Defining custom elements in FEniCSx

Matthew Scroggs (University of Cambridge)

mscroggs.co.uk
mws48@cam.ac.uk
mscroggs
@mscroggs

Chris Richardson (University of Cambridge)

Garth Wells (University of Cambridge)

FEniCS 2022
San Diego

Ciarlet finite element

$$(R, \mathcal{V}, \mathcal{L})$$

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Ciarlet finite element

Reference element → $$(R, \mathcal{V}, \mathcal{L})$$

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Ciarlet finite element

$$(R, \mathcal{V}, \mathcal{L})$$

Reference element

Set of polynomials

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Ciarlet finite element

$$(R, \mathcal{V}, \mathcal{L})$$

Reference element

Set of polynomials

Set of functionals $\{l_0, l_1, \ldots, l_{n-1}\}$

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Ciarlet finite element

$$(R, \mathcal{V}, \mathcal{L})$$

Reference element

Set of polynomials

Set of functionals $\{l_0, l_1, \ldots, l_{n-1}\}$

Each functional l_i is associated with a sub-entity of the reference element

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Ciarlet finite element

$$(R, \mathcal{V}, \mathcal{L})$$

Reference element

Set of polynomials

Set of functionals $\{l_0, l_1, ..., l_{n-1}\}$

Each functional l_i is associated with a sub-entity of the reference element

Basis functions are defined by:

$$\phi_j \in \mathcal{V} \qquad l_i(\phi_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Ciarlet, P. G. The Finite Element Method for Elliptic Problems, 1978.

Example: Lagrange element

$$(R, \mathcal{V}, \mathcal{L})$$

Example: Lagrange element

$$(R, \mathcal{V}, \mathcal{L})$$

Triangle

Example: Lagrange element

$$(R \quad \mathcal{V} \quad \mathcal{L})$$

Triangle

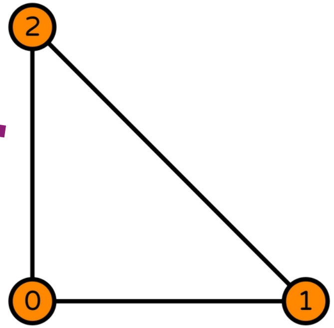
$$\mathrm{span}\{1, x, y\}$$

Example: Lagrange element

$(R, \mathcal{V}, \mathcal{L})$

Triangle

$\text{span}\{1, x, y\}$

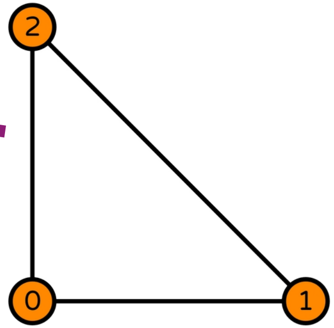


Example: Lagrange element

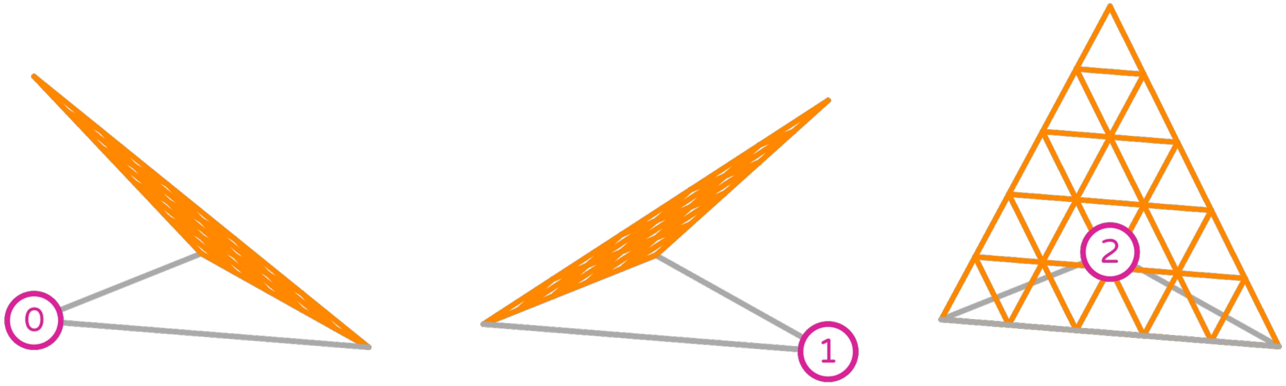
$(R, \mathcal{V}, \mathcal{L})$

Triangle

$\text{span}\{1, x, y\}$



$$\phi_0 = 1 - x - y$$
$$\phi_1 = x$$
$$\phi_2 = y$$



Example: TNT element

$$(R, \mathcal{V}, \mathcal{L})$$

Cockburn, Bernardo and Fu, Guosheng. A systematic construction of finite element commuting exact sequences, *SIAM journal on numerical analysis* 55, 1650–1688, 2017. [DOI: 10.1137/16M1073352]

