Bernstein-Bezier Techniques in High Order Finite Elements

Mark Ainsworth

Division of Applied Mathematics

Brown University

Mark_Ainsworth@brown.edu

BROWN

Outline

- Bernstein-Bezier Polynomials
- De Casteljau Algorithm
- Sum-Factorisation and AAD Algorithm
- Bernstein-Bezier Basis for Raviart-Thomas Elements
- Applications

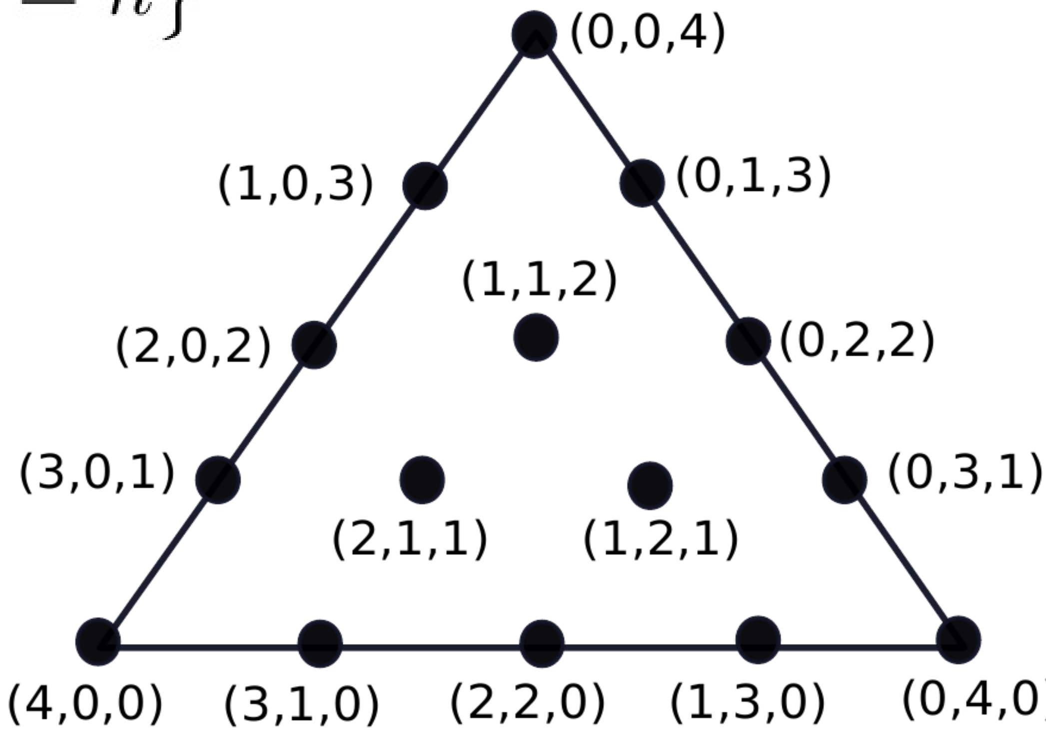
Nomenclature

$$\mathcal{I}_n = \{\boldsymbol{\alpha} \in \mathbb{Z}_+^3 : \ |\boldsymbol{\alpha}| = n\}$$

'Multi-Indices'

'Domain Points'

$$\boldsymbol{x}_{\boldsymbol{\alpha}} = \frac{1}{n}\sum_{k=1}^{3} \alpha_k \boldsymbol{x}_k$$



Case: n=4

BROWN

Bernstein-Bezier Polynomials

$$\boxed{B_{\boldsymbol{\alpha}}^n(\boldsymbol{x}) = \binom{n}{\boldsymbol{\alpha}} \boldsymbol{\lambda}^{\boldsymbol{\alpha}}, \quad \boldsymbol{\alpha} \in \mathcal{I}_n}$$

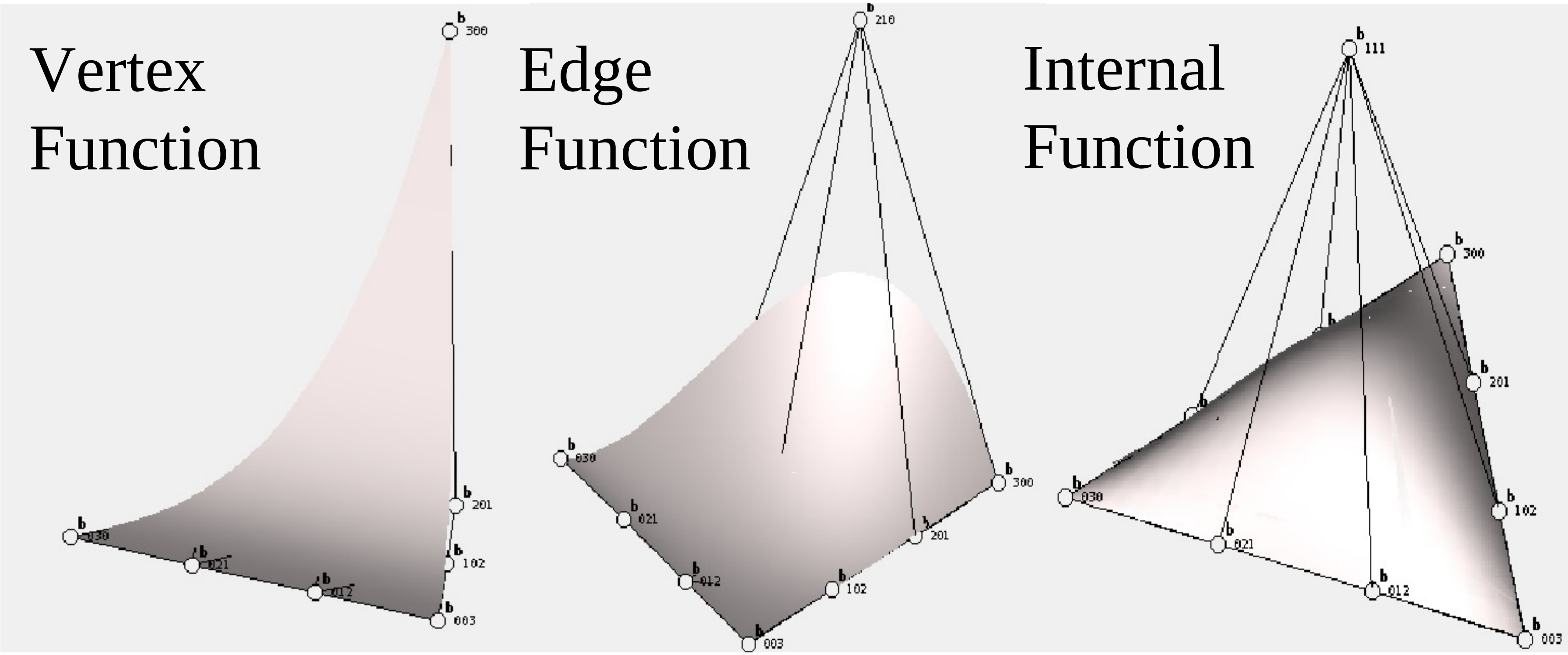
$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ *barycentric coordinates*

Non-negative, partition of unity $\sum_{\boldsymbol{\alpha} \in \mathcal{I}_n} B_{\boldsymbol{\alpha}}^n = 1.$

Natural identifications:

$$B_{\boldsymbol{\alpha}}^n(\boldsymbol{x}) \longleftrightarrow \boldsymbol{x}_{\boldsymbol{\alpha}} \longleftrightarrow \boldsymbol{\alpha} \in \mathcal{I}_n$$

Bernstein-Bezier Polynomials



Typical degree 3 Bernstein-Bezier Polynomials

Interactive View

BROWN

Why Bernstein-Bezier?

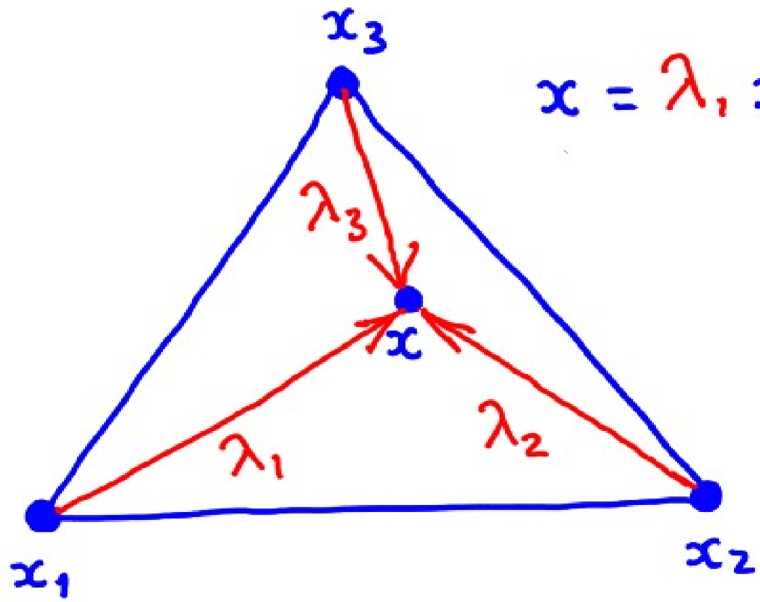
- Elegant, efficient and stable algorithms. e.g. de Casteljau, ...
- Industry standard for graphics. e.g. psfonts defined as Bezier curves, CAD/CAM packages use Bezier extensively.
- Industry standard for graphics hardware. e.g. OpenGL hardware optimised routines to render Bezier curves and surfaces.

Some Nice Properties of Bernstein Polynomials

$$\int_T B^n_\alpha(\mathbf{x})\,\mathrm{d}\mathbf{x} = \frac{|T|}{\binom{n+d}{d}}, \quad \alpha \in \mathcal{I}^n_d$$

$$B^m_\alpha B^n_\beta = \frac{\binom{\alpha+\beta}{\alpha}}{\binom{m+n}{m}} B^{m+n}_{\alpha+\beta}, \quad \alpha \in \mathcal{I}^m_d, \beta \in \mathcal{I}^n_d$$

De Casteljau Algorithm ($n=1$)

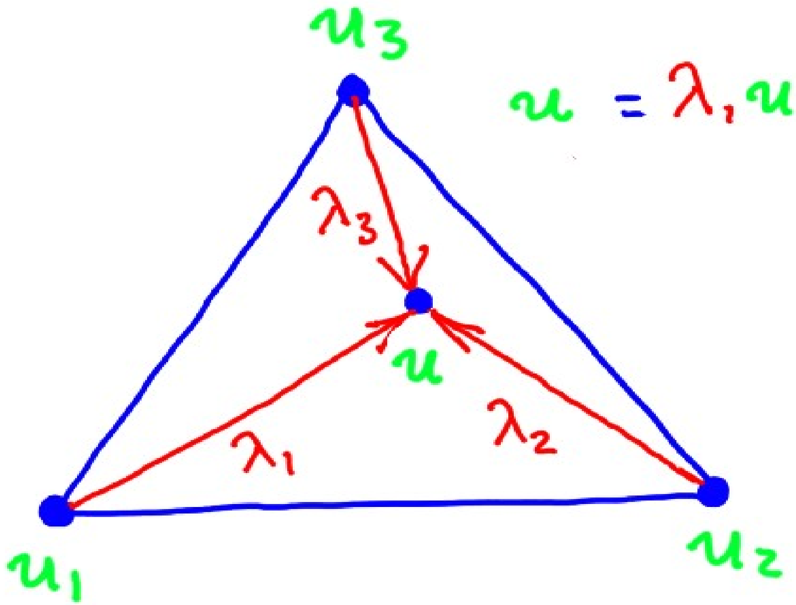


$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$$

How to evaluate BB poly ($n=1$) at x?

