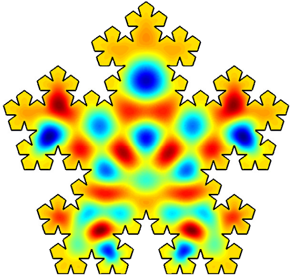
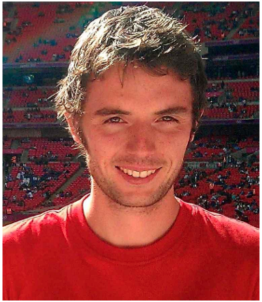
The ultraspherical spectral element method

Dan Fortunato

Alex Townsend



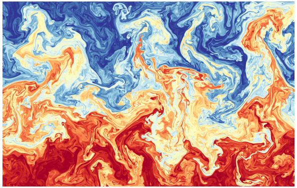


Nick Hale

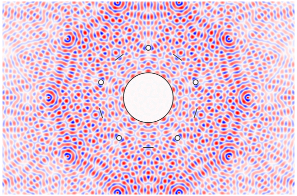
Introduction
Global spectral methods

- ✓ Spectrally accurate convergence to solution (e.g. exponential)
- ✓ High accuracy
- ✓ Low numerical dissipation and dispersion

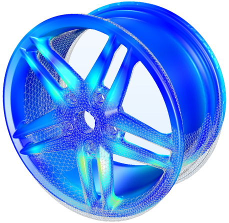
High Reynolds number flows
[Dedalus Project, 2019]

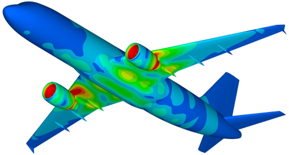


High frequency scattering
[Slevinsky & Olver, 2017]



- ✗ Lack geometric flexibility
- ✗ Globalize corner singularities

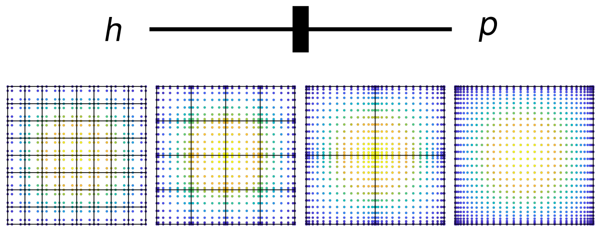




Spectral element methods and *hp*-adaptivity
Theory vs. practice

SEMs combine:

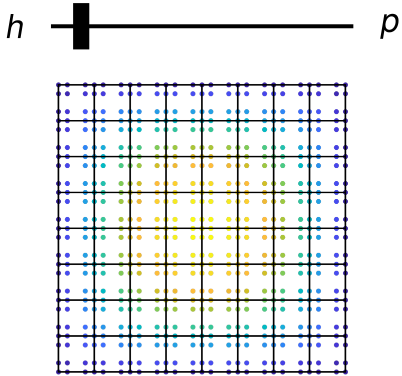
- the flexibility of finite element methods
- the convergence properties of global spectral methods



Spectral element methods and *hp*-adaptivity
Theory vs. practice

SEMs combine:

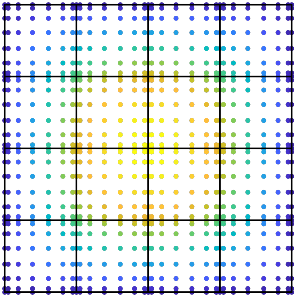
- the flexibility of finite element methods
- the convergence properties of global spectral methods



Spectral element methods and *hp*-adaptivity
Theory vs. practice

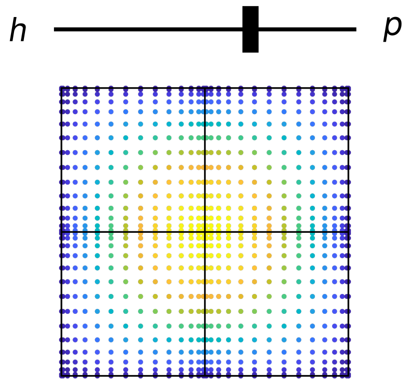
SEMs combine:

- the flexibility of finite element methods
- the convergence properties of global spectral methods



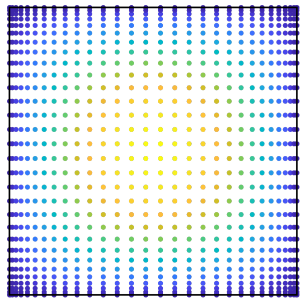
SEMs combine:

- the flexibility of finite element methods
- the convergence properties of global spectral methods



SEMs combine:

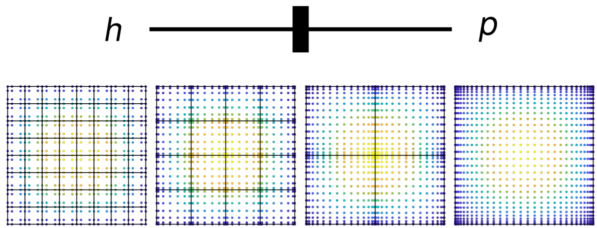
- the flexibility of finite element methods
- the convergence properties of global spectral methods



Spectral element methods and *hp*-adaptivity
Theory vs. practice

SEMs combine:

- the flexibility of finite element methods
- the convergence properties of global spectral methods



Most SEMs cost $\mathcal{O}(p^6/h^2) = \mathcal{O}(N p^4)$, so the slider is biased.

