trimmed Serendipity and tensor product elements in 2D, shown in **??**. This confirms that for a basic projection, we are getting the convergence rates for the trimmed Serendipity elements that we expect. Next we wanted to work on the primal Poisson and mixed Poisson problems.
We solve the primal Poisson problem described below on a unit square domain Ω

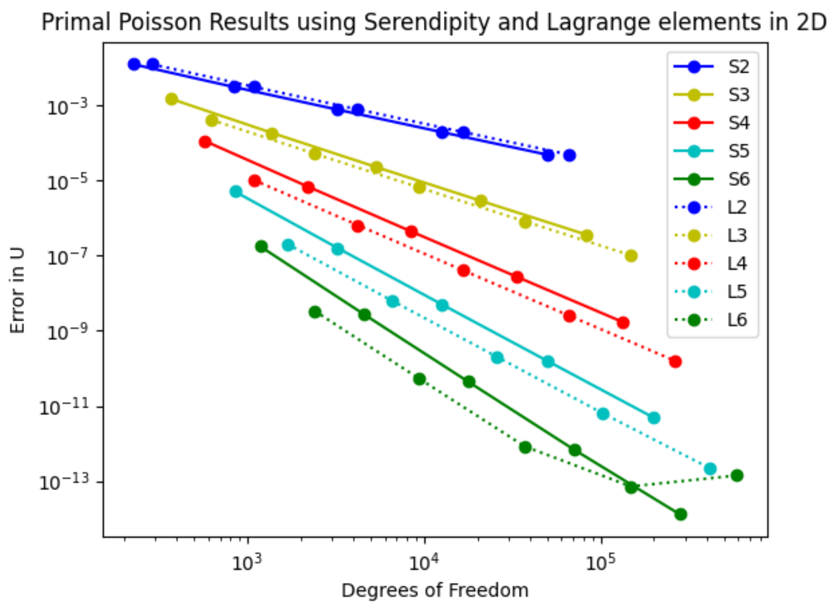
$$
\begin{aligned}
-\nabla^2 u &= 2\pi^2 \sin(\pi x)\sin(\pi y) \in \Omega \qquad (1)\\
u|_{\partial\Omega} &= 0\\
\nabla u \cdot n &= 0 \text{ on } \partial\Omega
\end{aligned}
$$

which yields the solution $u(x, y) = \sin(\pi x)\sin(\pi y)$. The corresponding mixed Poisson problem that we solve is then

$$
\begin{aligned}
\sigma - \nabla u &= 0 \qquad (2)\\
\nabla \cdot \sigma &= -f\\
u|_{\partial\Omega} &= 0
\end{aligned}
$$

which has the same solution as the primal problem above. The results for the primal problem can be seen in **??**, and the results for the mixed problem are shown in **??**. Error for the primal problem is measured in the energy norm, while the mixed problem error is computed using the L^2 norm on u.

Fig. 2. Degrees of Freedom vs Energy Error analysis of Serendipity and Lagrange L^2 elements.



In **??**, we see a comparison of 1-forms for solving the mixed Poisson problem in 2D.

5.2 DOFs and Memory: 3D Elements

In the 3D case, we extend the primal and mixed problems to have $u = \sin(\pi x)\sin(\pi y)\sin(\pi z)$ as the solution. The results found in **??** refer to solving the mixed Poisson problem using tensor product and trimmed Serendipity H(div) elements in 3D.

We expect that the rates of convergence should track closely at each order. Trimmed Serendipity elements in general use less degrees of freedom, especially on the interior of the cubes forming the mesh. The results that we get are consistent with that fact, illustrating that trimmed Serendipity elements do in fact require less computation time. Any data points that are illustrated required less than 3TB of memory and less than 48 hours from an HPC job starting to ending. Data points that are missing would have required more than 48 hours to run.

Fig. 3. Degrees of Freedom vs Error analysis of S^{-}(Div) and RTCE elements for a mixed Poisson PDE, with exact solution $sin(\pi x)sin(\pi y)$!