De Casteljau Algorithm (n=1)

Replace coordinates by control points

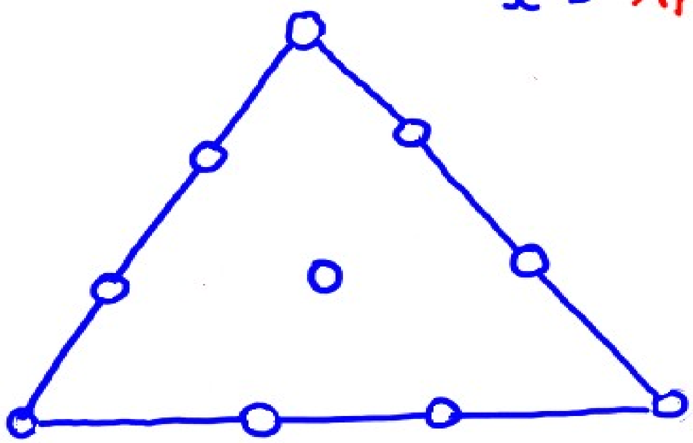


$$u = \lambda_1 u_1 + \lambda_2 u_2 + \lambda_3 u_3$$

... simply linear interpolation.

De Casteljau Algorithm ($n=3$)

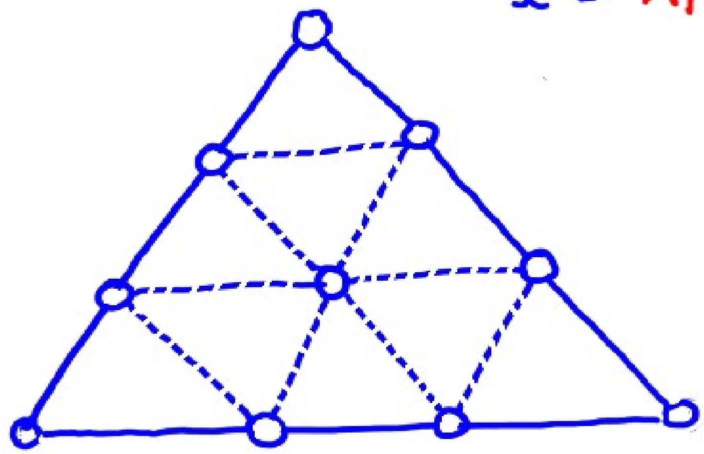
$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$



How to evaluate BB poly ($n=3$) at x?

De Casteljau Algorithm ($n=3$)

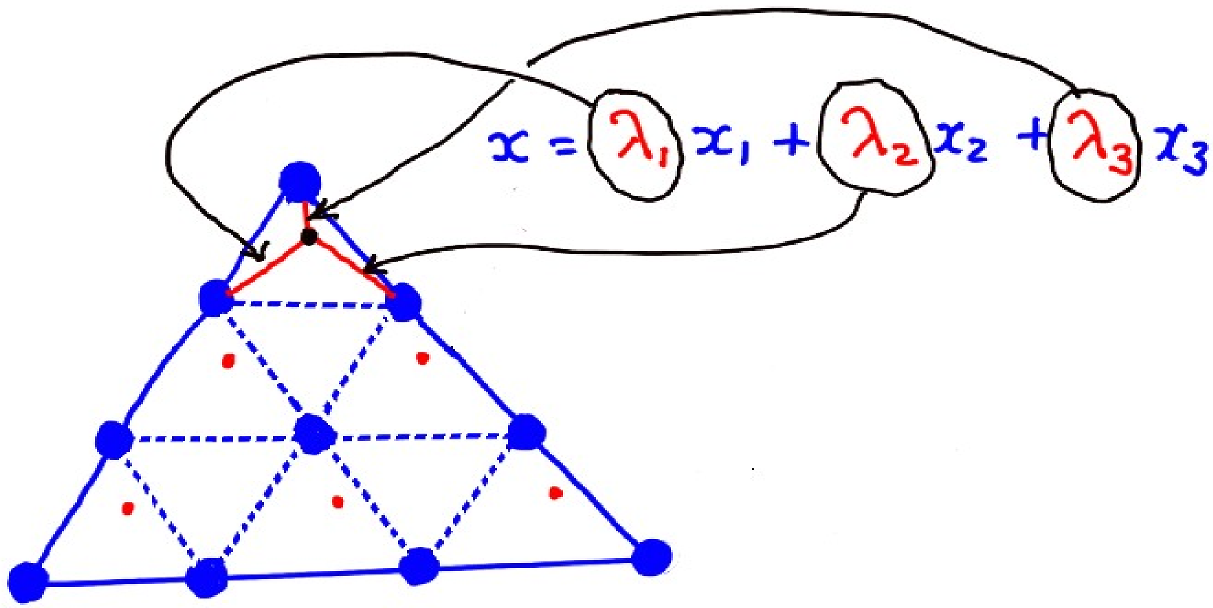
$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$



Introduce (virtual) micro-mesh

BROWN

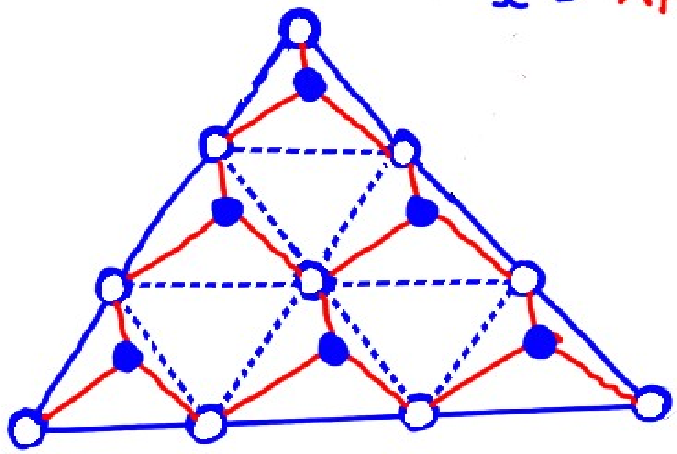
De Casteljau Algorithm (n=3)



LOCAL linear interpolation (as for n=1)

De Casteljau Algorithm (n=3)

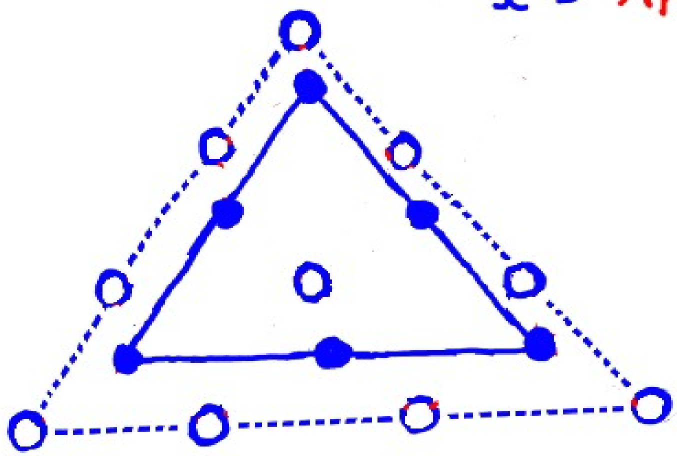
$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$$



LOCAL linear interpolation (as for n=1)

BROWN

De Casteljau Algorithm (n=3)

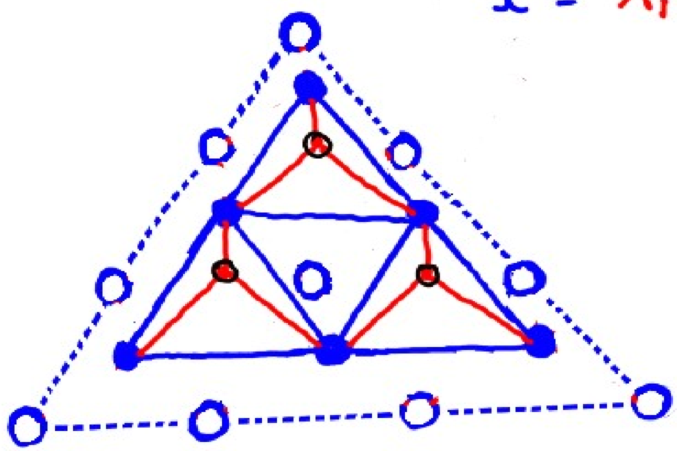


$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$$

Form lattice from "new control points"

BROWN

De Casteljau Algorithm (n=3)



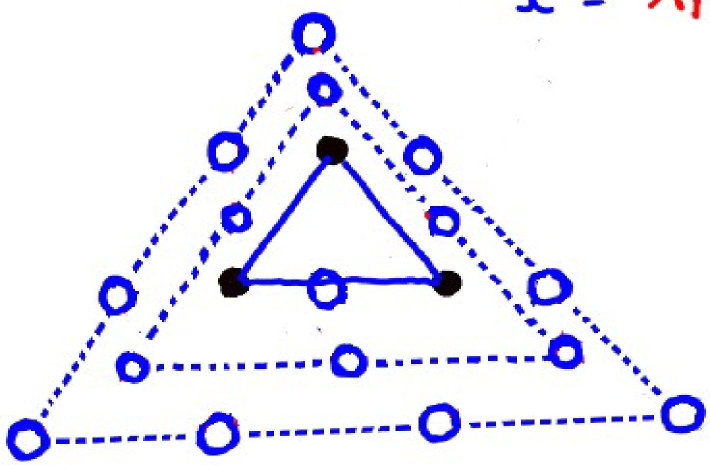
$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$

Perform **LOCAL** linear interpolation again

BROWN

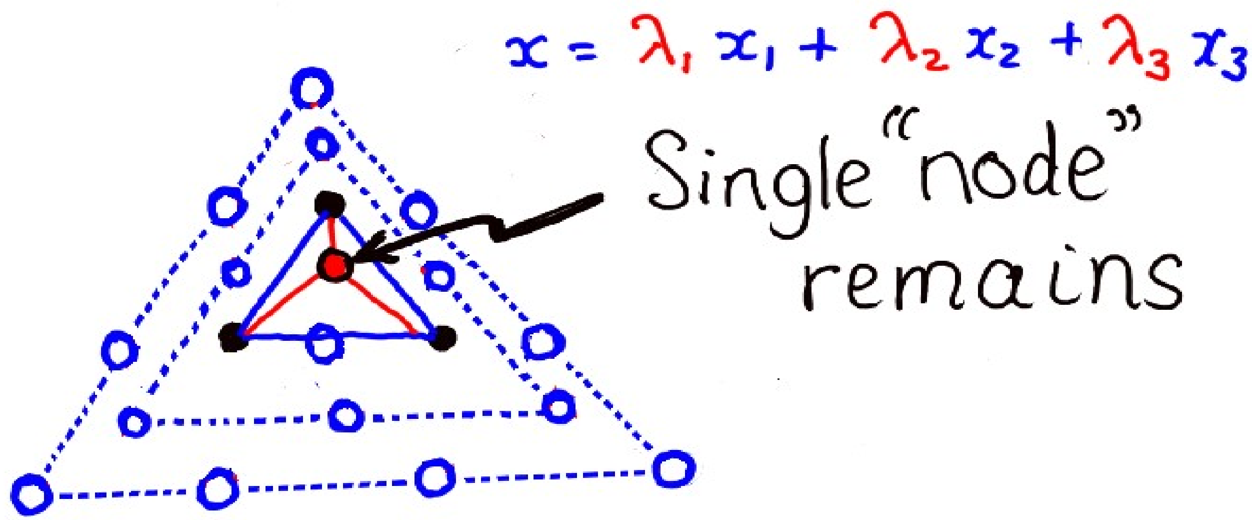
De Casteljau Algorithm (n=3)

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$$



Form lattice from "new control points"

De Casteljau Algorithm (n=3)

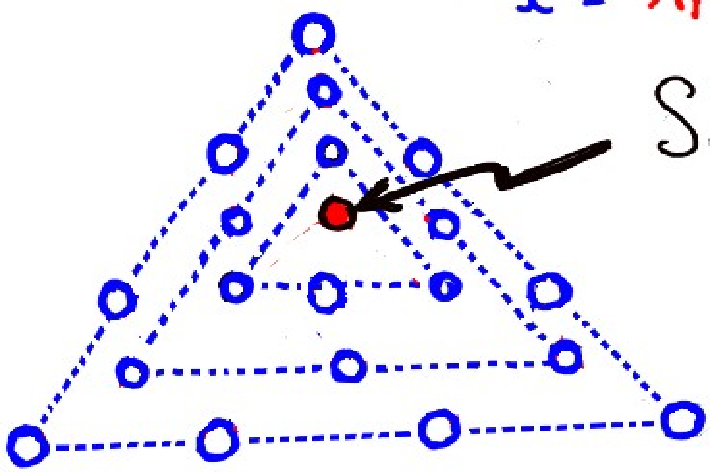


Perform linear interpolation again

BROWN

De Casteljau Algorithm (n=3)

$x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$



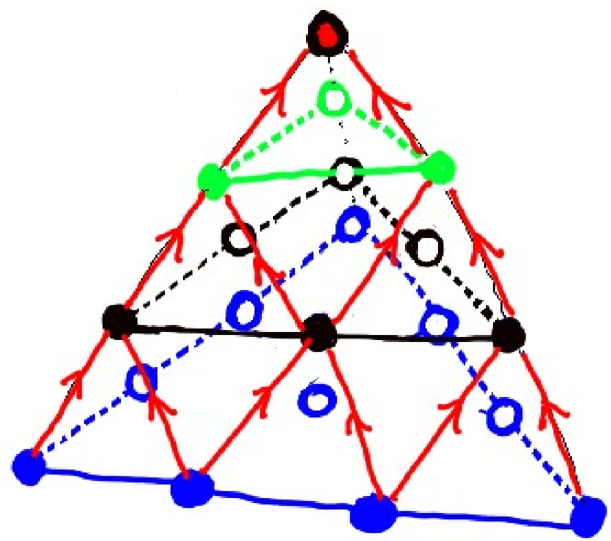
Single "node" remains

$=$

$u(x)$

Gives the value of cubic BB poly at x

De Casteljau Algorithm (n=3)



Stacking the arrays => Pyramid Algorithm

BROWN

BOOK: R. Goldman, *Pyramid Algorithms: A Dynamic Programming Approach to Curves and Surfaces for Geometric Modeling.*

Question

we consider following simple question:

what advantages do Bernstein polynomials offer

Question

we consider following simple question:

what advantages do Bernstein polynomials offer (if any)

Question

we consider following simple question:

what advantages do Bernstein polynomials offer (if any) for high order FEM?