"In practice, hp-adaptivity means $p \lesssim 6$." [Sherwin, 2014]

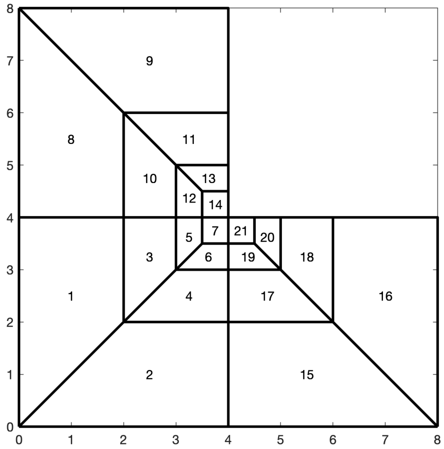
Spectral element methods and *hp*-adaptivity
Theory vs. practice

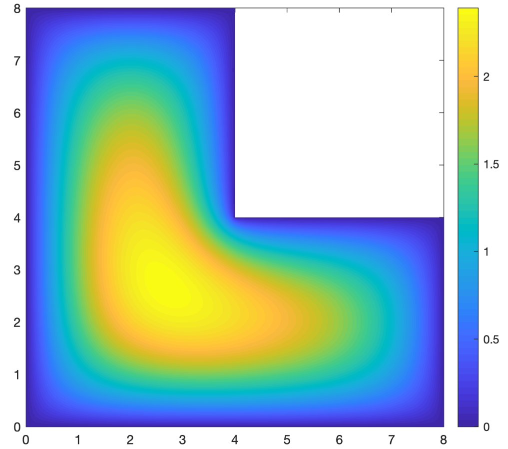
"As expected, the numerical results indicate that in the case of smooth solutions, one should fix the mesh and vary the polynomial order according to the desired accuracy (p-convergence)." [Sherwin, 2014]

"While flow discontinuities are understandably better resolved with h-refinement, it is found that in regions of smooth flow, p-refinement offers a higher accuracy with the same number of degrees of freedom." [Li & Jameson, 2010]

"Within each of these elements the solution is represented by Nth-order polynomials, where $N = 5$–15 is most common but $N = 1$–100 or beyond is feasible." [Fischer, 2016]

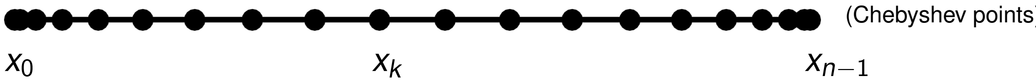
Want to choose *hp* based on physical considerations, not computational ones.



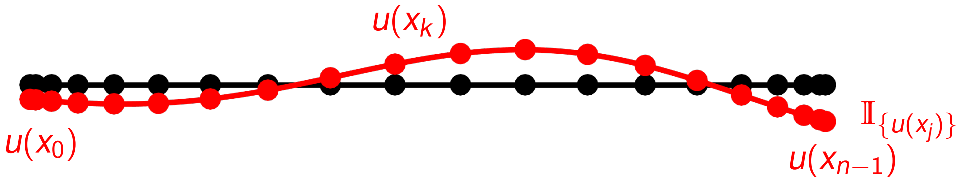


Why do spectral methods get a bad rap?
Spectral collocation

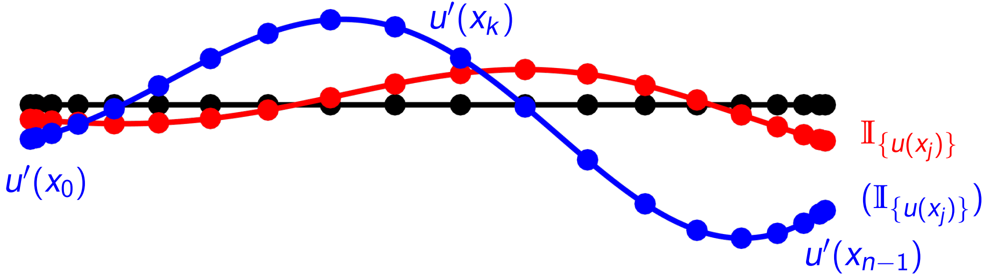
Given values on a grid, what are the values of the derivative on that same grid?



Given values on a grid, what are the values of the derivative on that same grid?



Given values on a grid, what are the values of the derivative on that same grid?



Differentiation $\{x_k\} \to \{x_k\}$ is **dense**:

$$u'(x_k) = (\mathbb{I}_{\{u(x_j)\}})'(x_k)$$

The derivative at the k-th point depends on the values of u **at all points**.

Why do spectral methods get a bad rap?

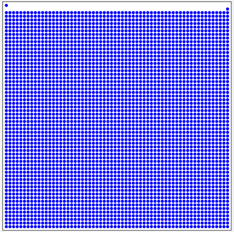
Spectral collocation

1. Dense matrices
2. Ill-conditioned matrices
3. When has it converged? Tricky.

Why do spectral methods get a bad rap?
Spectral collocation

1. Dense matrices
2. Ill-conditioned matrices
3. When has it converged? Tricky.

$$u''(x) + \cos(x)u(x) = 0$$



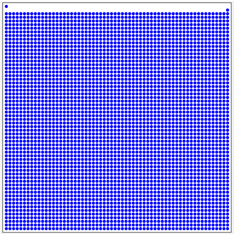
$\mathcal{O}(n^3)$

Why do spectral methods get a bad rap?

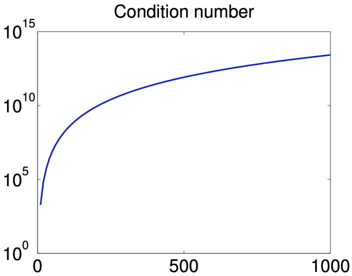
Spectral collocation

1. Dense matrices
2. Ill-conditioned matrices
3. When has it converged? Tricky.

$u''(x) + \cos(x)u(x) = 0$



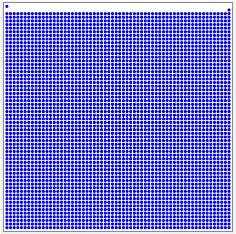
$\mathcal{O}(n^3)$



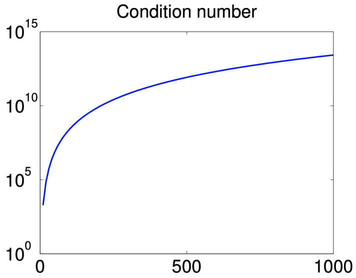
Why do spectral methods get a bad rap?
Spectral collocation

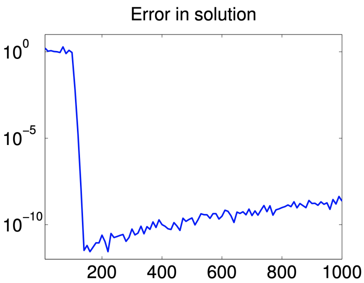
1. Dense matrices
2. Ill-conditioned matrices
3. When has it converged? Tricky.

$$u''(x) + \cos(x)u(x) = 0$$



$\mathcal{O}(n^3)$





Spectral methods can be sparse and well-conditioned

Fourier spectral method (for periodic problems)

Idea: Represent u as coefficients of a Fourier series instead of values on a grid.