Example: TNT element

$$(R, \mathcal{V}, \mathcal{L})$$

Quadrilateral

Cockburn, Bernardo and Fu, Guosheng. A systematic construction of finite element commuting exact sequences, *SIAM journal on numerical analysis* 55, 1650–1688, 2017. [DOI: 10.1137/16M1073352]

Example: TNT element

$$(R, \mathcal{V}, \mathcal{L})$$

Quadrilateral

$$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

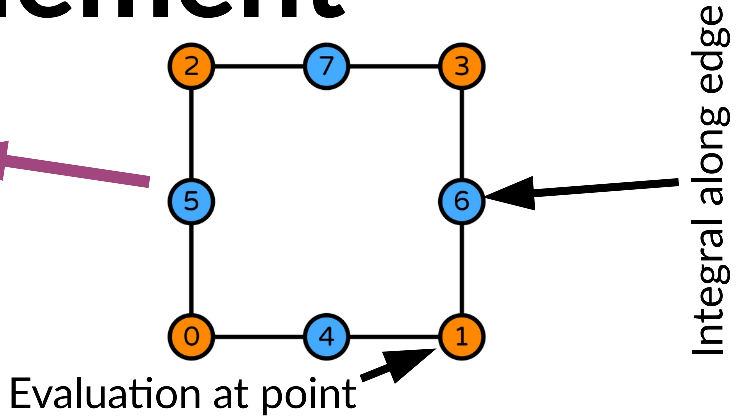
Cockburn, Bernardo and Fu, Guosheng. A systematic construction of finite element commuting exact sequences, *SIAM journal on numerical analysis* 55, 1650–1688, 2017. [DOI: 10.1137/16M1073352]

Example: TNT element

$$(R\ \mathcal{V}\ \mathcal{L})$$

Quadrilateral

$$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$



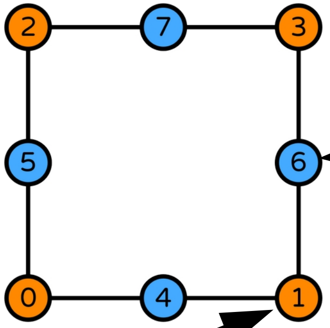
Cockburn, Bernardo and Fu, Guosheng. A systematic construction of finite element commuting exact sequences, *SIAM journal on numerical analysis* 55, 1650–1688, 2017. [DOI: 10.1137/16M1073352]