Question

we consider following simple question:

what advantages do Bernstein polynomials offer (if any) for high order FEM?

Question motivated by:

- almost ubiquitous use of Bernstein polys in CAGD community
- ... and in spline literature.
- Similar philosophy to IGA (Hughes et al.)

BROWN

Bernstein-Bezier H^1 FEM

Previous work on using Bernstein-Bezier basis Awanou (PhD Thesis), Arnold et al. (2009), …
BUT don't take advantage of special properties of BB (could equally well used Lagrange basis).

Work seeking to exploit properties of BB:

* R.C. Kirby, *Numer. Math.*, (2011). Constant data, affine simplices in 2D/3D.
* Ainsworth, Andriamaro & Davydov, *SIAM J. Sci. Comp.*, (2011). Variable data, curvilinear elements, non-linear problems, simplices, prisms, bricks, …, any dimension.

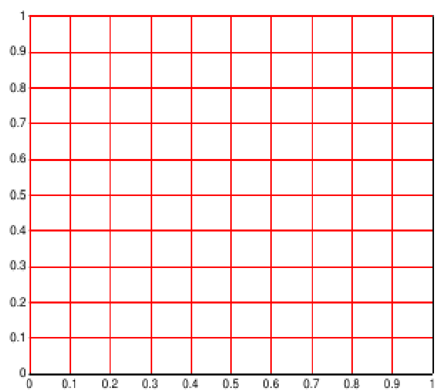
Duffy Transformation

Define $\mathbf{x} : [0,1]^d \to T = \mathrm{conv}(\mathbf{x}_1, \ldots, \mathbf{x}_{\mathrm{d}+1})$ by rule

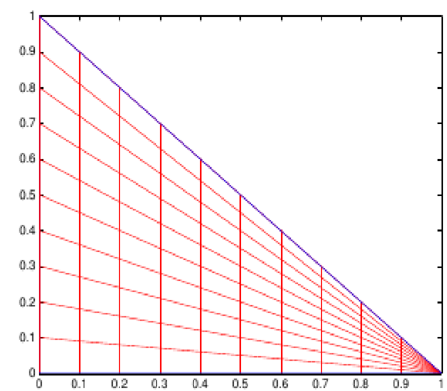
$$\mathbf{x}(\mathbf{t}) = \sum_{k=1}^{d+1} \lambda_k \mathbf{x}_k$$

where

$$\lambda_1 = t_1, \lambda_2 = t_2(1-\lambda_1), \ldots, \lambda_d = t_d(1 - \lambda_1 - \cdots - \lambda_{d-1}).$$



$\Longrightarrow$



BROWN

Ref: Duffy '81, Dubiner '92, Warburton et al. '99

Stroud Conical Quadrature Rule

Duffy transformation gives

$$\int_T f(\mathbf{x})\,\mathrm{d}\mathbf{x} =$$

$$d!|T| \int_0^1 \mathrm{d}t_1(1-t_1)^{d-1} \int_0^1 \mathrm{d}t_2(1-t_2)^{d-2} \cdots \int_0^1 \mathrm{d}t_d(f \circ \mathbf{x})(\mathbf{t}).$$

Approximate integral over t_k variable by Gauss-Jacobi rule:

$$\int_0^1 (1-s)^{d-k} g(s)\,\mathrm{d}s \approx \sum_{j=1}^{q} \omega_j^{(d-k)} g(\xi_j^{(d-k)})$$

BROWN

Stroud Conical Quadrature Rule

Gives

$$\int_T f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx d!|T| \sum_{i_1=1}^{q} \omega_{i_1}^{(d-1)} \sum_{i_2=1}^{q} \omega_{i_2}^{(d-2)} \cdots \sum_{i_d=1}^{q} \omega_{i_d}^{(0)} f(\mathbf{x}_{i_1,i_2,\ldots,i_d}).$$

"Stroud conical quadrature"

- positive quadrature weights
- quadrature nodes on T

$$\mathbf{x}_{i_1,i_2,\ldots,i_d} = \mathbf{x}(\xi_{i_1}^{(d-1)}, \xi_{i_2}^{(d-2)} \ldots, \xi_{i_d}^{(0)})$$

$$1 \leq i_1, i_2, \ldots, i_d \leq q.$$

BROWN

Bernstein Polynomials & Duffy

How does Bernstein polynomial behave under Duffy transformation? $\mathbf{x}(\mathbf{t}) : [0,1]^d \to T$

Consider univariate Bernstein polynomial

$$B_k^m(t) = \binom{m}{k} t^k (1-t)^{m-k}, k \in \{0, 1, \ldots, m\}$$

then

$$B_\alpha^n(\mathbf{x}(\mathbf{t})) = B_{\alpha_1}^n(t_1) B_{\alpha_2}^{n-\alpha_1}(t_2) \cdots B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(t_d)$$

Tensorial Nature

BROWN

Bernstein Polynomials & Duffy

KEY OBSERVATION:

$$B_{\alpha}^{n}(\mathbf{x}(\mathbf{t})) = B_{\alpha_1}^{n}(t_1) B_{\alpha_2}^{n-\alpha_1}(t_2) \cdots B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(t_d)$$

Bernstein polynomials possess key property needed for *Sum Factorisation Algorithm.*

Ref. Orszag, 1980

BUT basis not tied to a tensorial construction.

BROWN

Application: Evaluation of BBFEM

How to *efficiently* evaluate a BBFEM approx at all of Stroud points? $\mathbf{x}_{i_1,i_2,\ldots,i_d}$

$$u(\mathbf{x}) = \sum_{\alpha \in \mathcal{I}_d^n} c_\alpha B_\alpha^n(\mathbf{x})$$

BROWN

Application: Evaluation of BBFEM

How to *efficiently* evaluate a BBFEM approx at all of Stroud points? $\mathbf{x}_{i_1,i_2,\ldots,i_d}$

$$u(\mathbf{x}) = \sum_{\alpha \in \mathcal{I}_d^n} c_\alpha B_\alpha^n(\mathbf{x})$$

Method 1: Apply de Casteljau Algorithm.

=> Cost of $\mathcal{O}(n^{d+1})$ per point.

BROWN

Application: Evaluation of BBFEM

How to *efficiently* evaluate a BBFEM approx at all of Stroud points? $\mathbf{x}_{i_1,i_2,\ldots,i_d}$

$$u(\mathbf{x}) = \sum_{\alpha \in \mathcal{I}_d^n} c_\alpha B_\alpha^n(\mathbf{x})$$

Method 2: Apply sum factorisation.

Application: Evaluation of BBFEM

$$(u \circ x)(t) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(t_1) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(t_2) \sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B^{n-\alpha_1-\alpha_2}_{\alpha_3}(t_3)\, c_{\alpha_1\alpha_2\alpha_3}$$

using KEY OBSERVATION, where x is Duffy transformation.

Application: Evaluation of BBFEM

$$(u \circ x)(t) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^0_{i_1}) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^1_{i_2}) \sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B^{n-\alpha_1-\alpha_2}_{\alpha_3}(\xi^2_{i_3})\, C_{\alpha_1\alpha_2\alpha_3}$$

where $t = (\xi^0_{i_1}, \xi^1_{i_2}, \xi^2_{i_3})$,

$0 \le i_1, i_2, i_3 \le q$

i.e. want to evaluate at Stroud points.

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^0_{i_1}) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^1_{i_2}) \sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B^{n-\alpha_1-\alpha_2}_{\alpha_3}(\xi^2_{i_3})\, C_{\alpha_1\alpha_2\alpha_3}$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^n_{\alpha_1}(\xi^0_{i_1}) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^1_{i_2}) \underbrace{\sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B^{n-\alpha_1-\alpha_2}_{\alpha_3}(\xi^2_{i_3}) \, c_{\alpha_1\alpha_2\alpha_3}}_{\overset{\text{define}}{=} \; C^1(\alpha_1,\alpha_2,i_3)}$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^{0}_{i_1}) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^{1}_{i_2})\, C^{1}(\alpha_1, \alpha_2, i_3)$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^0_{i_1}) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^1_{i_2}) C^1(\alpha_1, \alpha_2, i_3)$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^{0}_{i_1}) \underbrace{\sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^{1}_{i_2}) C^{1}(\alpha_1, \alpha_2, i_3)}_{\overset{\text{def}}{=}\; C^{2}(\alpha_1, i_2, i_3)}$$

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \sum_{\alpha_1=0}^{n} B^n_{\alpha_1}(\xi^0_{i_1})\, C^2(\alpha_1, i_2, i_3)$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = \underbrace{\sum_{\alpha_1=0}^{n} B^n_{\alpha_1}(\xi^0_{i_1})\, C^2(\alpha_1, i_2, i_3)}_{\substack{\| \text{ def} \\ C^3(i_1, i_2, i_3)}}$$

BROWN

Application: Evaluation of BBFEM

$$(u \circ x)(\underset{\sim}{t}) = C^3(i_1, i_2, i_3)$$

where $t = (\xi^0_{i_1}, \xi^1_{i_2}, \xi^2_{i_3})$

$0 \le i_1, i_2, i_3 \le q$

Application: Evaluation of BBFEM

$$C^3(i_1, i_2, i_3) = \sum_{\alpha_1=0}^{n} B^n_{\alpha_1}(\xi^0_{i_1})\, C^2(\alpha_1, i_2, i_3)$$

$$0 \le i_1, i_2, i_3 \le q$$

BROWN

Application: Evaluation of BBFEM

$$C^2(\alpha_1, i_2, i_3) = \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(\xi^1_{i_2})\, C^1(\alpha_1, \alpha_2, i_3)$$

$$C^3(i_1, i_2, i_3) = \sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(\xi^0_{i_1})\, C^2(\alpha_1, i_2, i_3)$$

$$0 \le i_1, i_2, i_3 \le q$$

BROWN

Application: Evaluation of BBFEM

$$C^2(\alpha_1, i_2, i_3) = \sum_{\alpha_2=0}^{n-\alpha_1} B_{\alpha_2}^{n-\alpha_1}(\xi_{i_2}^1)\, C^1(\alpha_1, \alpha_2, i_3)$$

$$C^3(i_1, i_2, i_3) = \sum_{\alpha_1=0}^{n} B_{\alpha_1}^{n}(\xi_{i_1}^0)\, C^2(\alpha_1, i_2, i_3)$$

$$0 \le i_1, i_2, i_3 \le q$$

BROWN

Application: Evaluation of BBFEM

$$C^1(\alpha_1,\alpha_2,i_3) = \sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B_{\alpha_3}^{n-\alpha_1-\alpha_2}(\xi_{i_3}^2)\, c_{\alpha_1\alpha_2\alpha_3}$$