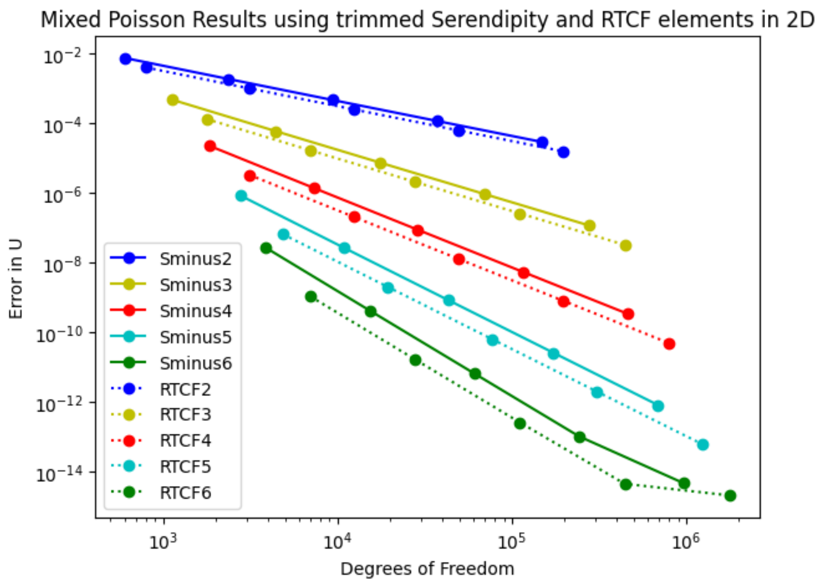


Fig. 4. Preliminary results for testing a mixed Poisson problem in 3D, using S^{-}(Div) and NCF elements.

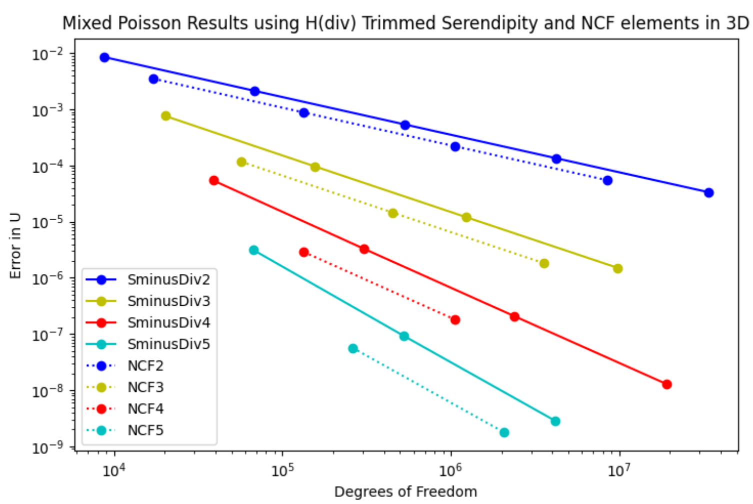


Fig. 5. Analyzing convergence rates of trimmed Serendipity vs tensor product elements in 3D using H(div) elements.

6 ACCURACY AND CONVERGENCE RATE EXPERIMENTS

In this section, we will explore the affect that mesh size has on the error and convergence rates of the trimmed Serendipity and tensor product elements. The test problems, as before, are projection, primal Poisson, and mixed Poisson, allowing us to illustrate the effectiveness of scalar elements as well as H(div) and H(curl) elements.

6.1 Error: 2D Elements

As in the previous sections, we give results for the same test problems (projection, primal Poisson, mixed Poisson) to give a uniform idea of the benefits and costs to using trimmed Serendipity.

Fig. 6. Analysis of projection using S^-(Curl) and RTCE.