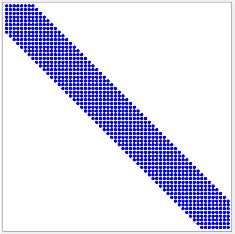
$$u(x) = \sum_{k=0}^{n-1} u_k e^{ikx}$$

Differentiation $\{e^{ikx}\} \to \{e^{ikx}\}$ is **sparse**:

$$\frac{d}{dx} e^{ikx} = ike^{ikx}$$

The classical Fourier spectral method is sparse and well-conditioned for periodic problems.

$u''(x) + \cos(x)u(x) = 0$



Spectral methods can be sparse and well-conditioned
Chebyshev tau method

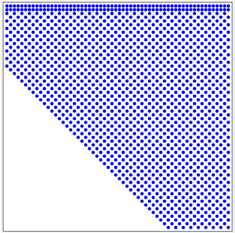
Idea: Represent u as coefficients of a Chebyshev series.

$$u(x) = \sum_{k=0}^{n-1} u_k T_k(x), \qquad T_k(x) = \cos(k \cos^{-1} x)$$

Differentiation $\{T_k(x)\} \to \{T_k(x)\}$ is **dense**:

$$T_k'(x) = \begin{cases} 2k \sum_{j \text{ odd}}^{k-1} T_j(x), & k \text{ even}, \\ 2k \sum_{j \text{ even}}^{k-1} T_j(x) - 1, & k \text{ odd}. \end{cases}$$

$u''(x) + \cos(x)u(x) = 0$



The Chebyshev tau method is dense and ill-conditioned.

Spectral methods can be sparse and well-conditioned
Ultraspherical spectral method

Idea: Represent *u* as coefficients of a Chebyshev series.

$$u(x) = \sum_{k=0}^{n-1} u_k T_k(x), \qquad T_k(x) = \cos(k \cos^{-1} x)$$

Remedy: Let differentiation convert to ultraspherical bases.

$$T_k'(x) = kC_{k-1}^{(1)}(x), \qquad T_k''(x) = 2kC_{k-2}^{(2)}(x), \qquad T_k'''(x) = 8kC_{k-3}^{(3)}(x), \quad \ldots$$

Then differentiation $\{T_k(x)\} \to \{C_k^{(\lambda)}(x)\}$ is **sparse**.

Spectral methods can be sparse and well-conditioned
Ultraspherical spectral method

Differentiation:

$$T_k'(x) = kC_{k-1}^{(1)}(x), \qquad \mathcal{D} = \begin{pmatrix} 0 & 1 & & & \\ & & 2 & & \\ & & & 3 & \\ & & & & \ddots \end{pmatrix}$$

Conversion:

$$T_k(x) = \tfrac{1}{2}\left(C_k^{(1)} - C_{k-2}^{(1)}\right), \qquad \mathcal{S} = \begin{pmatrix} 1 & 0 & -\frac{1}{2} & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & \\ & & \frac{1}{2} & 0 & \ddots \\ & & & \ddots & \ddots \end{pmatrix}$$

Multiplication:

$$a(x) \approx \sum_{k=0}^{m-1} a_k T_k(x), \quad T_j(x)T_k(x) = \tfrac{1}{2}\left(T_{|j-k|} + T_{j+k}\right), \quad m\text{-banded operation}$$

Spectral methods can be sparse and well-conditioned

Ultraspherical spectral method

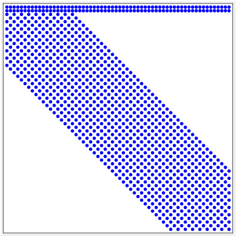
1. Almost banded matrices
2. Well-conditioned matrices
3. When has it converged? Easy.

Spectral methods can be sparse and well-conditioned

Ultraspherical spectral method

1. Almost banded matrices
2. Well-conditioned matrices
3. When has it converged? Easy.

$$u''(x) + \cos(x)u(x) = 0$$



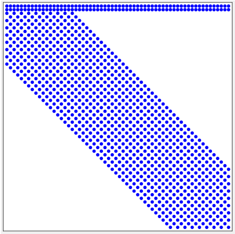
$$\mathcal{O}(nm^2)$$

Spectral methods can be sparse and well-conditioned

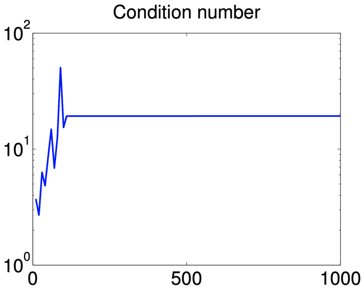
Ultraspherical spectral method

1. Almost banded matrices
2. Well-conditioned matrices
3. When has it converged? Easy.

$u''(x) + \cos(x)u(x) = 0$



$\mathcal{O}(nm^2)$

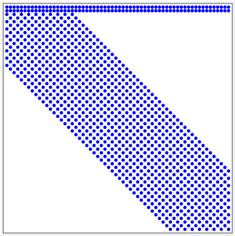


Spectral methods can be sparse and well-conditioned

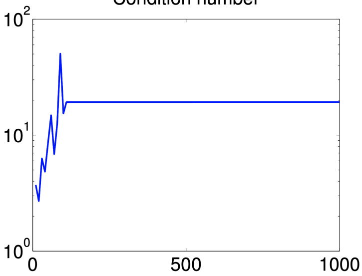
Ultraspherical spectral method

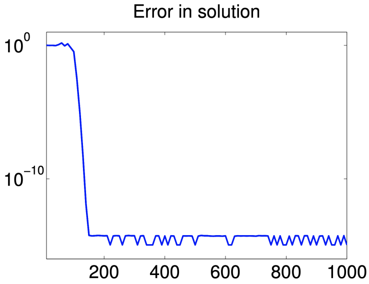
1. Almost banded matrices
2. Well-conditioned matrices
3. When has it converged? Easy.