(Given) control points

$$C^2(\alpha_1,i_2,i_3) = \sum_{\alpha_2=0}^{n-\alpha_1} B_{\alpha_2}^{n-\alpha_1}(\xi_{i_2}^1)\, C^1(\alpha_1,\alpha_2,i_3)$$

$$C^3(i_1,i_2,i_3) = \sum_{\alpha_1=0}^{n} B_{\alpha_1}^{n}(\xi_{i_1}^0)\, C^2(\alpha_1,i_2,i_3)$$

$$0 \le i_1, i_2, i_3 \le q$$

BROWN

Application: Evaluation of BBFEM

$$C^1(\alpha_1,\alpha_2,i_3) = \sum_{\alpha_3=0}^{n-\alpha_1-\alpha_2} B_{\alpha_3}^{n-\alpha_1-\alpha_2}(\xi_{i_3}^2)\, c_{\alpha_1\alpha_2\alpha_3}$$

$$\downarrow \textcircled{2}$$

$$C^2(\alpha_1,i_2,i_3) = \sum_{\alpha_2=0}^{n-\alpha_1} B_{\alpha_2}^{n-\alpha_1}(\xi_{i_2}^1)\, C^1(\alpha_1,\alpha_2,i_3)$$

$$\downarrow \textcircled{3}$$

$$C^3(i_1,i_2,i_3) = \sum_{\alpha_1=0}^{n} B_{\alpha_1}^{n}(\xi_{i_1}^0)\, C^2(\alpha_1,i_2,i_3)$$

$$0 \le i_1, i_2, i_3 \le q$$

BROWN

Application: Evaluation of BBFEM

Step 1: Apply KEY OBSERVATION to write

$$(u \circ \mathbf{x})(\mathbf{t}) =$$

$$\sum_{\alpha_1=0}^{n} B^{n}_{\alpha_1}(t_1) \sum_{\alpha_2=0}^{n-\alpha_1} B^{n-\alpha_1}_{\alpha_2}(t_2) \dots \sum_{\alpha_d=0}^{n-\alpha_1-\dots-\alpha_{d-1}} B^{n-\alpha_1-\dots-\alpha_{d-1}}_{\alpha_d}(t_d)$$

BROWN

Application: Evaluation of BBFEM

Step 1: Apply KEY OBSERVATION to write

$$(u \circ \mathbf{x})(\mathbf{t}) = \sum_{\alpha_1=0}^{n} B_{\alpha_1}^{n}(t_1) \sum_{\alpha_2=0}^{n-\alpha_1} B_{\alpha_2}^{n-\alpha_1}(t_2) \ldots \sum_{\alpha_d=0}^{n-\alpha_1-\ldots-\alpha_{d-1}} B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(t_d)$$

Step 2: Express in recursive form

$$\begin{aligned}
C^0(\alpha_1, \ldots, \alpha_{d-1}, \alpha_d) &= c_{\alpha_1,\ldots,\alpha_{d-1},\alpha_d} \\
C^1(\alpha_1, \ldots, \alpha_{d-1}, i_d) &= \sum_{\alpha_d=0}^{n-\alpha_1-\ldots-\alpha_{d-1}} B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(\xi_{i_d}^{(0,0)}) C^0(\alpha_1, \alpha_2, \ldots, \alpha_d) \\
C^2(\alpha_1, \ldots, i_{d-1}, i_d) &= \sum_{\alpha_{d-1}=0}^{n-\alpha_1-\ldots-\alpha_{d-2}} B_{\alpha_{d-1}}^{n-\alpha_1-\ldots-\alpha_{d-2}}(\xi_{i_{d-1}}^{(1,0)}) C^1(\alpha_1, \ldots, \alpha_{d-1}, i_d) \\
&\vdots \\
C^d(i_1, \ldots, i_{d-1}, i_d) &= \sum_{\alpha_1=0}^{n} B_{\alpha_1}^{n}(\xi_{i_d}^{(d-1,0)}) C^{d-1}(\alpha_1, \ldots, i_{d-1}, i_d)
\end{aligned}$$

BROWN

Application: Evaluation of BBFEM

Recursion leads to

$$u(\mathbf{x}_{i_1,\ldots,i_{d-1},i_d}) = C^d(i_1,\ldots,i_{d-1},i_d)$$

at total cost for *all* points of

$$\mathcal{O}(n^{d+1})$$

BROWN

Application: Evaluation of BBFEM

Recursion leads to

$$u(\mathbf{x}_{i_1,\ldots,i_{d-1},i_d}) = C^d(i_1,\ldots,i_{d-1},i_d)$$

at total cost for all points of

$$\mathcal{O}(n^{d+1})$$

i.e. we get all points at same cost for de Casteljau to get a single point.

Application: Evaluation of BBFEM

Recursion leads to

$$u(\mathbf{x}_{i_1,\ldots,i_{d-1},i_d}) = C^d(i_1,\ldots,i_{d-1},i_d)$$

at total cost for all points of

$$\mathcal{O}(n^{d+1})$$

i.e. we get all points at same cost for de Casteljau to get a single point.

... including the cost of evaluating basis functions 'on the fly'.

BROWN

Evaluation of Moments

Bernstein-Bezier Moments of f defined by

$$\mu_{\alpha}^{n}(f) = \int_T B_{\alpha}^{n}(\mathbf{x}) f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha} \in \mathcal{I}_d^n.$$

... needed for element load vector.

Evaluation of Moments

Bernstein-Bezier Moments of f defined by

$$\mu_\alpha^n(f) = \int_T B_\alpha^n(\mathbf{x}) f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha} \in \mathcal{I}_d^n.$$

... needed for element load vector.

If data f constant, then have simple closed form

$$\mu_\alpha^n(f) = \frac{|T|}{\binom{n+d}{d}} f_{|T}, \quad \boldsymbol{\alpha} \in \mathcal{I}_d^n$$

Evaluation of Moments

Bernstein-Bezier Moments of f defined by

$$\mu_\alpha^n(f) = \int_T B_\alpha^n(\mathbf{x}) f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \alpha \in \mathcal{I}_d^n.$$

... needed for element load vector.

General data: need quadrature rule with $\mathcal{O}(q^d)$ Points where $q = \mathcal{O}(n)$

Total of $\mathcal{O}(n^d)$ moments => potentially costly.

BROWN

Evaluation of Moments

Duffy transformation and KEY OBSERVATION gives

$$\begin{aligned}\mu^n_{\alpha}(f) = \quad & d!|T| \int_0^1 \mathrm{d}t_d B^{n-\alpha_1-\ldots-\alpha_{d-1}}_{\alpha_d}(t_d) \\ & \cdot \int_0^1 \mathrm{d}t_{d-1}(1-t_{d-1}) B^{n-\alpha_1-\ldots-\alpha_{d-2}}_{\alpha_{d-1}}(t_{d-1}) \\ & \cdots \\ & \cdot \int_0^1 \mathrm{d}t_1 (1-t_1)^{d-1} B^n_{\alpha_1}(t_1)(f \circ \mathbf{x})(\mathbf{t})\end{aligned}$$

Evaluation of Moments

Apply Stroud conical quadrature rule to obtain recursive formulae:

$$
\begin{aligned}
F^0(i_1, i_2, \ldots, i_d) = &\quad (f \circ \mathbf{x})(\xi_{i_1}^{(d-1)}, \xi_{i_2}^{(d-2)}, \ldots, \xi_{i_d}^{(0)}) \\
F^1(\alpha_1, i_2, \ldots, i_d) = &\quad \sum_{i_1=1}^{q} \omega_{i_1}^{(d-1)} B_{\alpha_1}^{n}(\xi_{i_1}^{(d-1)}) F^0(i_1, i_2, \ldots, i_d) \\
F^2(\alpha_1, \alpha_2, \ldots, i_d) = &\quad \sum_{i_2=1}^{q} \omega_{i_2}^{(d-2)} B_{\alpha_2}^{n-\alpha_1}(\xi_{i_2}^{(d-2)}) F^1(\alpha_1, i_2, \ldots, i_d) \\
&\quad \vdots \\
F^d(\alpha_1, \alpha_2, \ldots, \alpha_d) = &\quad \sum_{i_d=1}^{q} \omega_{i_d}^{(0)} B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(\xi_{i_d}^{(0)}) F^{d-1}(\alpha_1, \alpha_2, \ldots,
\end{aligned}
$$

Evaluation of Moments

Apply Stroud conical quadrature rule to obtain recursive formulae:

$$
\begin{aligned}
F^0(i_1, i_2, \ldots, i_d) &= (f \circ \mathbf{x})(\xi_{i_1}^{(d-1)}, \xi_{i_2}^{(d-2)}, \ldots, \xi_{i_d}^{(0)}) \\
F^1(\alpha_1, i_2, \ldots, i_d) &= \textstyle\sum_{i_1=1}^{q} \omega_{i_1}^{(d-1)} B_{\alpha_1}^{n}(\xi_{i_1}^{(d-1)}) F^0(i_1, i_2, \ldots, i_d) \\
F^2(\alpha_1, \alpha_2, \ldots, i_d) &= \textstyle\sum_{i_2=1}^{q} \omega_{i_2}^{(d-2)} B_{\alpha_2}^{n-\alpha_1}(\xi_{i_2}^{(d-2)}) F^1(\alpha_1, i_2, \ldots, i_d) \\
&\vdots \\
F^d(\alpha_1, \alpha_2, \ldots, \alpha_d) &= \textstyle\sum_{i_d=1}^{q} \omega_{i_d}^{(0)} B_{\alpha_d}^{n-\alpha_1-\ldots-\alpha_{d-1}}(\xi_{i_d}^{(0)}) F^{d-1}(\alpha_1, \alpha_2, \ldots,
\end{aligned}
$$

Result of recursion

$$\mu_{\alpha}^{n}(f) = d!|T|F^d(\alpha_1, \alpha_2, \ldots, \alpha_d)$$

BROWN

Evaluation of Moments

Claim: The number of operations needed to compute $\mu_\alpha^n(f)$ is $\mathcal{O}(n^{d+1})$, even including the cost of evaluating basis functions on the fly.

- Can obtain element load vector at a cost of $\mathcal{O}(n^{d+1})$ operations, even with variable data f.
- Hence, curvilinear elements also handled in same complexity.

Ref: Ainsworth, Andriamaro & Davydov, SISC 2011

BROWN

Evaluation of Element Mass Matrix

$$\mathbf{M}^T_{\alpha\beta} = \int_T c(\mathbf{x}) B^n_{\boldsymbol{\alpha}}(\mathbf{x}) B^n_{\boldsymbol{\beta}}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{I}^n_d$$

Dimension $\binom{n+d}{d} \times \binom{n+d}{d}$ **i.e.** $\mathcal{O}(n^{2d})$

Is it possible to compute matrix in $\mathcal{O}(1)$ operation per entry? i.e. complexity $\mathcal{O}(n^{2d})$

Evaluation of Element Mass Matrix

$$\mathbf{M}^T_{\alpha\beta} = \int_T c(\mathbf{x}) B^n_{\alpha}(\mathbf{x}) B^n_{\beta}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{I}^n_d$$

Dimension $\binom{n+d}{d} \times \binom{n+d}{d}$ i.e. $\mathcal{O}(n^{2d})$

Is it possible to compute matrix in $\mathcal{O}(1)$ operation per entry? i.e. complexity $\mathcal{O}(n^{2d})$

Karniadakis & Sherwin approach gives $\mathcal{O}(n^{2d+1})$

BROWN

Evaluation of Element Mass Matrix

$$\mathbf{M}^T_{\alpha\beta} = \int_T c(\mathbf{x}) B^n_{\boldsymbol{\alpha}}(\mathbf{x}) B^n_{\boldsymbol{\beta}}(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{I}^n_d$$

Dimension $\binom{n+d}{d} \times \binom{n+d}{d}$ i.e. $\mathcal{O}(n^{2d})$

Is it possible to compute matrix in $\mathcal{O}(1)$ operation per entry? i.e. complexity $\mathcal{O}(n^{2d})$

Karniadakis & Sherwin approach gives $\mathcal{O}(n^{2d+1})$

Eibner & Melenk (2006) gives $\mathcal{O}(n^{2d})$

BuT requires 6-fold increase in dimension.