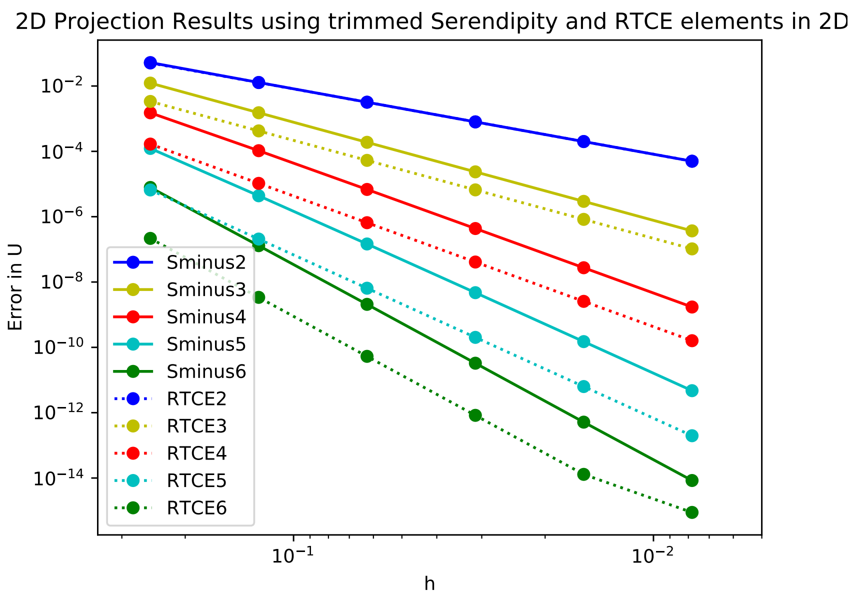
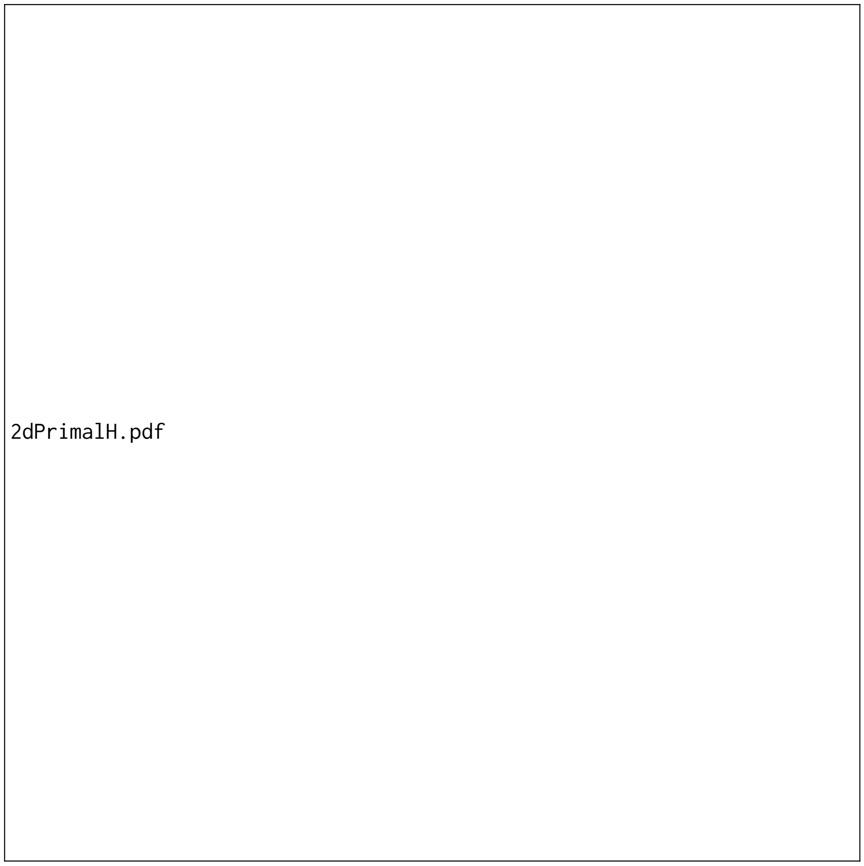


Fig. 7. Primal Poisson analysis of h vs error.



6.2 Error: 3D Elements

We tested out the H(curl) elements in 3D on the cavity resonator problem, where the Maxwell equations give us the eigenvalue problem **AG:** (1) The wording is a little ambiguous - are you about the state the resonator problem or Maxwell's equations? Or are these the same? (2) I prefer to see the statement of the variables prior to the PDE i.e. move the part after the equation to before it. (3) You need to state the domain - I think it's $[0,1]^3 \subset \mathbb{R}^3$ - and the boundary conditions (periodic?) (4) Give the eigenvalue equation a label so you can reference it later.