Example: TNT element

$(R, \mathcal{V}, \mathcal{L})$

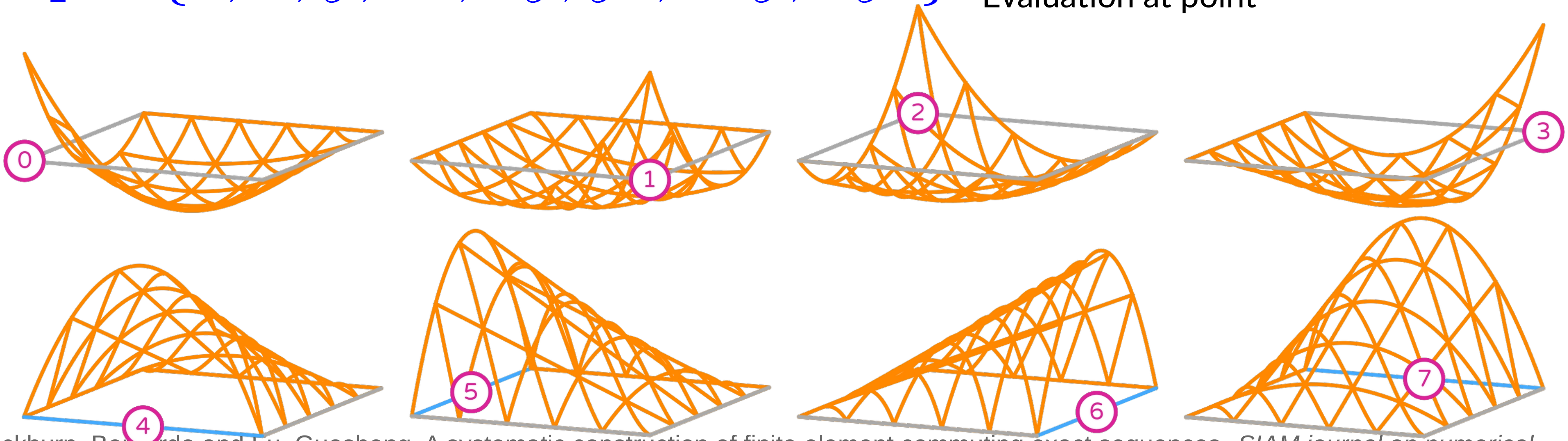
Quadrilateral

$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$



Integral along edge

Evaluation at point



Cockburn, Bernardo and Fu, Guosheng. A systematic construction of finite element commuting exact sequences, *SIAM journal on numerical analysis* 55, 1650–1688, 2017. [DOI: 10.1137/16M1073352]

Implementing Ciarlet elements

R

$\mathcal{V}$

$\mathcal{L}$

Basix

Finite element definition and tabulation library used by FEniCSx.

C++ with a Python interface.

https://github.com/FEniCS/basix

M. W. Scroggs, I. A. Baratta, C. N. Richardson & G. N. Wells. Basix: a runtime finite element basis evaluation library, *Journal of Open Source Software* 7(73), 2022, 3982. [DOI: 10.21105/joss.03982]

Implementing Ciarlet elements

R

$\mathcal{V}$

$\mathcal{L}$

Implementing Ciarlet elements

R `basix.CellType.triangle`, `basix.CellType.quadrilateral`, etc

$\mathcal{V}$

$\mathcal{L}$

Implementing Ciarlet elements

R `basix.CellType.triangle`,
`basix.CellType.quadrilateral`, etc

$\mathcal{V}$ Coefficients of basis of $\mathcal{V}$ in terms of orthonormal polynomials

$\mathcal{L}$

Implementing Ciarlet elements

R `basix.CellType.triangle`, `basix.CellType.quadrilateral`, etc

$\mathcal{V}$ Coefficients of basis of $\mathcal{V}$ in terms of orthonormal polynomials

$\mathcal{L}$ For each sub-entity:

Implementing Ciarlet elements

R `basix.CellType.triangle`,
`basix.CellType.quadrilateral`, etc

$\mathcal{V}$ Coefficients of basis of $\mathcal{V}$ in terms of orthonormal polynomials

$\mathcal{L}$ For each sub-entity:

- Set of points

Implementing Ciarlet elements

R `basix.CellType.triangle`, `basix.CellType.quadrilateral`, etc

$\mathcal{V}$ Coefficients of basis of $\mathcal{V}$ in terms of orthonormal polynomials

$\mathcal{L}$ For each sub-entity:

- Set of points
- “Matrix” defining how values at these points can be combined to evaluate functionals. Shape of these is:

number of DOFs × value size × number of points × number of derivatives

R

TNT elements in Basix

`basix.CellType.quadrilateral`

$\mathcal{V}$

TNT elements in Basix

$$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

$\mathcal{V}$ TNT elements in Basix

$$\operatorname{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

Degree 3 orthonormal polynomials:

$1, \sqrt{3}(2y-1), \sqrt{5}(6y^2-6y+1), \sqrt{3}(2x-1), 12xy-6x-6y+3, \sqrt{15}(12xy^2-12xy+2x-6y^2+6y-1),$

$\sqrt{5}(6x^2-6x+1), \sqrt{15}(12x^2y-6x^2-12xy+6x+2y-1),$

$180x^2y^2-180x^2y+30x^2-180xy^2+180xy-30x+30y^2-30y+5$

$\mathcal{V}$

TNT elements in Basix

$$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

Degree 3 orthonormal polynomials:

$1, \sqrt{3}(2y-1), \sqrt{5}(6y^2-6y+1), \sqrt{3}(2x-1), 12xy-6x-6y+3, \sqrt{15}(12xy^2-12xy+2x-6y^2+6y-1),$
$\sqrt{5}(6x^2-6x+1), \sqrt{15}(12x^2y-6x^2-12xy+6x+2y-1),$
$180x^2y^2-180x^2y+30x^2-180xy^2+180xy-30x+30y^2-30y+5$

TNT elements in Basix

$\mathcal{V}$

$$\text{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

Degree 3 orthonormal polynomials:

$1, \sqrt{3}(2y-1), \sqrt{5}(6y^2-6y+1), \sqrt{3}(2x-1), 12xy-6x-6y+3, \sqrt{15}(12xy^2-12xy+2x-6y^2+6y-1),$
$\sqrt{5}(6x^2-6x+1), \sqrt{15}(12x^2y-6x^2-12xy+6x+2y-1),$

~~$180x^2y^2-180x^2y+30x^2-180xy^2+180xy-30x+30y^2-30y+5$~~

TNT elements in Basix

$\mathcal{V}$

$$\operatorname{span}\{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$$

Degree 3 orthonormal polynomials:

$1, \sqrt{3}(2y-1), \sqrt{5}(6y^2-6y+1), \sqrt{3}(2x-1), 12xy -$

$\sqrt{5}(6x^2-6x+1), \sqrt{15}(12x^2y-6x^2-12xy+6x+$

~~$180x^2y^2-180x^2y+30x^2-180xy^2+180xy-30x+$~~

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

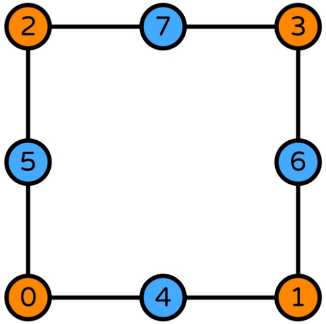
TNT elements in Basix

$\mathcal{V}$

```
wcoeffs = np.eye(8, 9)
```

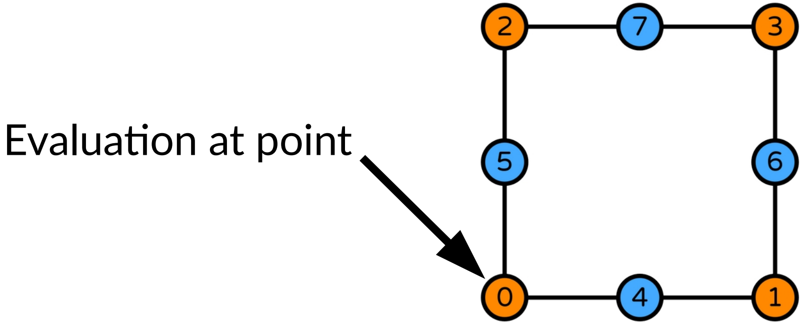
$\mathcal{L}$

TNT elements in Basix



TNT elements in Basix

$\mathcal{L}$

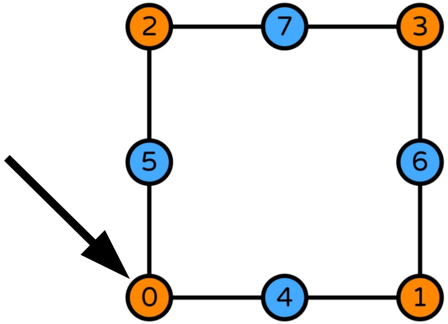


TNT elements in Basix

$\mathcal{L}$

Evaluation at point

$$\{(0,0)\}$$

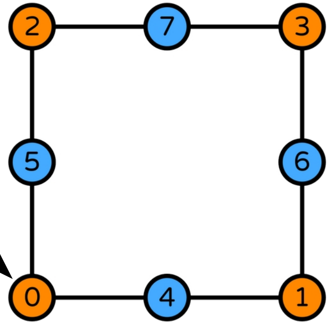


TNT elements in Basix

$\mathcal{L}$

Evaluation at point

$$\{(0,0)\}$$



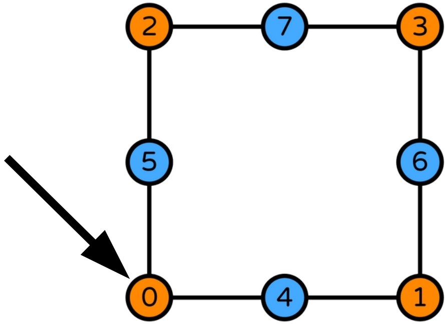
number of DOFs × value size × number of points × number of derivatives