$u''(x) + \cos(x)u(x) = 0$



$\mathcal{O}(nm^2)$

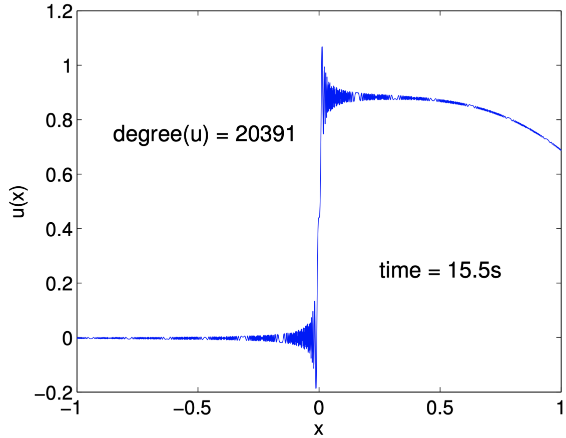




Spectral methods can be sparse and well-conditioned

Ultraspherical spectral method

$$u'(x) + x^3 u(x) = 100\sin(20000x^2), \qquad u(-1) = 0$$



The ultraspherical spectral method in 2D
Solving PDEs on rectangles

Solve the elliptic PDE

$$\begin{aligned} \mathcal{L}u(x,y) &= f(x,y) \text{ in } [-1,1]^2 \\ u(x,y) &= g(x,y) \text{ on boundary} \end{aligned}$$

where

$$\mathcal{L} = \sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{ij}(x,y) \frac{\partial^{i+j}}{\partial x^i \partial y^j}$$

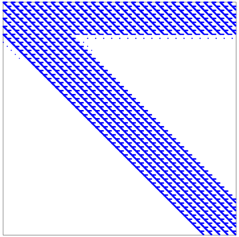
Solve the elliptic PDE

$$\begin{aligned} \mathcal{L}u(x,y) &= f(x,y) \text{ in } [-1,1]^2 \\ u(x,y) &= g(x,y) \text{ on boundary} \end{aligned}$$

where

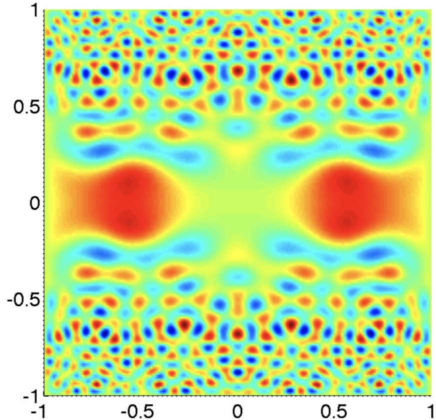
$$\mathcal{L} = \sum_{i=0}^{2}\sum_{j=0}^{2-i} a_{ij}(x,y)\frac{\partial^{i+j}}{\partial x^i \partial y^j} \approx \sum_{k=1}^{K}\left(\mathcal{L}_k^y \otimes \mathcal{L}_k^x\right)$$

1-D ultraspherical method



Almost banded-block-banded
Woodbury solve: $\mathcal{O}(n^4)$
Conditioning: $\mathcal{O}(n^3)$

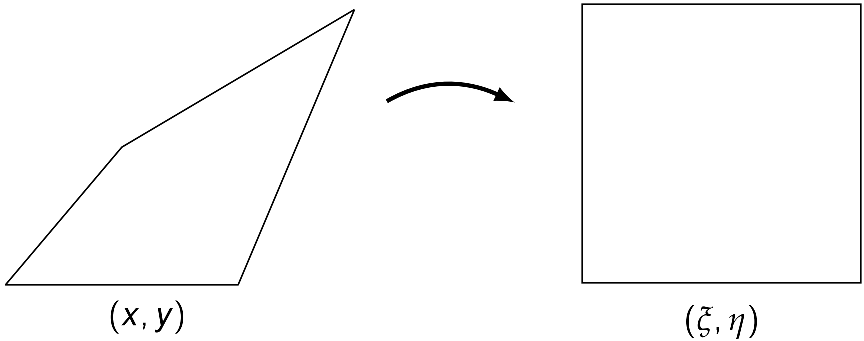
$$\nabla^2 u + 10000\cos^2 y(\tfrac{1}{2} + \sin^2 x)u = \cos xy, \qquad u(\cdot, \pm 1) = u(\pm 1, \cdot) = 1$$



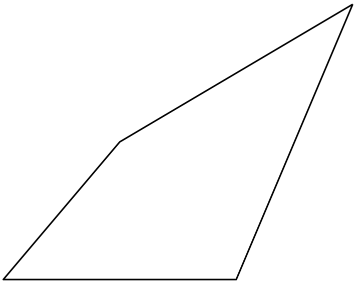
Discretization $\approx 1000 \times 1000$

The ultraspherical spectral method in 2D

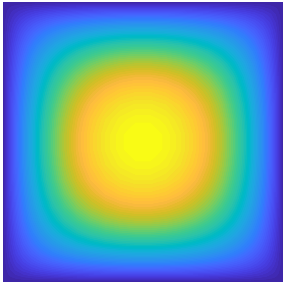
Solving PDEs on kites



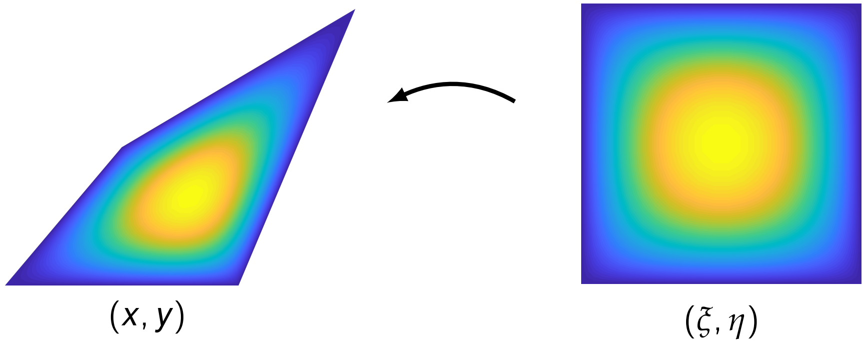
The ultraspherical spectral method in 2D
Solving PDEs on kites



(x, y)

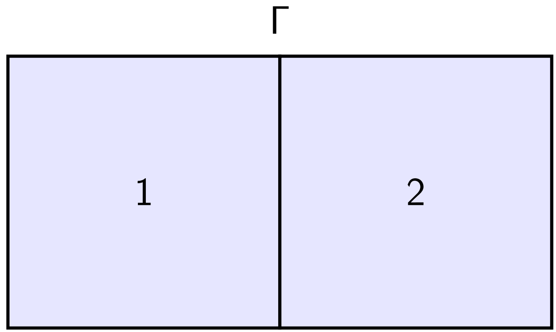


(ξ, η)