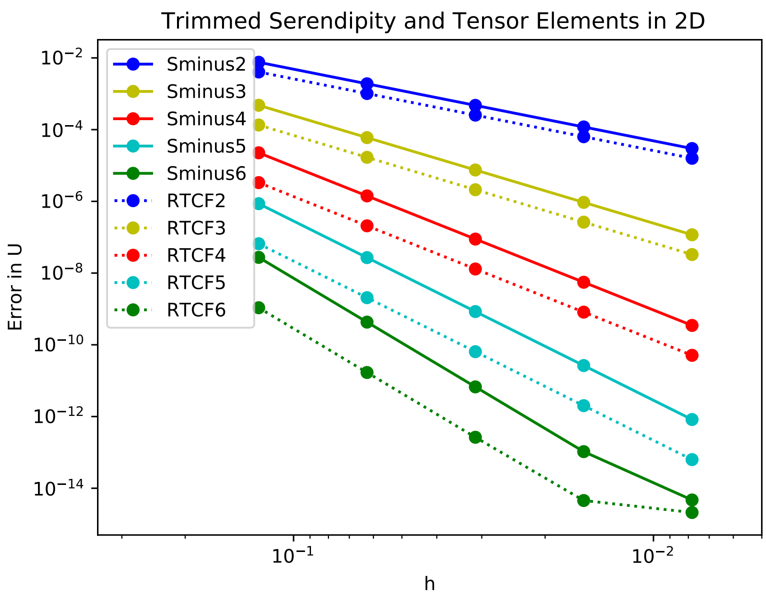
$$\langle \mathrm{curl}(F), \mathrm{curl}(E) \rangle = \omega^2 \langle F, E \rangle,$$

where we want to find resonances ω and eigenfunctions E such that $\omega \in \mathbb{R}$ and $E \in$ H(curl). The exact eigenvalues should follow the formula

$$\omega^2 = m_1^2 + m_2^2 + m_3^2$$

where $m_i \in \mathbb{N} \cup 0$ and no more than one of m_1, m_2, m_3 may be equal to 0 at a time. **AG:** Cite a reference for this - probably Boffi?

Fig. 8. Degrees of Freedom vs h analysis of S^-(Div) and RTCE elements for a mixed Poisson PDE, with exact solution $sin(\pi x)sin(\pi y)$
!



AG: Put discretized version of eigenvalue problem here (see comment below)

AG: (1) For this second paragraph, the first (topic) sentence is good - you are going to discuss the table. The next few sentences need to be *what* the table is. What are the columns? What are the rows? What is the top vs the bottom of the table? What does each entry in the table represent? What is in the parentheses? *Then* give the formula for convergence rate, and explain what the !-4 notation means, and the specifics about SLEPc. Imagine you are teaching this to a first year graduate student - you have to explain everything. That should be the entire content of this paragraph. Then have a new paragraph with topic sentence something like "Table **??** highlights a number of revealing differences between computing with NCE and serendipity elements." Then launch into what those differences are. (2) FYI: I added 'Table' before the ref command. Also I had to move the label command before the end table command as it was picking up the section number instead of the table number. (3) You're not *really* solving the eigenvalue problem - you're solving a discretization of it. I think we need to state the discretized version formally. We can discuss this when we talk if you're not sure how / where to find the formulation.

In Table **??**, we look at the convergence rates of different eigenvalues based off solving the problem with tensor product (NCE) elements and trimmed Serendipity (S^-) elements in 3D. Note that the convergence rates are computed by doing

$$r = \frac{\log\left(\frac{\tilde{\lambda}_{i,N}-\lambda_{i,N}}{\tilde{\lambda}_{i,N+1}-\lambda_{i,N+1}}\right)}{\log\left(\frac{h_N}{h_{N+1}}\right)}$$