BROWN

Evaluation of Element Mass Matrix

Claim: Bernstein-Bezier basis achieves optimal complexity (without tinkering with the space).

$$\mathbf{M}^T_{\alpha\beta} = \int_T c(\mathbf{x}) B^n_\alpha(\mathbf{x}) B^n_\beta(\mathbf{x})\,\mathrm{d}\mathbf{x}, \quad \alpha, \beta \in \mathcal{I}^n_d$$

Recall property of Bernstein polynomials

$$B^n_\alpha B^n_\beta = \frac{\binom{\alpha+\beta}{\alpha}}{\binom{2n}{n}} B^{2n}_{\alpha+\beta}, \quad \alpha, \beta \in \mathcal{I}^n_d$$

BROWN

Evaluation of Element Mass Matrix

Claim: Bernstein-Bezier basis achieves optimal complexity (without tinkering with the space).

$$\mathbf{M}^T_{\alpha\beta} = \int_T c(\mathbf{x}) B^n_{\alpha}(\mathbf{x}) B^n_{\beta}(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \quad \alpha, \beta \in \mathcal{I}^n_d$$

Hence,

$$\mathbf{M}^T_{\alpha\beta} = \frac{\binom{\alpha+\beta}{\alpha}}{\binom{2n}{n}} \mu^{2n}_{\alpha+\beta}(c), \quad \alpha, \beta \in \mathcal{I}^n_d$$

BROWN

Evaluation of Element Mass Matrix

Apply AAD Algorithm to compute the moments

$$\mu^{2n}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(c)$$

Complexity: $\mathcal{O}((2n)^{d+1})$

Evaluation of Element Mass Matrix

Apply AAD Algorithm to compute the moments

$$\mu^{2n}_{\alpha+\beta}(c)$$

Complexity: $\mathcal{O}((2n)^{d+1})$

$$\mathbf{M}^T_{\alpha\beta} = \frac{\binom{\alpha+\beta}{\alpha}}{\binom{2n}{n}} \mu^{2n}_{\alpha+\beta}(c), \quad \alpha, \beta \in \mathcal{I}^n_d$$

Remarkably, multinomials dominate the cost!

... careful treatment gives $\mathcal{O}(n^{2d})$ complexity.

Evaluation of Element Stiffness Matrix

$$\mathbf{S}^T_{\alpha\beta} = \int_T \operatorname{grad} B^n_\beta(\mathbf{x}) \cdot \mathbf{A}(\mathbf{x}) \cdot \operatorname{grad} B^n_\alpha(\mathbf{x})\, \mathrm{d}\mathbf{x}, \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{I}^n_d$$

Evaluation of Element Stiffness Matrix

$$\mathbf{S}^{T}_{\alpha\beta} = \int_{T} \operatorname{grad} B^{n}_{\beta}(\mathbf{x}) \cdot \mathbf{A}(\mathbf{x}) \cdot \operatorname{grad} B^{n}_{\alpha}(\mathbf{x})\, \mathrm{d}\mathbf{x}, \quad \alpha, \beta \in \mathcal{I}^{n}_{d}$$

Another useful property of Bernstein polys

$$\operatorname{grad} B^{n}_{\alpha}(\mathbf{x}) = n \sum_{k=1}^{d+1} B^{n-1}_{\alpha-\mathbf{e}_k}(\mathbf{x}) \operatorname{grad} \lambda_k, \quad \alpha \in \mathcal{I}^{n}_{d}$$

Evaluation of Element Stiffness Matrix

$$\mathbf{S}^T_{\alpha\beta} = \int_T \operatorname{grad} B^n_\beta(\mathbf{x}) \cdot \mathbf{A}(\mathbf{x}) \cdot \operatorname{grad} B^n_\alpha(\mathbf{x})\, \mathrm{d}\mathbf{x}, \quad \alpha, \beta \in \mathcal{I}^n_d$$

Another useful property of Bernstein polys

$$\operatorname{grad} B^n_\alpha(\mathbf{x}) = n \sum_{k=1}^{d+1} B^{n-1}_{\alpha-\mathbf{e}_k}(\mathbf{x}) \operatorname{grad} \lambda_k, \quad \alpha \in \mathcal{I}^n_d$$

Hence

$$\mathbf{S}^T_{\alpha\beta} = n^2 \sum_{k,\ell=1}^{d+1} \frac{\binom{\alpha-\mathbf{e}_k+\beta-\mathbf{e}_\ell}{\alpha-\mathbf{e}_k}}{\binom{2n-2}{n-1}} \operatorname{grad} \lambda_k \cdot \mu^{2n-2}_{\alpha-\mathbf{e}_k+\beta-\mathbf{e}_\ell}(\mathbf{A}) \cdot \operatorname{grad} \lambda_\ell$$

Bernstein-Bezier => Optimal Complexity

Theorem: All element matrices can be assembled in optimal complexity $\mathcal{O}(n^{2d})$ including cases where

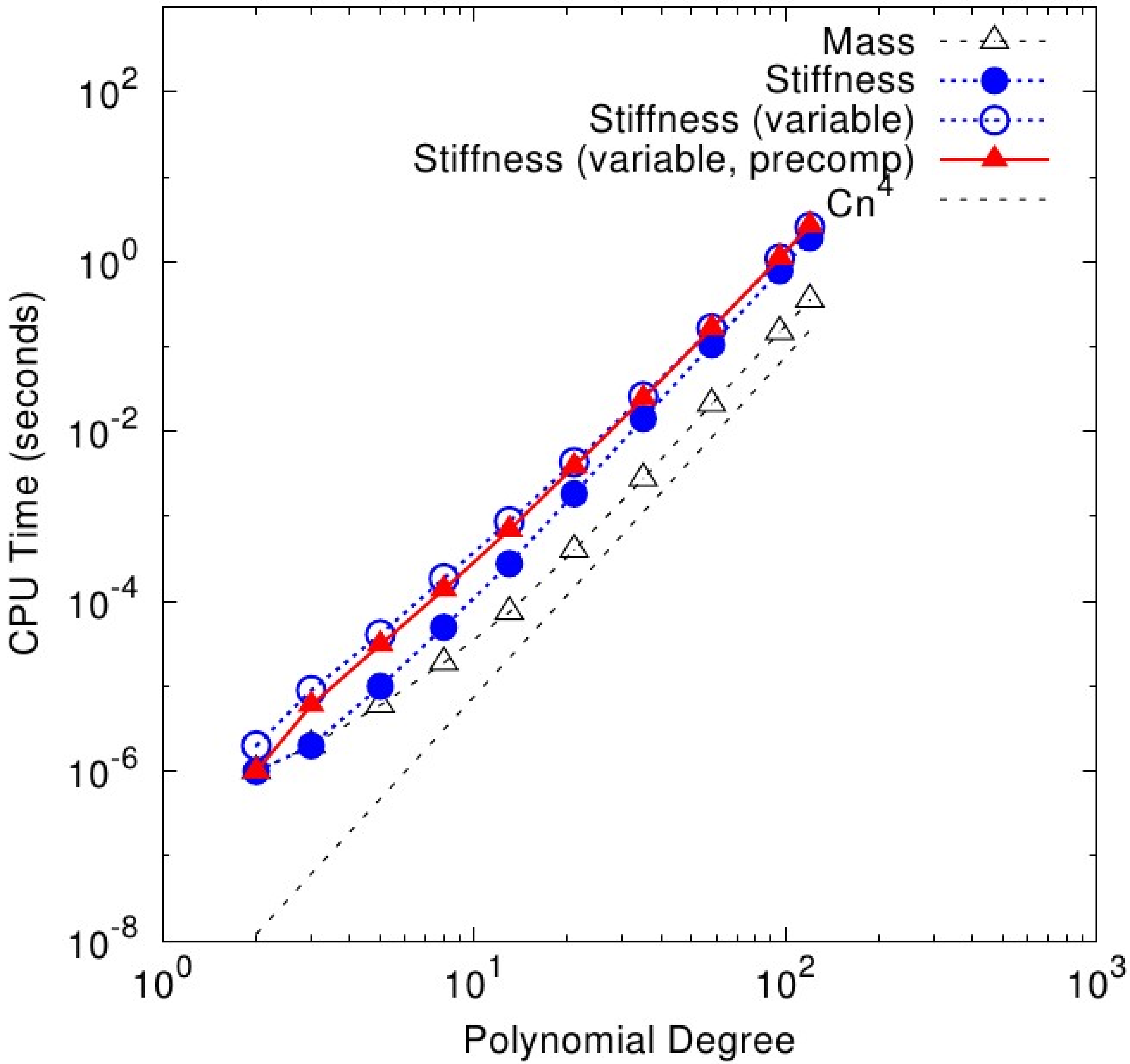
- non-constant coefficients
- non-affine elements
- coefficients depending on solution u (and/or its derivatives).

Moreover, complexity achieved even if basis functions evaluated on the fly.