$\mathcal{L}$

TNT elements in Basix

Evaluation at point

$$\{(0,0)\}$$



number of DOFs × value size × number of points × number of derivatives

1 × 1 × 1 × 1

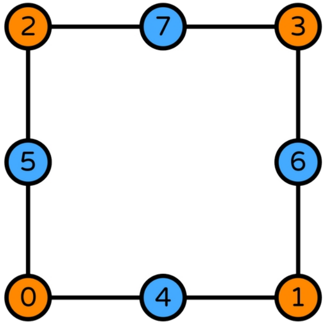
$\mathcal{L}$

TNT elements in Basix

```
geometry = basix.geometry(basix.CellType.quadrilateral)
topology = basix.topology(basix.CellType.quadrilateral)
x = [[], [], [], []]
M = [[], [], [], []]

for v in topology[0]:
    x[0].append(np.array(geometry[v]))
    M[0].append(np.array([[[[1.]]]]))
```

$\mathcal{L}$

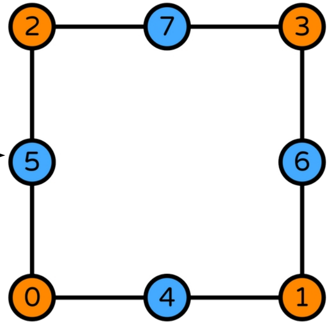
TNT elements in Basix



TNT elements in Basix

$\mathcal{L}$

Integral along edge

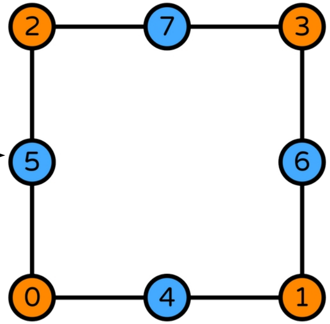


TNT elements in Basix

$\mathcal{L}$

Integral along edge

$\{\text{quadrature points}\}$

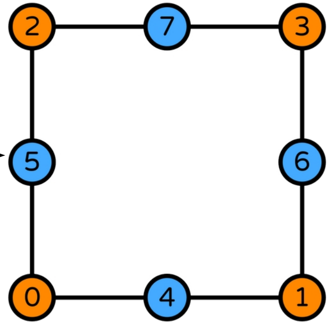


TNT elements in Basix

$\mathcal{L}$

Integral along edge

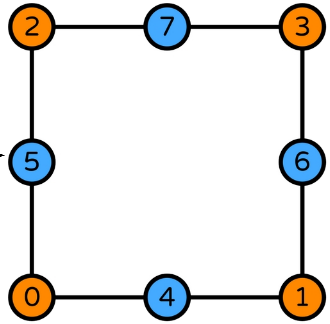
$$\{\text{quadrature points}\}$$



number of DOFs × value size × number of points × number of derivatives

TNT elements in Basix

$\mathcal{L}$



Integral along edge

$\{\text{quadrature points}\}$

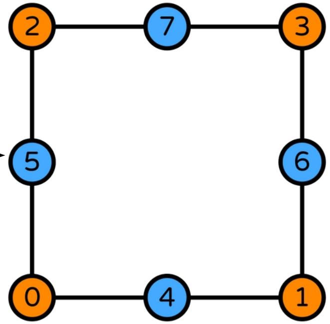
number of DOFs × value size × number of points × number of derivatives

1 × 1 × number of points × 1

TNT elements in Basix

$\mathcal{L}$

Integral along edge →



$\{\text{quadrature points}\}$

number of DOFs × value size × number of points × number of derivatives

1 × 1 × number of points × 1

Quadrature weights

TNT elements in Basix

$\mathcal{L}$

```
pts, wts = basix.make_quadrature(basix.CellType.interval, 2)
for e in topology[1]:
    v0 = geometry[e[0]]
    v1 = geometry[e[1]]
    edge_pts = np.array([v0 + p * (v1 - v0) for p in pts])
    x[1].append(edge_pts)

    mat = np.zeros((1, 1, pts.shape[0], 1))
    mat[0, 0, :, 0] = wts
    M[1].append(mat)
```

TNT elements in Basix

$\mathcal{L}$

```
x[2].append(np.zeros([0, 2]))
M[2].append(np.zeros([0, 1, 0, 1]))
```

TNT elements in Basix

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

TNT elements in Basix

Value shape

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

TNT elements in Basix

Value shape

Number of derivatives used by functionals

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

TNT elements in Basix

Number of derivatives used by functionals

Value shape

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

Push forward / pull back map type

TNT elements in Basix

Value shape

Number of derivatives
used by functionals

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

Push forward / pull back
map type

Is the element
discontinuous?