NCE Elements				
Actual	N = 4	N = 8	N = 16	N = 32
2	2.0010243	2.0000655 (3.97)	2.0000041 (3.99)	2.0000003 (4.00)
2	2.0010243	2.0000655 (3.97)	2.0000041 (3.99)	2.0000003 (4.00)
2	2.0010243	2.0000655 (3.97)	2.0000041 (3.99)	2.0000003 (4.00)
3	3.0015364	3.0000983 (3.97)	3.0000062 (3.99)	3.0000004 (4.00)
3	3.0015364	3.0000983 (3.97)	3.0000062 (3.99)	3.0000004 (4.00)
! $\rightarrow$ 4	4.0300893	4.0020486 (3.88)	4.0001311 (3.97)	4.0000082 (3.99)
! $\rightarrow$ 4	4.0300893	4.0020486 (3.88)	4.0001311 (3.97)	4.0000082 (3.99)
5	5.0306014	5.0020813 (3.88)	5.0001331 (3.97)	5.0000084 (3.99)
5	5.0306014	5.0020813 (3.88)	5.0001331 (3.97)	5.0000084 (3.99)
5	5.0306014	5.0020813 (3.88)	5.0001331 (3.97)	5.0000084 (3.99)
5	-	-	5.0001331	5.0000084 (3.99)
6	-	6.0021141	6.0001352 (3.97)	-
6	-	6.0021141	6.0001352 (3.97)	-
DOF	1944	13872	104544	811200
EPS Solve Time (seconds)	0.0514	0.2811	3.2795	29.2620
S^- H(curl) Elements				
Actual	N = 4	N = 8	N = 16	N = 32
2	2.0010919	2.0000664 (4.04)	2.0000041 (4.01)	2.0000003 (4.00)
2	2.0010919	2.0000664 (4.04)	2.0000041 (4.01)	2.0000003 (4.00)
2	2.0059537	2.0003900 (3.93)	2.0000247 (3.98)	2.0000015 (4.00)
3	3.0090182	3.0005863 (3.94)	3.0000370 (3.98)	3.0000023 (4.00)
3	3.0090182	3.0005863 (3.94)	3.0000370 (3.98)	3.0000023 (4.00)
! $\rightarrow$ 4	4.0300893	4.0020486 (3.88)	4.0001311 (3.97)	4.0000082 (3.99)
! $\rightarrow$ 4	4.0300893	4.0020486 (3.88)	4.0001311 (3.97)	4.0000082 (3.99)
5	5.0320266	5.0020970 (3.93)	5.0001333 (3.98)	5.0000084 (3.99)
5	5.0320266	5.0020970 (3.93)	5.0001333 (3.98)	5.0000084 (3.99)
5	5.0736899	5.0052310 (3.82)	5.0001333 (5.29)	5.0000212 (2.65)
5	5.0736899	5.0052310 (3.82)	5.0003371 (3.96)	5.0000212 (3.99)
6	-	6.0049765	6.0003192 (3.96)	6.0000201 (3.99)
6	-	-	-	-
DOF	1080	7344	53856	411840
EPS Solve Time (seconds)	0.0367	0.1339	1.5125	17.6765

Table 1. A comparison of how order 2 NCE and S^- finite elements solve the Maxwell cavity resonator eigenvalue problem, $\langle \mathrm{curl}(F), \mathrm{curl}(E) \rangle = \omega^2 \langle F, E \rangle$.

and are indicated in the chart by using parentheses. We use H(curl) to solve the problems, corresponding with edge elements in 3D. Based off earlier eigenvalue works [**?**], we expect that the rate of convergence be double the order of the finite element used to solve the problem. This is reflected in the table in most spots, except for one of the eigenvalues of 5, where it tends to oscillate a bit. This specific eigenvalue overall converges at a rate near 4.00 if we instead using the values at $N = 8$ and $N = 32$, ignoring the intermediate value at $N = 16$.

Based off the closed form formula, note that 4 should never be an eigenvalue in 3D. However, we see it consistently found by both the NCE and S^- elements. Any eigenvalue that has a $-$ spot is to be interpreted as the eigenvalue solver did not find that specific eigenvalue in the number of iterations it required to find the first 15 requested eigenvalue-eigenvector pairs.