Ref: Ainsworth, Andriamaro & Davydov, SISC 2011

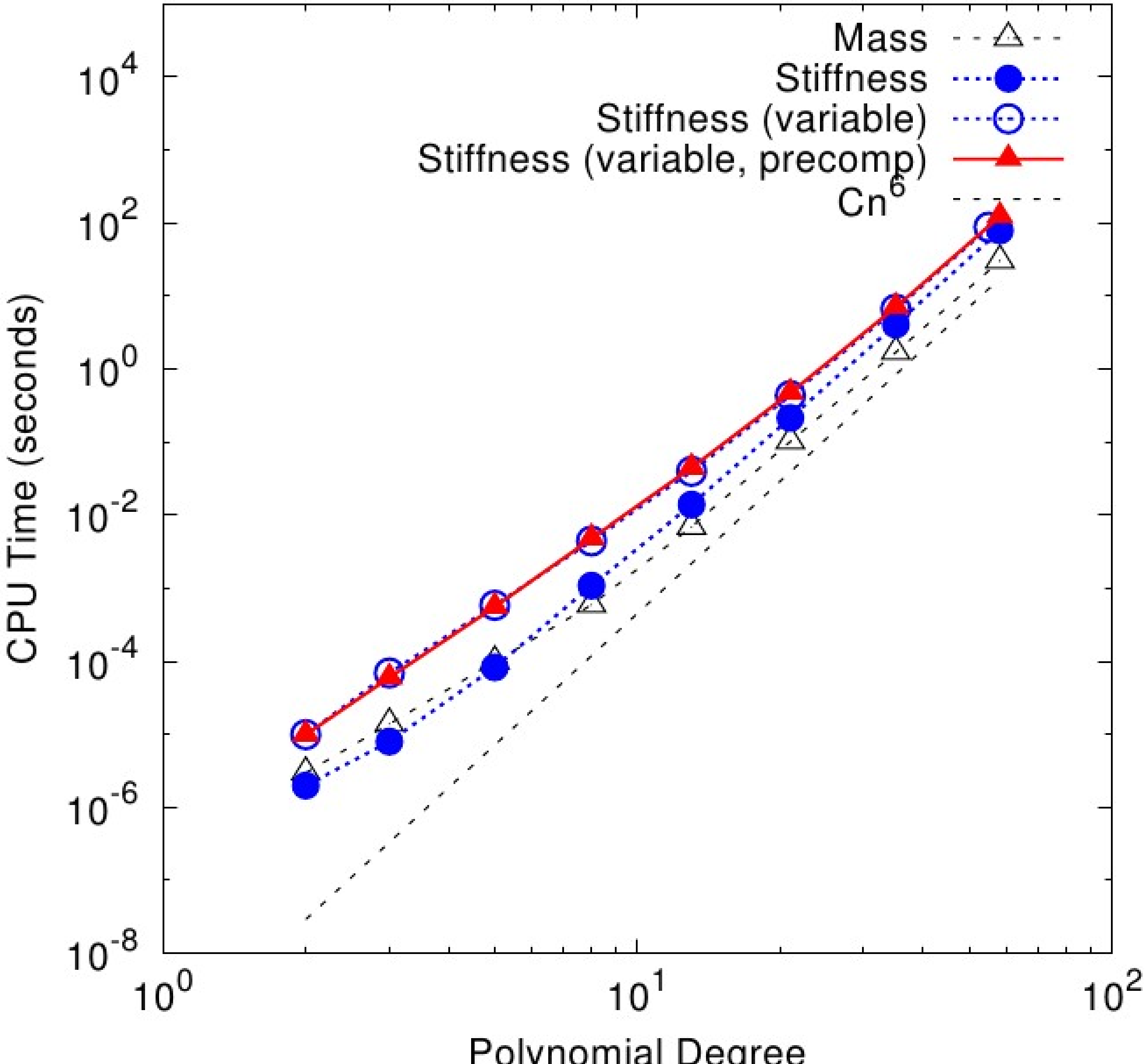
BROWN

Example: 2D



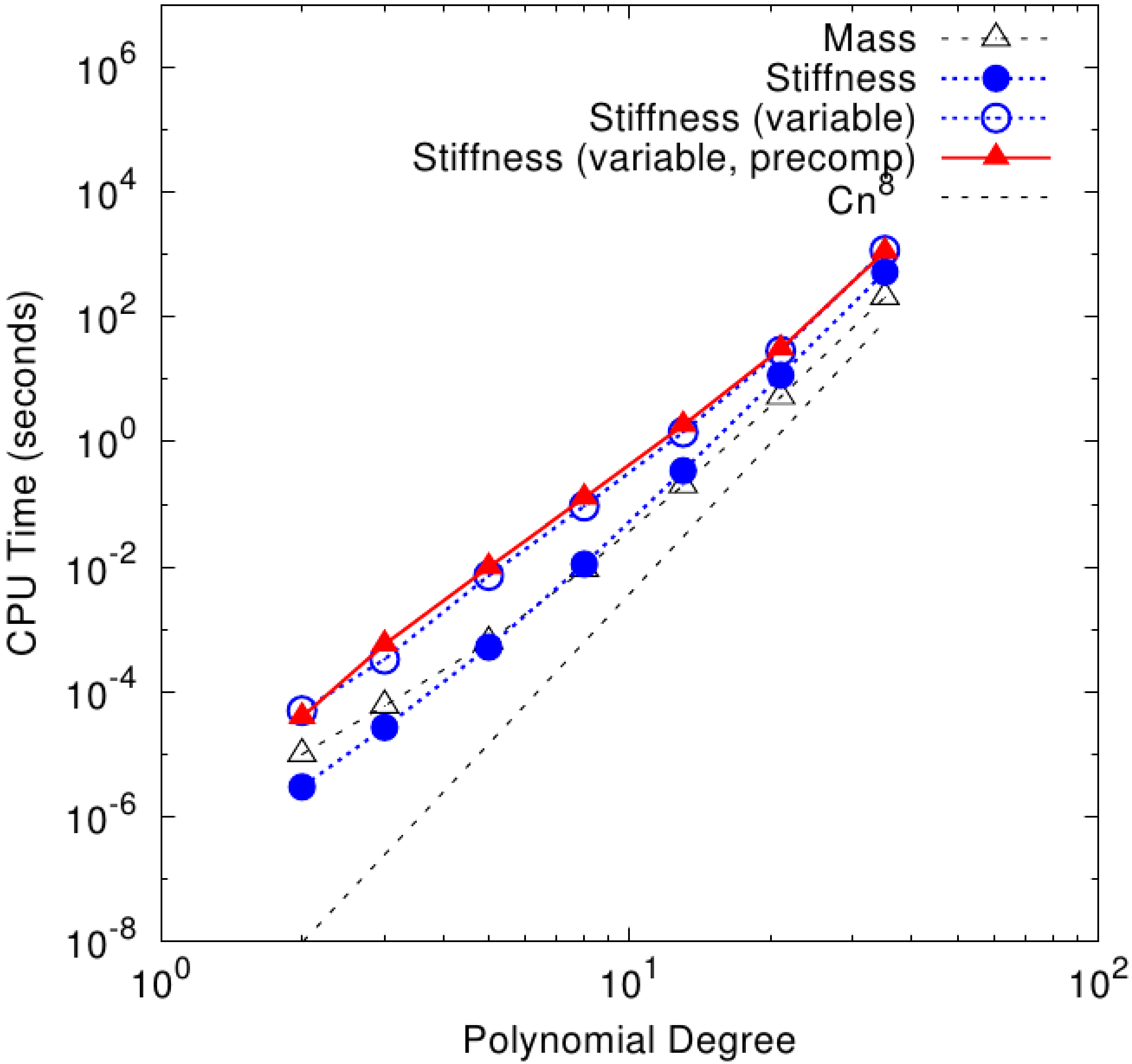
Mass
Stiffness
Stiffness (variable)
Stiffness (variable, precomp)
Cn^4
CPU Time (seconds)
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{0}
10^{1}
10^{2}
10^{3}
Polynomial Degree

Example: 3D



Mass
Stiffness
Stiffness (variable)
Stiffness (variable, precomp)
Cn^6
CPU Time (seconds)
Polynomial Degree
10^4
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^0
10^1
10^2

Example: 4D



Mass
Stiffness
Stiffness (variable)
Stiffness (variable, precomp)
Cn^8
CPU Time (seconds)
Polynomial Degree
10^6
10^4
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^0
10^1
10^2

Bernstein-Bezier Basis for Raviart-Thomas Elements

BROWN

Raviart-Thomas Elements

Raviart-Thomas finite element $\mathbb{RT}_n$ of order $n \in \mathbb{Z}_+$

$$\mathbb{RT}_n = \mathbb{P}_n^2 + \boldsymbol{x}\mathbb{P}_n$$

dimension $(n+1)(n+3)$.

e.g. $V_\omega = \text{span}\{\boldsymbol{\omega}_k : k = 1,2,3\}$ coincides with $\mathbb{RT}_0$.

$$\boldsymbol{\omega}_k = \frac{1}{2|T|}(\boldsymbol{x} - \boldsymbol{x}_k), \quad k = 1,2,3.$$

BROWN

Raviart-Thomas (n=0)

"Business as usual"

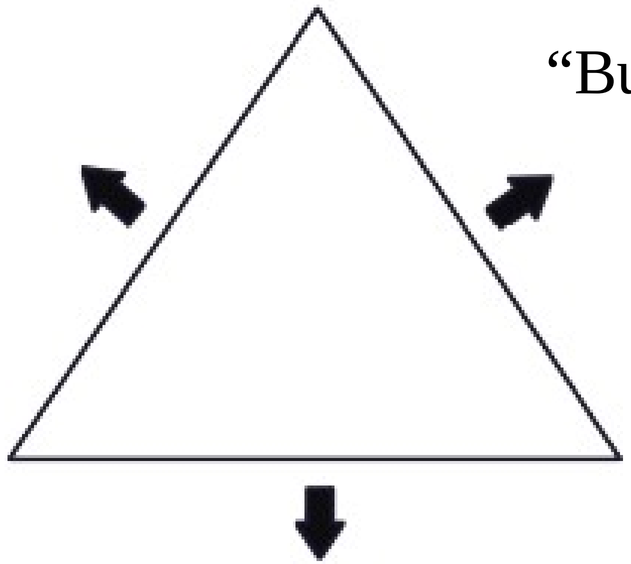


FIGURE 1. Basis functions for $\mathbb{RT}_0$ with Whitney functions indicated by arrows.

Equivalent Expression for Basis

Standard RT0 Basis

$$\boldsymbol{\omega}_k = \frac{1}{2|T|}(\boldsymbol{x} - \boldsymbol{x}_k)$$

Identical

Whitney functions

$$\boldsymbol{\omega}_1 = \lambda_2 \,\mathbf{curl}\, \lambda_3 - \lambda_3 \,\mathbf{curl}\, \lambda_2$$

$$= \begin{vmatrix} 1 & 0 & 0 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \mathbf{curl}\, \lambda_1 & \mathbf{curl}\, \lambda_2 & \mathbf{curl}\, \lambda_3 \end{vmatrix}$$

BROWN

'Generalised' Whitney Functions

By analogy with

$$\boldsymbol{\omega}_1 = \begin{vmatrix} 1 & 0 & 0 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \mathbf{curl}\,\lambda_1 & \mathbf{curl}\,\lambda_2 & \mathbf{curl}\,\lambda_3 \end{vmatrix}$$

For $\boldsymbol{\alpha} \in \mathcal{I}_n$ define

$$\boxed{\boldsymbol{\Upsilon}^n_{\boldsymbol{\alpha}} = (n+1) B^n_{\boldsymbol{\alpha}} \begin{vmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \mathbf{curl}\,\lambda_1 & \mathbf{curl}\,\lambda_2 & \mathbf{curl}\,\lambda_3 \end{vmatrix}}$$

Bernstein-Bezier Basis for $\mathbb{RT}_n$

Dispense with all *three* vertex functions:

$$V^n_{\mathbf{curl}} = \operatorname{span}\left\{\mathbf{curl}\, B^{n+1}_{\boldsymbol{\alpha}} : \ \boldsymbol{\alpha} \in \check{\mathcal{I}}_{n+1}\right\}$$

where

$$\check{\mathcal{I}}_n = \mathcal{I}_n - \{(n,0,0),(0,n,0),(0,0,n)\}$$

That leaves us two short ...

BROWN

Bernstein-Bezier Basis for $\mathbb{RT}_n$

Dispense with *one* 'generalised' Whitney function:

$$V^n_{\boldsymbol{\Upsilon}} = \operatorname{span}\{\boldsymbol{\Upsilon}^n_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{I}'_n\}$$

where

$\mathcal{I}'_n = \mathcal{I}_n$ *minus any one index*

We're three short now ...

BROWN

Bernstein-Bezier Basis for $\mathbb{RT}_n$

Include all *three* lowest order Whitney functions.

Theorem 3.6. *The set*

$$\{\boldsymbol{\Upsilon}^n_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{I}'_n\} \cup \left\{\mathbf{curl}\, B^{n+1}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \check{\mathcal{I}}_{n+1}\right\} \cup \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3\}$$

forms a basis for $\mathbb{RT}_n$.

What does it mean geometrically?

BROWN

Raviart-Thomas (n=1)

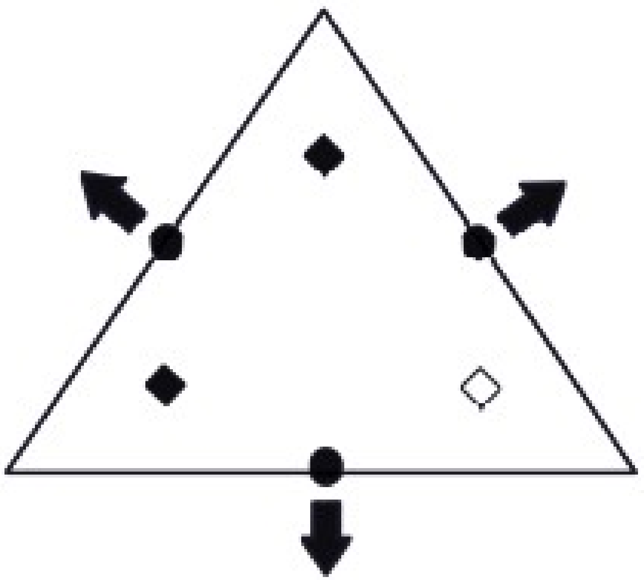
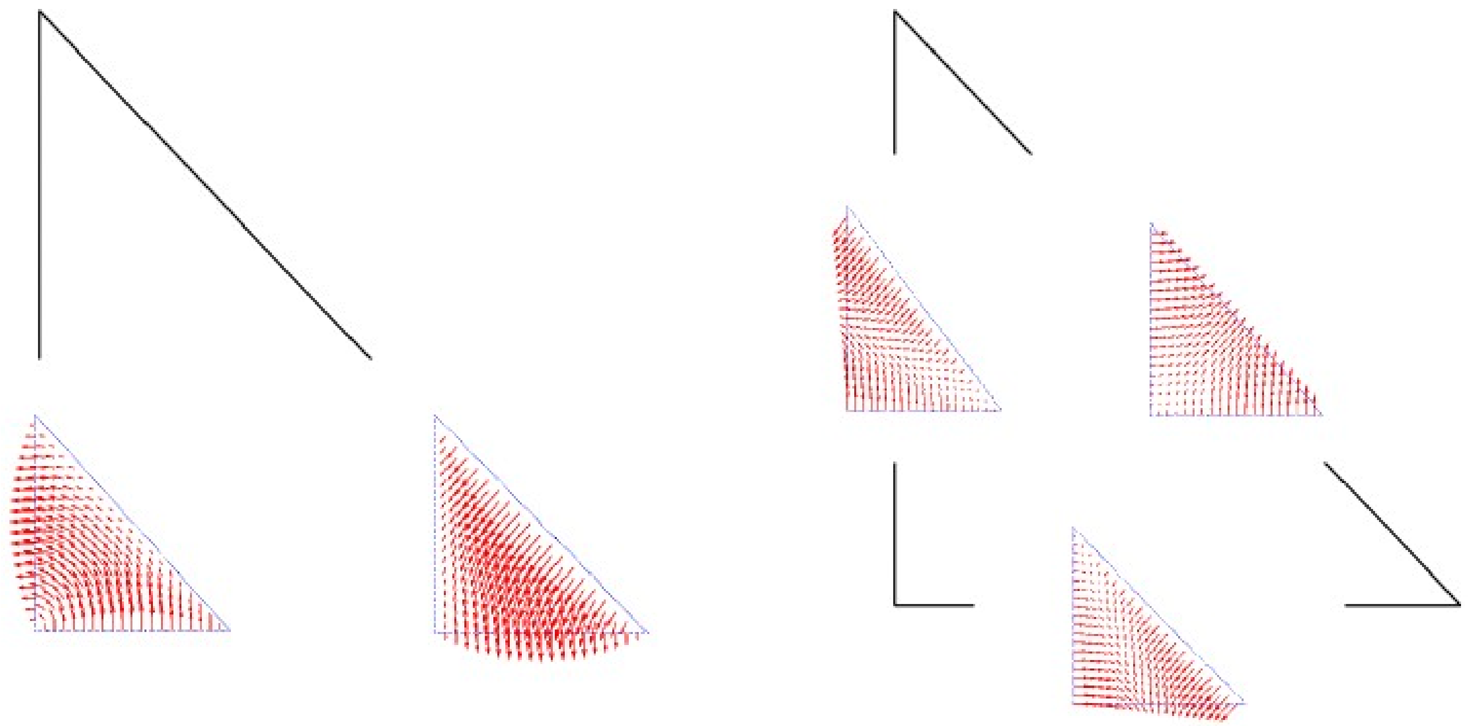