TNT elements in Basix

Value shape

Number of derivatives
used by functionals

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

Push forward / pull back
map type

Is the element
discontinuous?

Highest complete degree

TNT elements in Basix

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

Value shape

Number of derivatives used by functionals

Push forward / pull back map type

Is the element discontinuous?

Highest complete degree

Highest polynomial degree

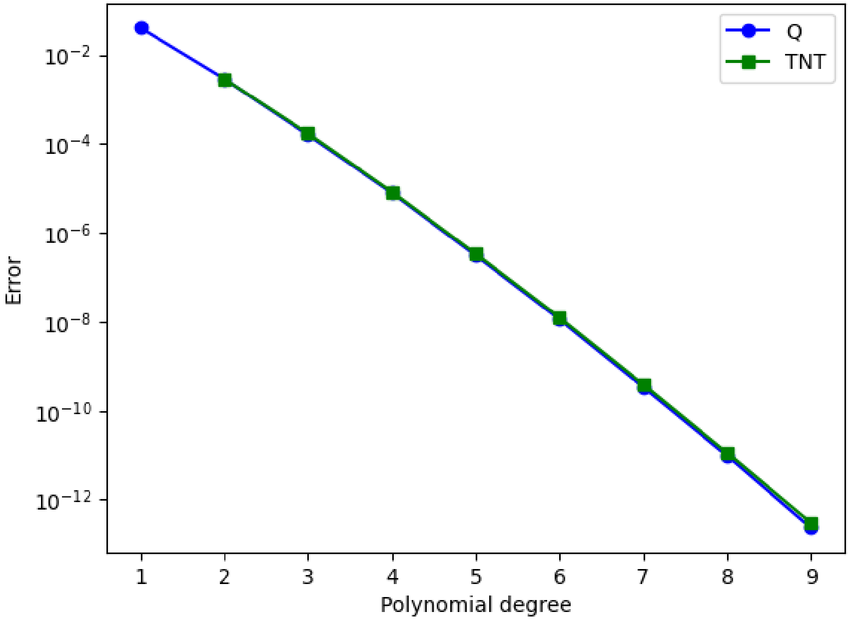
TNT elements in Basix

```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)
```