Knowing that both elements are solving this problem in a fashion that is expected theoretically, we can analyze the rest of the results shown in this table. Investigating the error in the eigenvalues in the chart compared to the exact values, we see that NCE elements are able to get results that are up to a magnitude better, though not for every eigenvalue. However, this loss of accuracy from the trimmed Serendipity elements is made up for by the comparison of the DOFs and solve time required. At every mesh refinement level, trimmed Serendipity elements have nearly half the DOFs of NCE elements, and correspondingly, require about half the time to solve for the eigenvalues. At higher orders, we expect that this will be even more exaggerated.

The experiment was done by using SLEPc in Firedrake, computing an inverted shift to a target of 3.0, then asking SLEPc for 15 eigenvalue-eigenvector pairs. SLEPc was then give a tolerance level of $1e-7$, and then a couple of specific mumps parameters (icntl 14 set to 200 and icntl 13 set to 1). We ignored the eigenvalues of 1, as they correspond only to the boundary conditions.

Fig. 9. Preliminary results for testing a mixed Poisson problem in 3D, using $S^{-}(\mathrm{Div})$ and NCF elements.

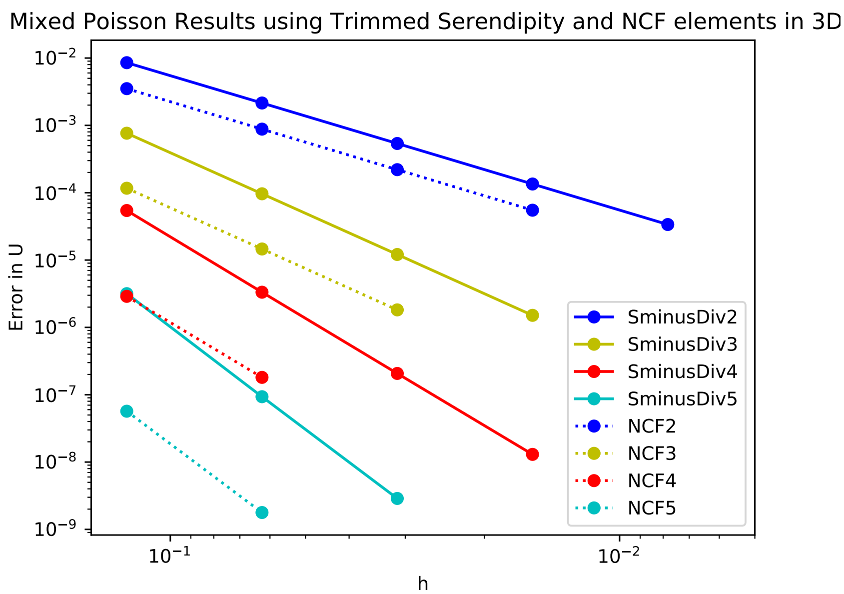


Fig. 10. Analyzing convergence rates of trimmed Serendipity vs tensor product elements in 3D using $H(\mathrm{div})$ elements.

7 CONCLUSION

Each finite element has a time and place where it could be considered beneficial to use. We explored the numerical properties of trimmed Serendipity elements to refine our understanding of how they act. From the theory, we knew that trimmed Serendipity elements should be able to converge at the same rate as tensor product and Serendipity elements, while using fewer degrees of freedom overall. The plots here demonstrate that the rate of convergence is consistent with what we expect at many orders for H(curl), H(div), and L^2 elements in both 2 and 3D.

Beyond the convergence rates hitting what we expect, we were able to analyze memory usage on these problems by studying the degrees of freedom required. Trimmed Serendipity, while generally have a worse error at a given order k, also uses significantly fewer degrees of freedom. This is illustrated well in the Maxwell Cavity Eigenvalue problem in **??**, where the degrees of freedom required were nearly half of what the tensor product elements used.

Another example of this is the 3D mixed Poisson problem, where we see that the trimmed Serendipity elements are able to be used at more refined meshes while the tensor product elements would need to be allotted more time on a high memory computer to be able to get results on the same sized mesh.

In general, it is clear from these results that trimmed Serendipity is not always a better choice compared to tensor product elements. However, these examples illustrates the benefit of trimmed Serendipity elements–in a setting where the mesh is fixed, the option to use a trimmed Serendipity element might give an extra way to refine a problem to increase the accuracy of a solution.