

A filtering framework for Finite Volume / Element schemes

Julia Docampo, Matthew Picklo,
Soraya Terrab and Jennifer K. Ryan

Post-processing data with SIAC filters

Bramble, Schatz (Math Comp 1977), Cockburn et al. (Math Comp 2002)

- ❶ **Superconvergence:** for a DG/FEM solution of degree p
 - Proven $2p + 1$ order (L^2 and L^∞ norms) for linear PDEs
 - Observed $2p + 1$ order for non-linear PDEs
- ❷ **Smoothness:** the filtered data is a (local) $2p + 1$ polynomial
 - Removes oscillations in the error
 - Recovers continuity levels across element interfaces

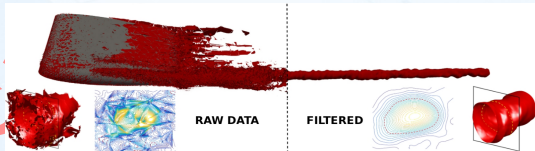
Applications

Flow visualization

Shock detection

Multiresolution analysis

Cut cells



Jallepalli, Docampo, Ryan, Haines, Kirby (TVCG 2017)

Post-processing data with SIAC filters

Goal:

- Establish a filtering framework for general purpose
- Create a standalone tool for general applications

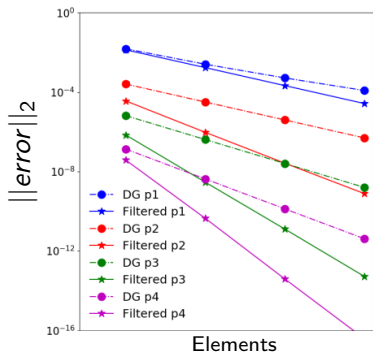
➤ SIAC filters overview

- Superconvergence, kernel structure and filter properties
- Applications to multidimensional data

➤ The software package

SIAC filtering: superconvergence for DG solutions

Advection 1D

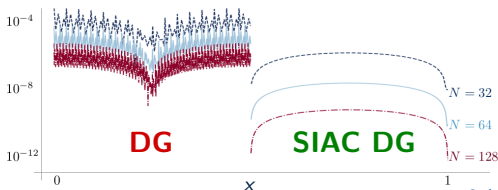


→ DG approximation order: $p + 1$

→ SIAC DG order: $2p + 1$

- ✓ Reduces oscillations
- ✓ General error reduction

Point-wise errors (log) for $p = 2$



Smoothness-Increasing Accuracy-Conserving filters

We post-process our data via convolution:

$$\text{data}^*(x) = \int_{\mathbb{R}} \mathbf{K}(y - x) \cdot \text{data}(y) dy$$

SIAC kernel: weighted sum of B-spline functions.

$$\mathbf{K}^{(r+1,n)}(\cdot) = \sum_{\gamma=1}^{r+1} \mathbf{c}_{\gamma} \cdot B_{\tau_{\gamma},n}(\cdot)$$

Choose \mathbf{c}'_{γ} s to satisfy consistency + r -moments:

$$\int_{\mathbb{R}} K(x) dx = 1, \quad \int_{\mathbb{R}} K(x) x^j dx = 0, \quad j = 1, 2, \dots, r.$$

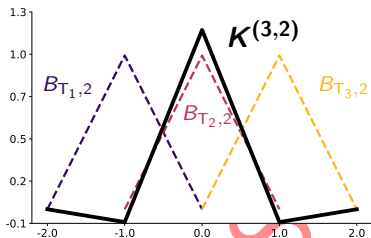
- This ensures data accuracy is preserved
- For $2p + 1$ superconvergence, we need $2p$ moments

$$\text{SIAC kernel: } K^{(r+1,n)} = \sum_{\gamma=1}^{r+1} c_{\gamma} \cdot B_{T_{\gamma},n}$$

(Some) B-splines properties:

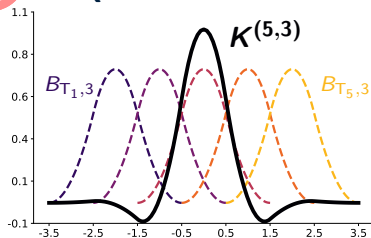
- Compact support (smaller integral region)
- Smoothness of $n - 2$ (remove oscillations)
- Derivatives as divided differences (superconvergence theory)