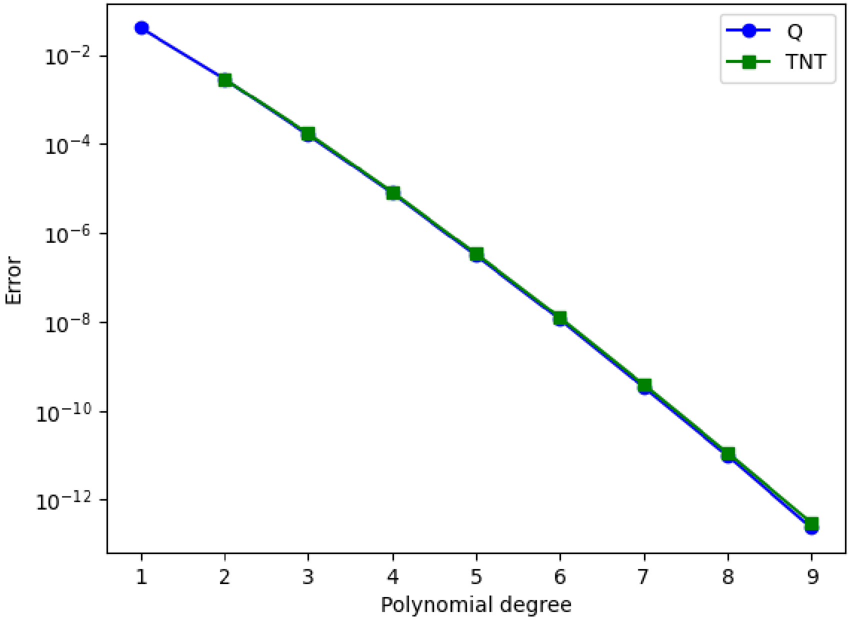
TNT elements in Basix

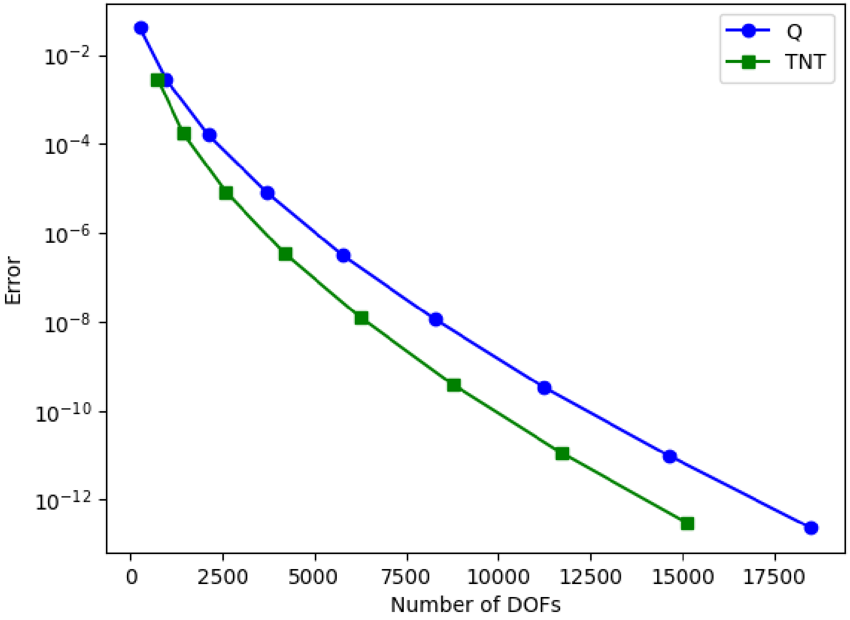
```
element = basix.create_custom_element(
    CellType.quadrilateral, [], wcoeffs, x, M, 0,
    MapType.identity, False, 1, 2)

import basix.ufl_wrapper
ufl_element = basix.ufl_wrapper.BasixElement(element)
```

TNT elements in DOLFINx



TNT elements in DOLFINx



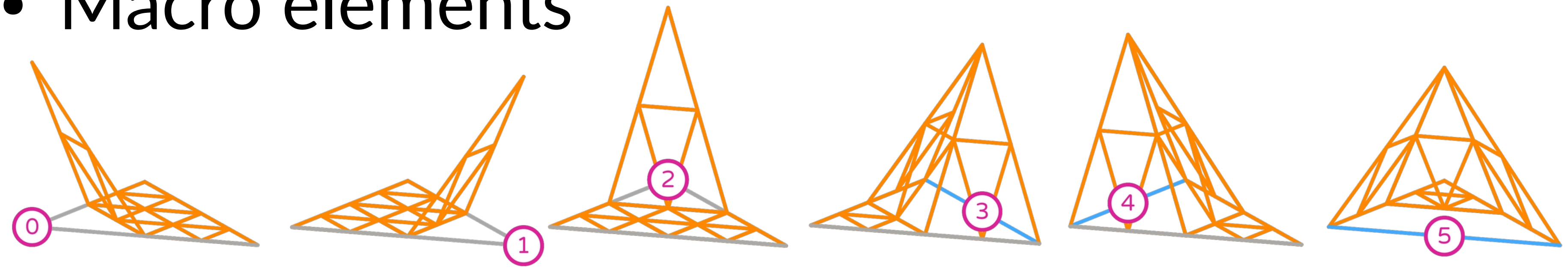


Summary

- Custom elements can be implemented in Basix using an implementation based on the Ciarlet definition
- Custom elements can be used with DOLFINx via Basix's UFL wrapper

Future work

- Macro elements



- H2 and H3 elements

Thanks for listening!

Matthew Scroggs (University of Cambridge / University College London)

mscroggs.co.uk
matthew.scroggs.14@ucl.ac.uk
mscroggs
@mscroggs

Chris Richardson (University of Cambridge)

Garth Wells (University of Cambridge)

FEniCS 2022
San Diego