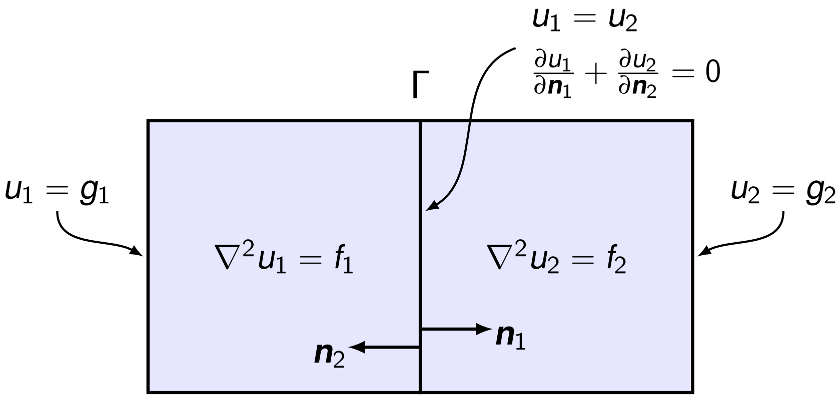

(x, y)
(ξ, η)

The ultraspherical spectral element method

Two glued squares

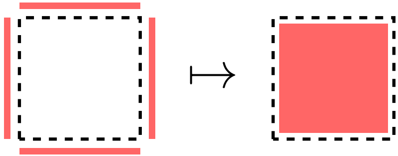


The ultraspherical spectral element method
Two glued squares

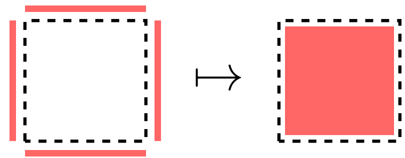


The ultraspherical spectral element method
Building blocks

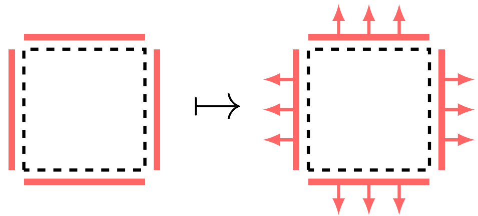
1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.

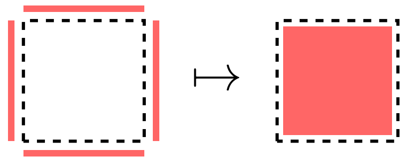


2. Dirichlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to n coefficients of the normal derivative of the solution on each side.

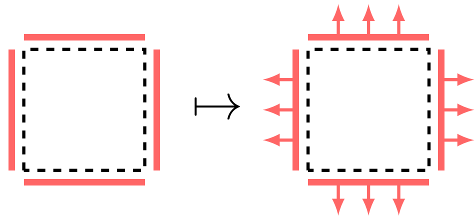


The ultraspherical spectral element method
Building blocks

1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



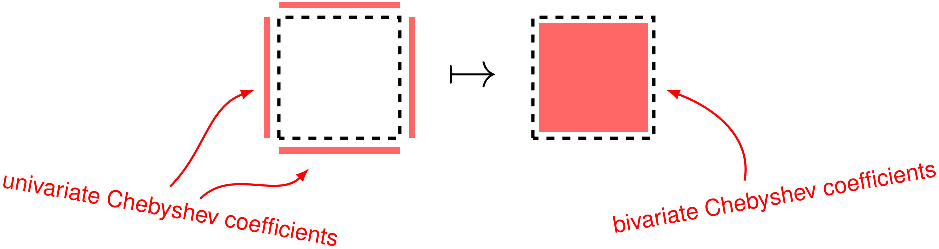
2. Dirichlet-to-Neumann map: $DtN \in \mathbb{R}^{4n \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to n coefficients of the normal derivative of the solution on each side.



$= D_{\boldsymbol{n}} S$

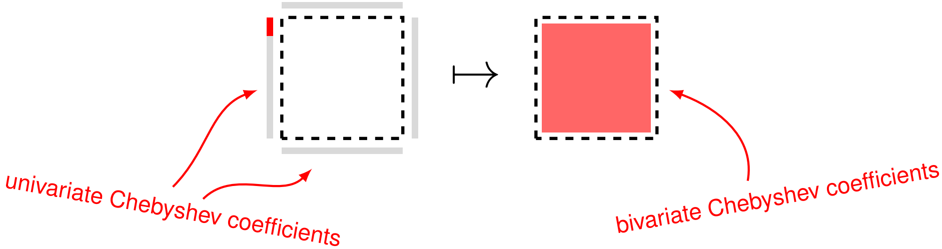
The ultraspherical spectral element method
Constructing the solution operator

1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



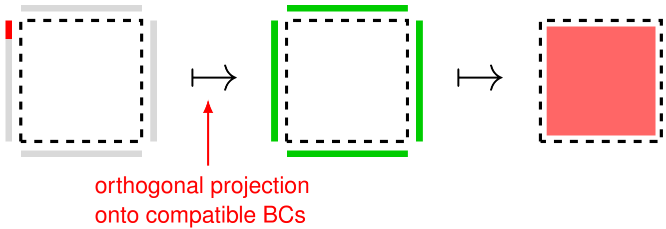
The ultraspherical spectral element method
Constructing the solution operator

1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



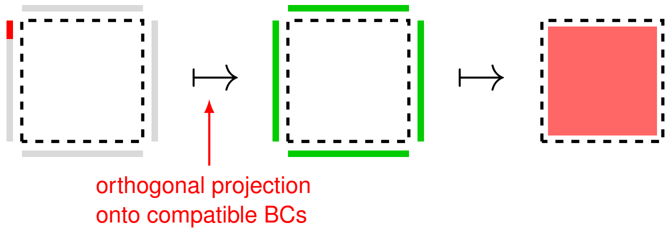
How does the n^{th} Dirichlet coefficient affect the solution?

1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.

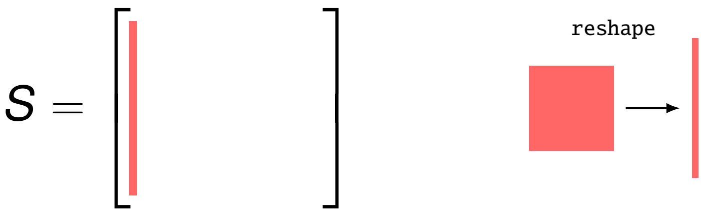


How does the n^{th} Dirichlet coefficient affect the solution?