FIGURE 2. Basis functions for $\mathbb{RT}_1$: $\bullet$ denotes functions belonging to $V^1_{\mathbf{curl}}$; $\blacklozenge$ correspond to functions belonging to $V^1_{\boldsymbol{\Upsilon}}$; $\lozenge$ denotes the (arbitrarily chosen) index omitted from the set $\mathcal{I}_1$ to obtain $\mathcal{I}'_1$.

Raviart-Thomas (n=1)



Raviart-Thomas (n=2)

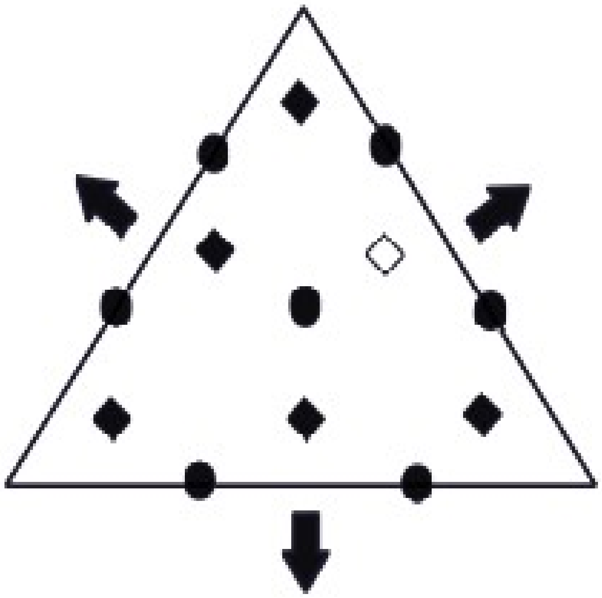
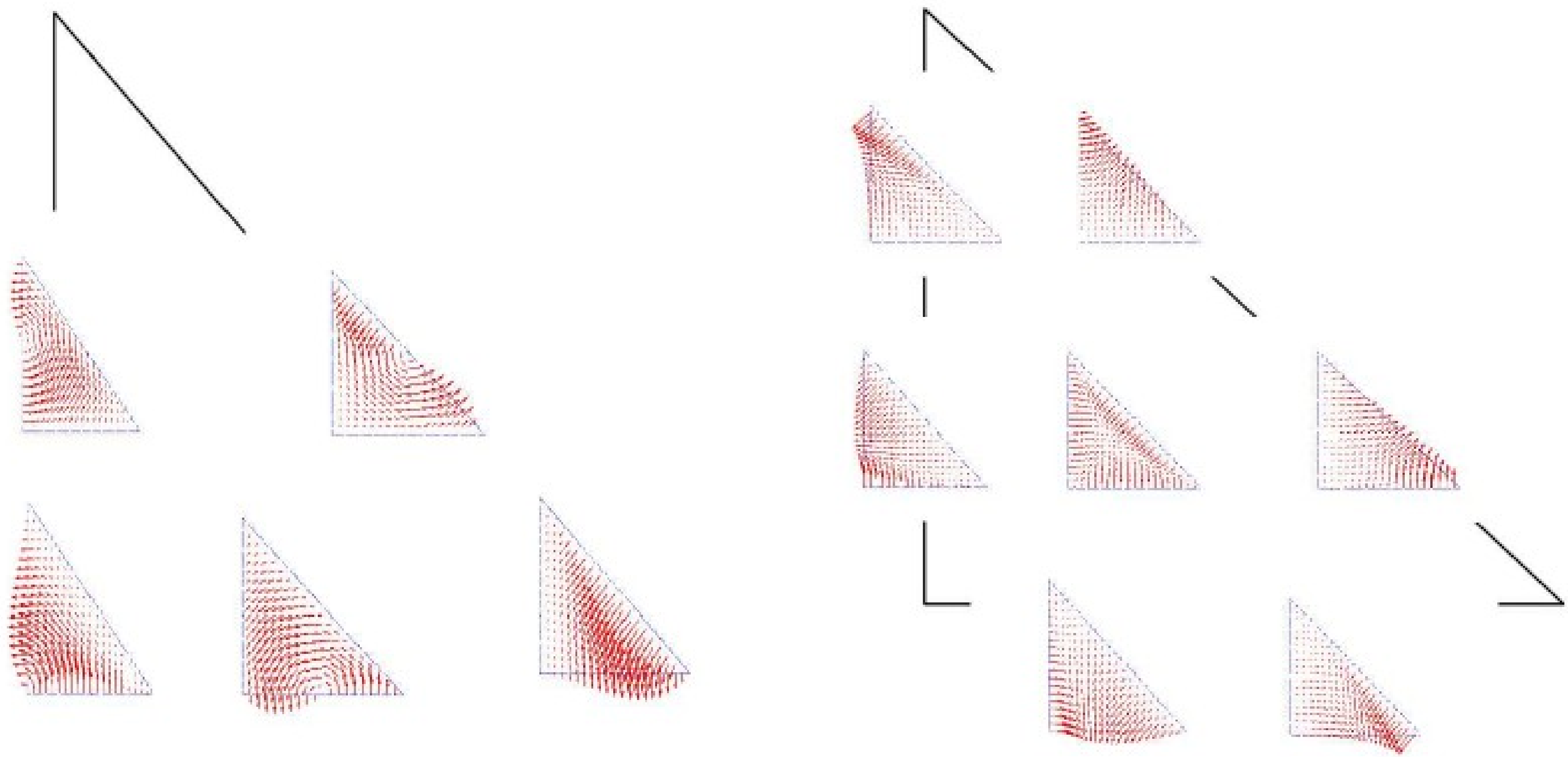


FIGURE 3. Basis functions for $\mathbb{RT}_2$: $\bullet$ denotes functions belonging to $V^2_{\mathbf{curl}}$; $\blacklozenge$ corresponds to functions belonging to $V^2_{\boldsymbol{\Upsilon}}$; $\lozenge$ denotes the (arbitrarily) chosen function omitted from the set $\mathcal{I}_2$ to obtain $\mathcal{I}'_2$.

BROWN

Raviart-Thomas (n=2)



BROWN

Raviart-Thomas (n=3)

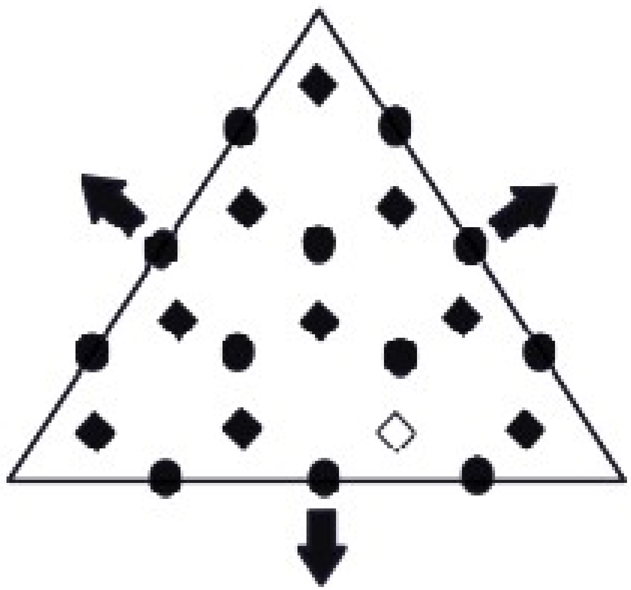


FIGURE 4. Basis functions for $\mathbb{RT}_3$: $\bullet$ denotes functions belonging to $V^3_{\mathbf{curl}}$; $\blacklozenge$ corresponds to functions belonging to $V^3_{\mathbf{\Upsilon}}$; $\lozenge$ denotes the (arbitrarily) chosen function omitted from the set $\mathcal{I}_3$ to obtain $\mathcal{I}'_3$.

Raviart-Thomas (General Case)

Order n	*Internal Dofs* V^n_{Υ}	$V^n_{\mathbf{curl}}$	*Edge Dofs* V_ω	$V^n_{\mathbf{curl}}$	*Total*
0	-	-	3	-	3
1	2	-	3	3	8
2	5	1	3	6	15
3	9	3	3	9	24
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$n(n+3)/2$	$n(n-1)/2$	3	$3n$	$(n+1)(n+3)$

BROWN

Application: Darcy Flow

$(\boldsymbol{u}, p) \in \boldsymbol{H}(\mathrm{div}; \Omega) \times L^2_0(\Omega)$ such that

$$\boldsymbol{n} \cdot \boldsymbol{u} = \psi \text{ on } \partial\Omega \text{ and}$$

$$\begin{aligned} \frac{\nu}{\kappa}(\boldsymbol{u}, \boldsymbol{v}) - (p, \mathrm{div}\, \boldsymbol{v}) &= -\varrho(\boldsymbol{g}, \boldsymbol{v}) \\ (\mathrm{div}\, \boldsymbol{u}, w) &= (f, w) \end{aligned}$$

for all $(\boldsymbol{v}, w) \in \boldsymbol{H}_0(\mathrm{div}; \Omega) \times L^2_0(\Omega)$