Linear kernel



- Moments: $r = 2$
- Continuity: C^0 ($n = 2$)
- Optimal order: 3

Quadratic kernel



- Moments: $r = 4$
- Continuity: C^1 ($n = 3$)
- Optimal order: 5

Polynomial functions: filtering linear data ($p = 1$)

Docampo, Jacobs, Li, Ryan (CAF 2020)

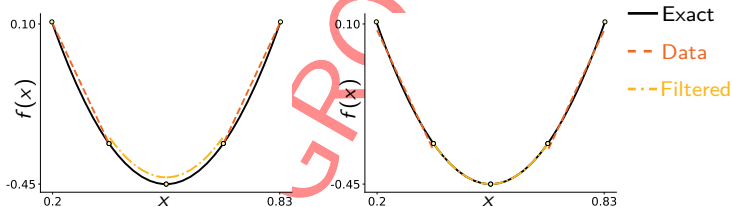
Target

Interpolant

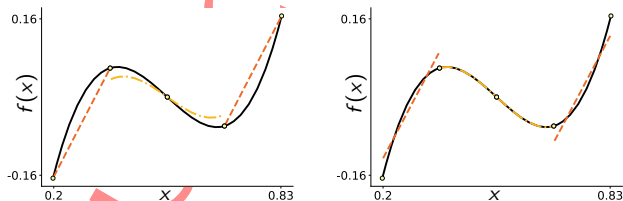
L^2 -projection

(No Galerkin orthogonality)

Quadratic
Function



Cubic
Function



We can recover the exact function for the L^2 -projected data!

Filtering in multidimensions

- **Tensor filter:** natural extension to higher dimension

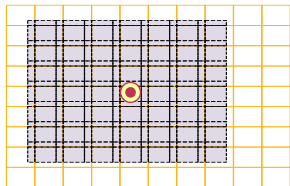
$$K = k_x \otimes k_y \Rightarrow u^*(\bar{x}, \bar{y}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_x \left(\frac{\bar{x} - x}{h_x} \right) \cdot k_y \left(\frac{\bar{y} - y}{h_y} \right) u(x, y) dx dy$$

- **Line filter:** choose angle of rotation and perform a line integral

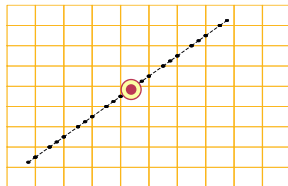
Docampo, Mirzargar, Ryan, and Kirby (SISC 2017)

$$K = k_{\Gamma} \Rightarrow u^*(\bar{x}, \bar{y}) = \int_{\Gamma} k_{\Gamma} \left(\frac{\Gamma(0) - \Gamma(t)}{h_t} \right) u(\Gamma(t)) dt$$

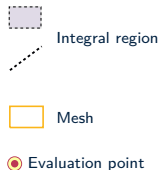
Footprints for a uniform mesh and quadratic kernels $K^{(5,3)}$



Tensor: 196 2d-integrals



Line: 21 1d-integrals



2D filters: superconvergence & error reduction

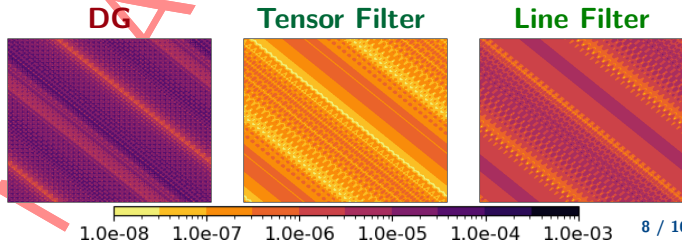
- **Provable** $2p + 1$ order for linear problems + uniform meshes
- **Observed** $2p + 1$ order also for non-linear equations

Example: Burgers eq. with source term and uniform quad mesh

		DG		LSIAC		TSIAC	
	N	$\ e\ _2$	Rate	$\ e\ _2$	Rate	$\ e\ _2$	Rate
$p = 1$	40^2	3.02e-03	2.01	3.34e-04	3.32	1.37e-04	3.05
	80^2	7.52e-04	2.00	4.74e-05	2.82	1.63e-05	3.07
$p = 2$	40^2	4.01e-05	3.00	9.37e-06	6.04	4.66e-07	4.65
	80^2	4.93e-06	3.02	2.95e-07	4.99	1.52e-08	4.93