1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.

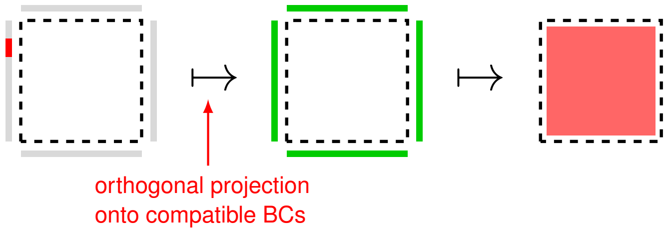


How does the n^{th} Dirichlet coefficient affect the solution?

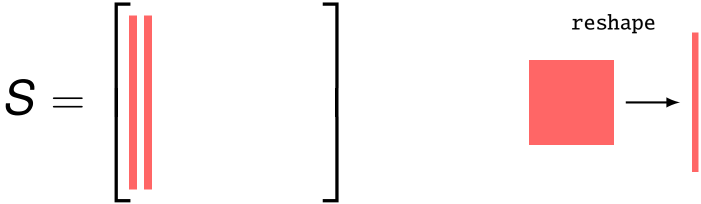


The ultraspherical spectral element method
Constructing the solution operator

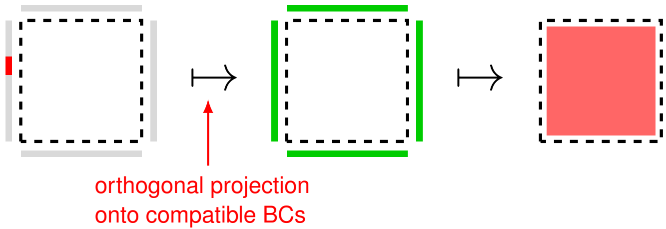
1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



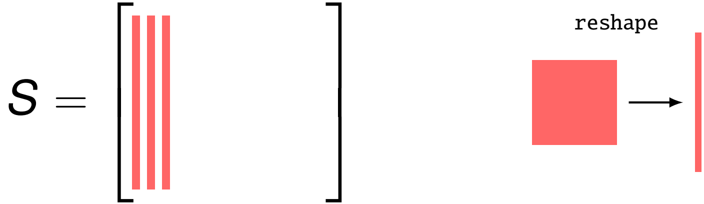
How does the n^{th} Dirichlet coefficient affect the solution?



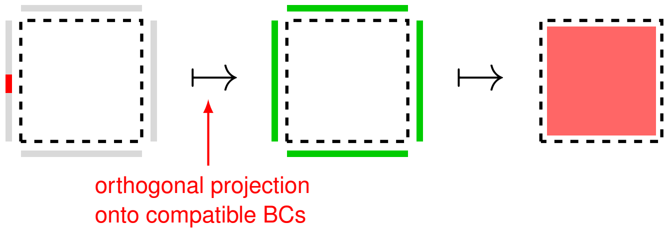
1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.



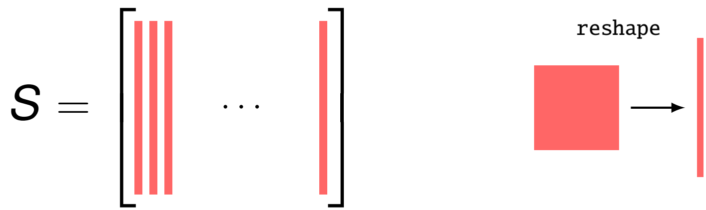
How does the n^{th} Dirichlet coefficient affect the solution?



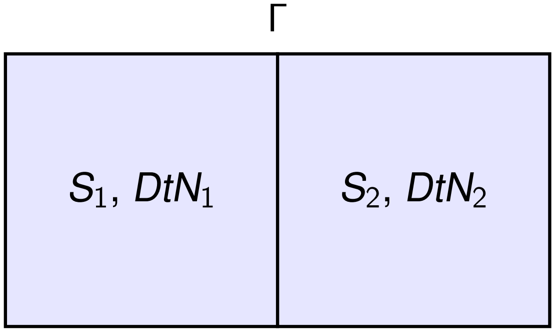
1. Solution operator: $S \in \mathbb{R}^{n^2 \times 4n}$
   - Maps n coefficients of Dirichlet data on each side to $n \times n$ coefficients of the solution.

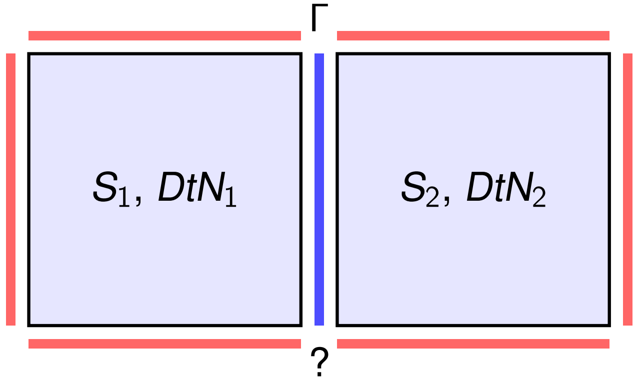


How does the n^{th} Dirichlet coefficient affect the solution?



The ultraspherical spectral element method
Merging operators

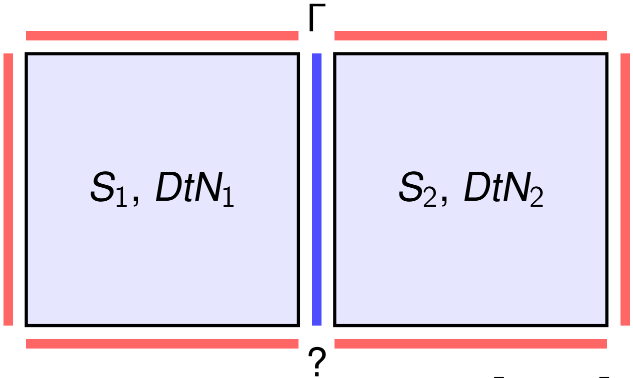




Γ
S_1, DtN_1
S_2, DtN_2
?