BROWN

Darcy: Element Residuals

Current Approximation:

$$(\boldsymbol{u}^\ell, p^\ell)$$

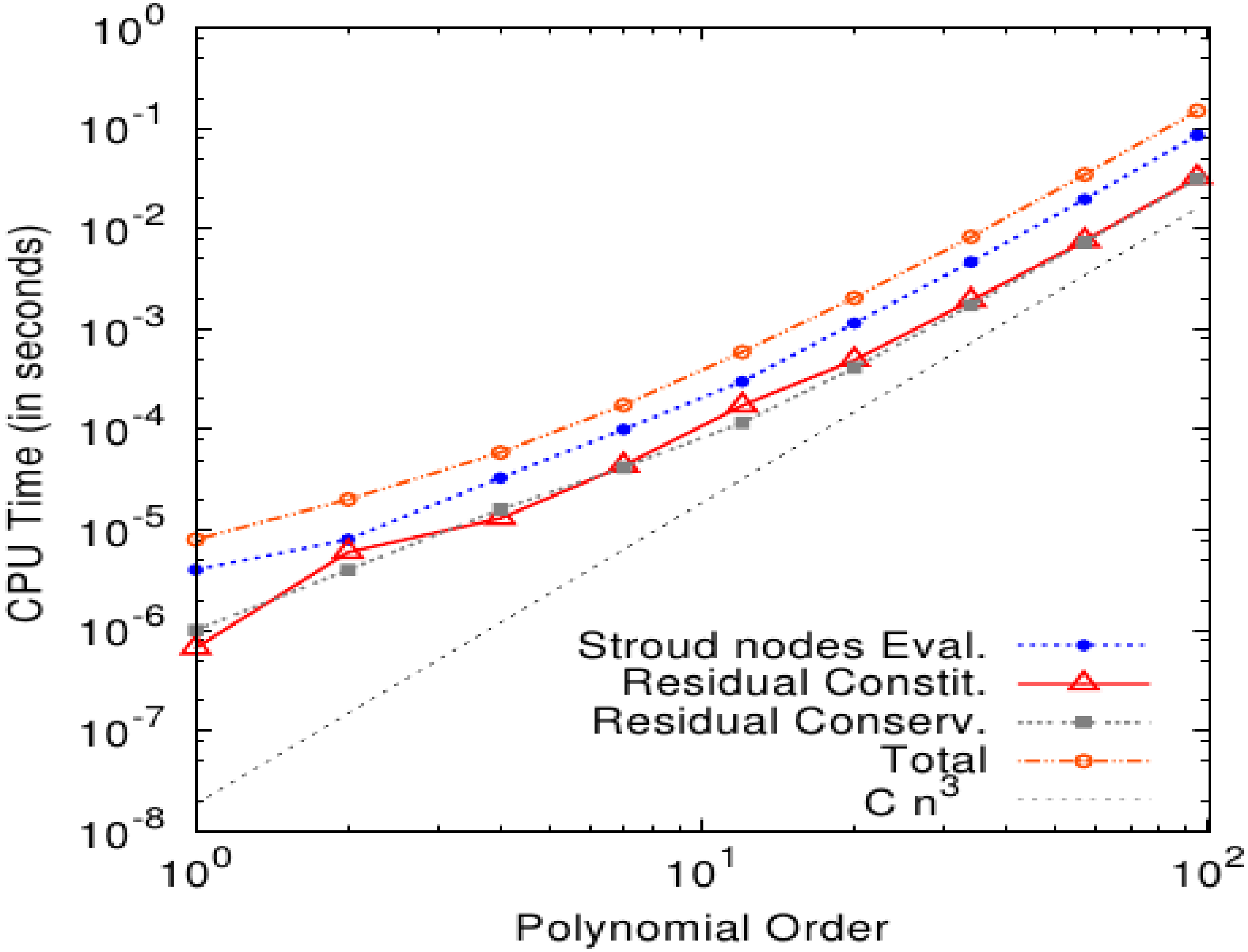
Residuals:

$$\begin{aligned} \boldsymbol{v} &\mapsto -\varrho(\boldsymbol{g}, \boldsymbol{v})_T - \frac{\nu}{\kappa}\left(\boldsymbol{u}^\ell, \boldsymbol{v}\right)_T + (p^\ell, \operatorname{div} \boldsymbol{v})_T \\ w &\mapsto (f, w)_T - (\operatorname{div} \boldsymbol{u}^\ell, w)_T \end{aligned}$$

$\boldsymbol{v}$ is chosen to be $\boldsymbol{\Upsilon}^n_{\boldsymbol{\alpha}}$, $\mathbf{curl}\, B^{n+1}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\omega}_k$, and $w = B^n_{\boldsymbol{\alpha}}$.

BROWN

CPU for Element Residuals



BROWN

Application: Maxwell's Equations

find $\mathbf{u} \in H_0(\mathrm{curl})$ and $\lambda \in \mathbb{R}$ such that

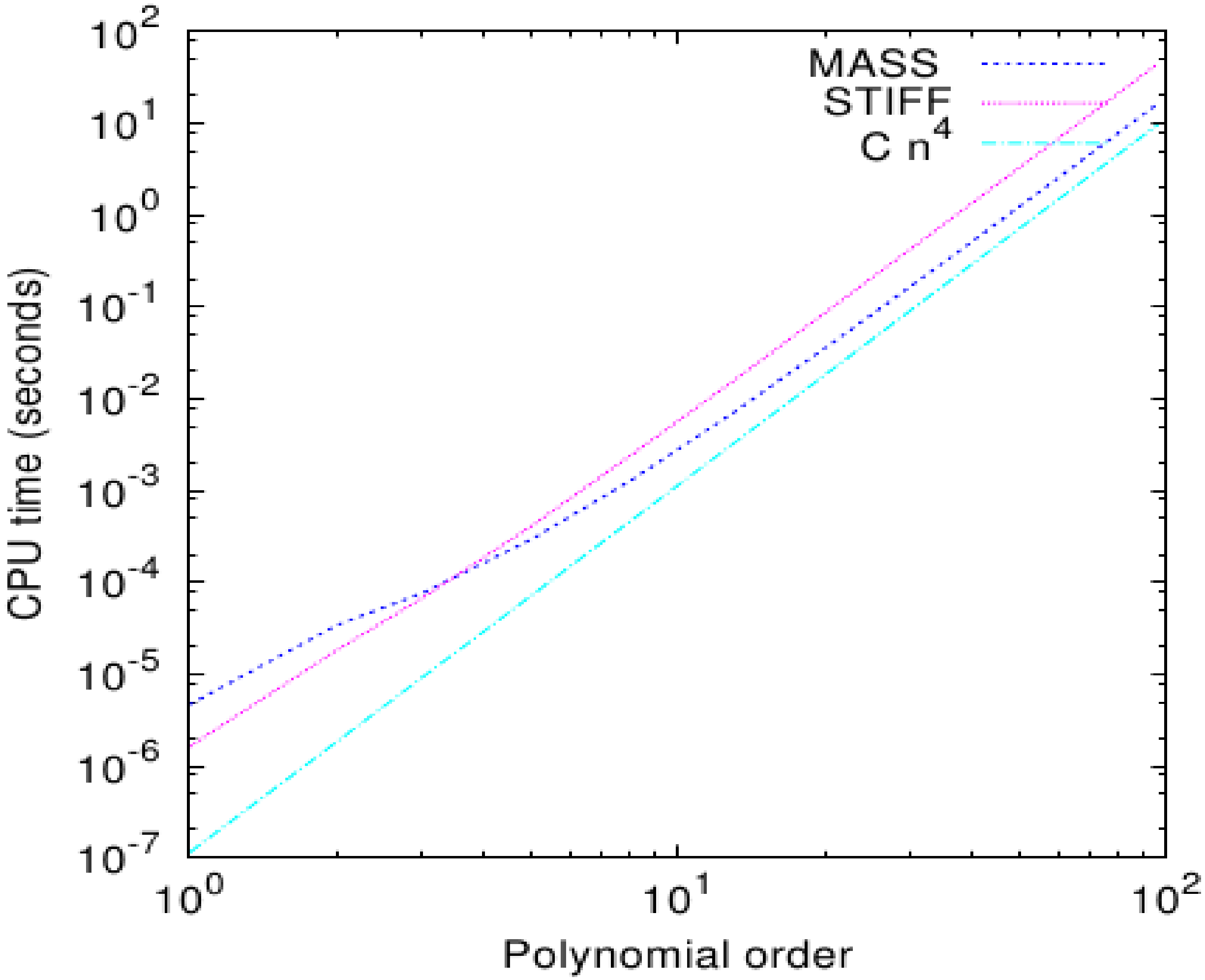
$$(\mathrm{curl}\,\mathbf{u}, \mathrm{curl}\,\mathbf{v}) = \lambda^2(\mathbf{u}, \mathbf{v}), \quad \mathbf{v} \in H_0(\mathrm{curl}).$$

$\Omega = [0, 1]^2$

True Solution: $\lambda_{k,\ell}^2 := \pi^2(k^2 + \ell^2), \quad k, \ell \in \mathbb{Z}_+.$

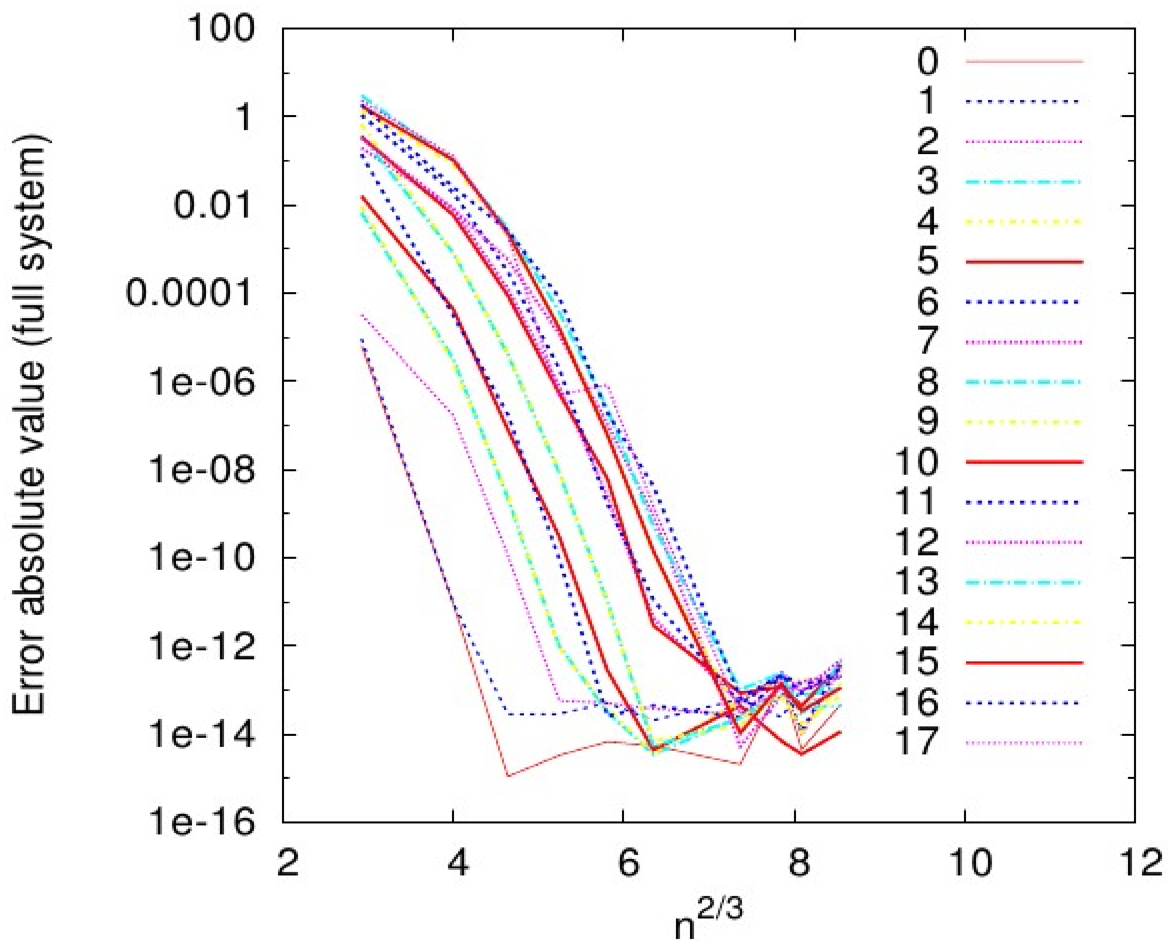
$$\mathbf{u}_{k,\ell}(x, y) := \begin{pmatrix} k\sin(k\pi x)\cos(\ell\pi x) \\ \ell\sin(\ell\pi y)\cos(k\pi x) \end{pmatrix}^{\perp}$$

Application: Maxwell's Equations



BROWN

Application: Maxwell's Equations



Summary

- *Conceptual simplicity:*

  Basis Functions <-> Nodes

- *Optimal complexity* computation of element matrices using AAD Algorithm/fast matrix multiply (MA, Andriamaro & Davydov, SISC, 2011)
- *Extension to H(div)/H(curl)* MA, Andriamaro & Davydov, Brown Tech. Rep. 20, 2012)
- *Non-uniform Local Polynomial Orders* with de Casteljau 'pyramid' algorithms for *entire* FE implementation (Ainsworth, SISC 36, 2014).