The ultraspherical spectral element method

Merging operators

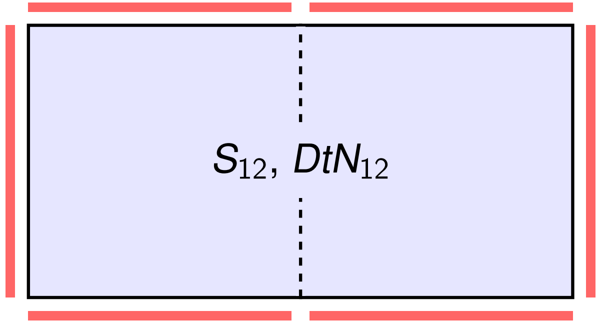


$$S_{12} = -\left(DtN_1^{\Gamma,\Gamma} + DtN_2^{\Gamma,\Gamma}\right)^{-1} \begin{bmatrix} DtN_1^{\Gamma,1} \\ DtN_2^{\Gamma,2} \end{bmatrix}$$

$$DtN_{12} = \begin{bmatrix} DtN_1^{\Gamma,1} & 0 \\ 0 & DtN_2^{\Gamma,2} \end{bmatrix} + \begin{bmatrix} DtN_1^{1,\Gamma} \\ DtN_2^{2,\Gamma} \end{bmatrix} S_{12}$$

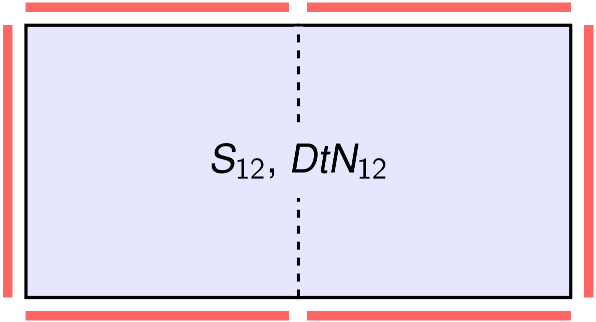
The ultraspherical spectral element method

Merging operators



$$S_{12} = -\left(DtN_1^{\Gamma,\Gamma} + DtN_2^{\Gamma,\Gamma}\right)^{-1} \begin{bmatrix} DtN_1^{\Gamma,1} \\ DtN_2^{\Gamma,2} \end{bmatrix}$$

$$DtN_{12} = \begin{bmatrix} DtN_1^{\Gamma,1} & 0 \\ 0 & DtN_2^{\Gamma,2} \end{bmatrix} + \begin{bmatrix} DtN_1^{1,\Gamma} \\ DtN_2^{2,\Gamma} \end{bmatrix} S_{12}$$



Recurse!

$$S_{12} = -\left(DtN_1^{\Gamma,\Gamma} + DtN_2^{\Gamma,\Gamma}\right)^{-1} \begin{bmatrix} DtN_1^{\Gamma,1} \\ DtN_2^{\Gamma,2} \end{bmatrix}$$

$$DtN_{12} = \begin{bmatrix} DtN_1^{\Gamma,1} & 0 \\ 0 & DtN_2^{\Gamma,2} \end{bmatrix} + \begin{bmatrix} DtN_1^{1,\Gamma} \\ DtN_2^{2,\Gamma} \end{bmatrix} S_{12}$$

The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

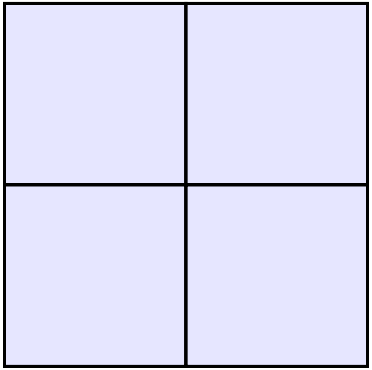
A hierarchical variant of the Schur complement method.

Gunnar Martinsson

Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.

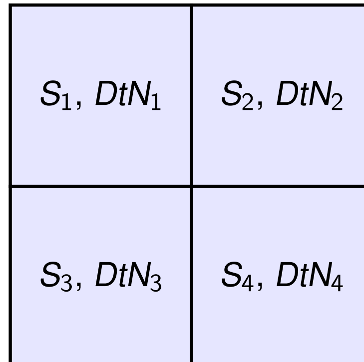
Gunnar Martinsson

Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Build element operators $\mathcal{O}(p^4)$



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.

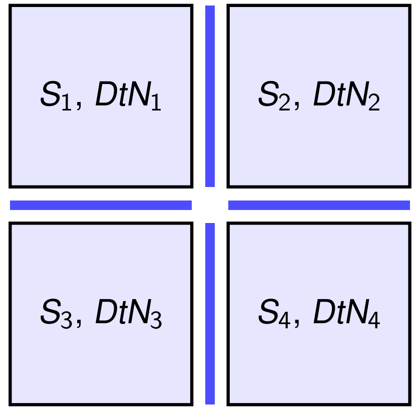

Gunnar Martinsson


Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Merge operators $\mathcal{O}(p^3)$



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.

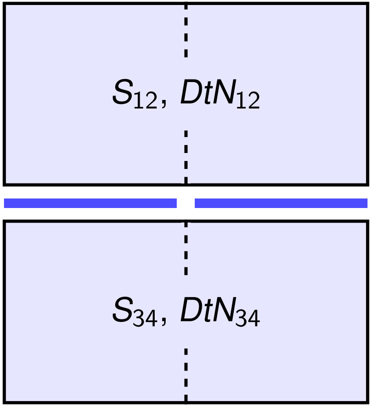
Gunnar Martinsson

Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Merge operators $\mathcal{O}(p^3)$



A hierarchical variant of the Schur complement method.

Gunnar Martinsson

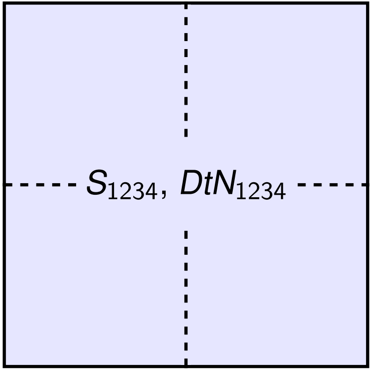
Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Merge operators $\mathcal{O}(p^3)$



A hierarchical variant of the Schur complement method.

Gunnar Martinsson

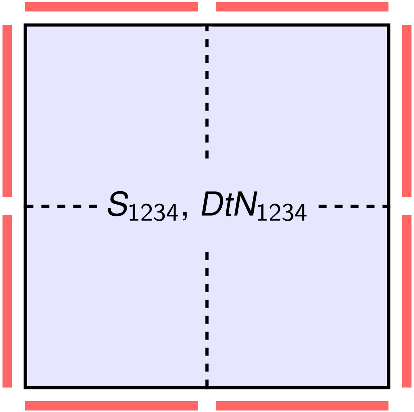
Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Inject Dirichlet data



A hierarchical variant of the Schur complement method.

Gunnar Martinsson

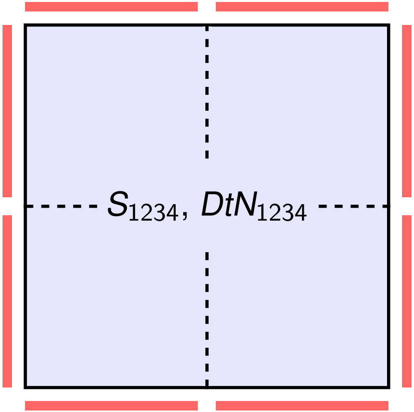
Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Apply merged operators $\mathcal{O}(p^2 + p \log p)$



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.

Gunnar Martinsson

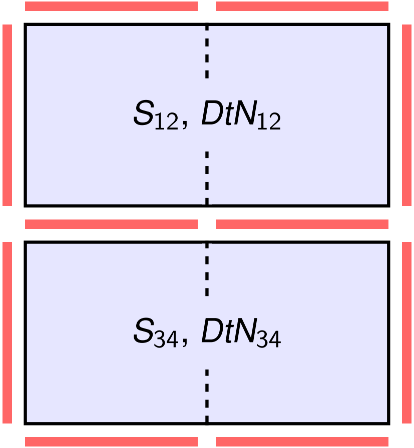
Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Apply merged operators $\mathcal{O}(p^2 + p \log p)$



The ultraspherical spectral element method

Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.


Gunnar Martinsson

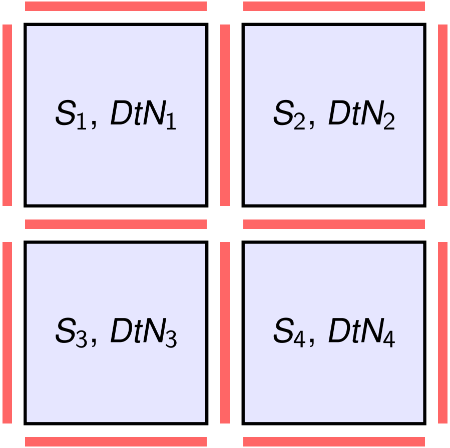

Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Apply merged operators $\mathcal{O}(p^2 + p \log p)$



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

A hierarchical variant of the Schur complement method.


Gunnar Martinsson

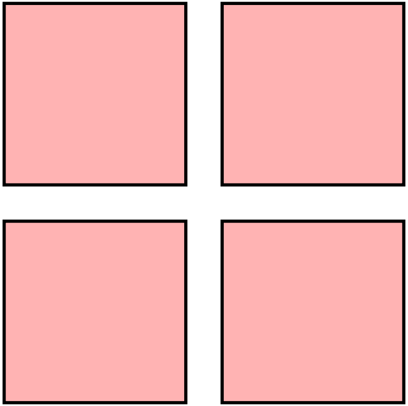

Adrianna Gillman

[Martinsson, 2013]

[Gillman & Martinsson, 2014]

Solution operators stored *in memory!*

Apply merged operators $\mathcal{O}(p^2 + p \log p)$



The ultraspherical spectral element method
Hierarchical Poincaré–Steklov scheme

The sparsity of the ultraspherical spectral method allows us to build solution operators on each leaf in $\mathcal{O}(p^4)$ instead of $\mathcal{O}(p^6)$. For $N = p^2/h^2$ degrees of freedom:

$$\underbrace{\frac{p^4}{h^2}}_{\text{leaf computation}} + \underbrace{\frac{p^3}{h^3}}_{\text{merge cost}} + \underbrace{\frac{p^2}{h} + \frac{p}{h}\log\frac{p}{h}}_{\text{solve cost}} \approx \frac{p^4}{h^2} + \frac{p^3}{h^3} \approx Np^2 + N^{3/2}$$

The storage complexity scales as $\mathcal{O}(p^3/h^2)$.

The ultraspherical spectral element method
Timestepping

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \nabla^2 u, && \boldsymbol{x} \text{ in interior} \\
u(\boldsymbol{x}, 0) &= f(\boldsymbol{x}), && \boldsymbol{x} \text{ in interior} \\
u(\boldsymbol{x}, t) &= g(\boldsymbol{x}, t), && \boldsymbol{x} \text{ on boundary}
\end{aligned}
$$

Discretize in time with with an implicit method, e.g., backward Euler. At each time point we must solve

$$
\begin{aligned}
\left(I - \Delta t \nabla^2\right) u^k &= u^{k-1}, && \boldsymbol{x} \text{ in interior} \\
u^k &= g^k, && \boldsymbol{x} \text{ on boundary}
\end{aligned}
$$

The ultraspherical spectral element method
Timestepping

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla^2 u, && \boldsymbol{x} \text{ in interior} \\ u(\boldsymbol{x}, 0) &= f(\boldsymbol{x}), && \boldsymbol{x} \text{ in interior} \\ u(\boldsymbol{x}, t) &= g(\boldsymbol{x}, t), && \boldsymbol{x} \text{ on boundary} \end{aligned}$$

Discretize in time with with an implicit method, e.g., backward Euler. At each time point we must solve

$$\begin{aligned} \underbrace{\left(I - \Delta t \nabla^2\right)}_{\text{construct solution operator once, apply in } \mathcal{O}(p^2) \text{ with downwards pass}} u^k &= u^{k-1}, && \boldsymbol{x} \text{ in interior} \\ u^k &= g^k, && \boldsymbol{x} \text{ on boundary} \end{aligned}$$

Solution operator at top level can be reused for **fast implicit solves**.

Demo

Ongoing work

- **Benchmarking**: rigorous timing tests to determine practicality.
- **Adaptivity**: automatically detect where to refine h and p.
- **Timestepping**: solution operator can be reused for fast implicit solves.
- **Skinny elements**: high accuracy on elements with small aspect ratio.
- **Parallelizablity**: leaf computations decouple.

Thank you

(Open-source code coming soon.)