Error contours

$p = 2, N = 40^2$



Post-processing data with SIAC filters

✓ SIAC filters overview

Moment preservation + B-splines + L^2 -initialization
⇒ **provable** $2p + 1$ **superconvergence**

Multidimensional data

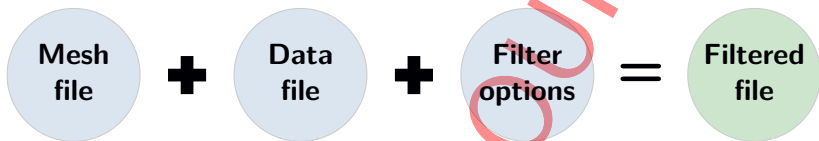
Line filter: computationally efficient

Tensor filter: theory extends naturally from 1d

➤ The software package

- Structure and compatibility
- Examples of applications

A standalone tool written in



- **Mesh map:** element indices, connectivity (and boundaries)
- **Data:** quadrature points or a modal file
- **Pipe** with .vtu files using VTKDataIO (needs Python)
- **Parallel:** using *Julia multi-threading & distributed memory*

Currently supporting 1D & 2D data:

	Mesh					
	Kernel		Uniform		Non-uniform	
Filter	Symmetric	Shifted	Quads	Triangles	Quads	Triangles
Tensor						
Line						

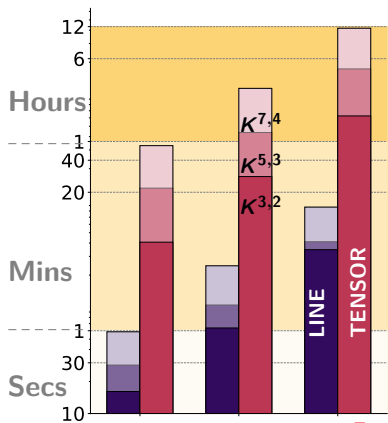


Theory & implemented



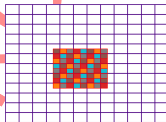
Implemented but no theory

CPU times: Dell XPS, 4 cores, 25 points per element

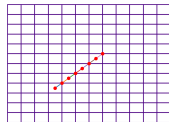


elements: 1.6k 6.4k 25.6k
points: 40k 160k 640k

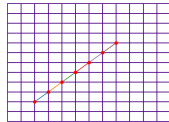
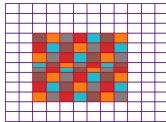
Tensor
 $K^{3,2}$



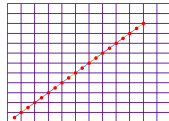
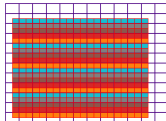
Line



$K^{5,4}$

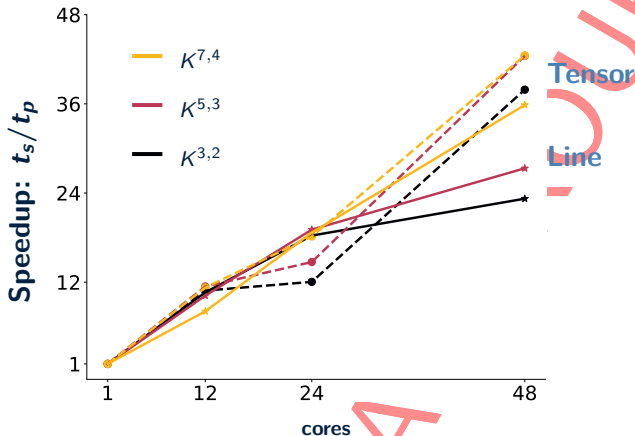


$K^{7,5}$



Filters support (1 point)

CPU times: MareNostrum4, max 48 cores (1 node)



25.6k elements 640k points

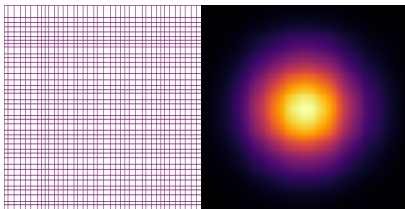
CPUs/ Order	Line (mins)		Tensor (hrs)	
	4	48	4	48
2	5.8	0.5	1.7	0.1
3	6.9	0.7	4.7	0.3
4	14.5	1.1	11.4	0.8

Examples on non-uniform meshes: quads

$p = 1$

Mesh: 40×40

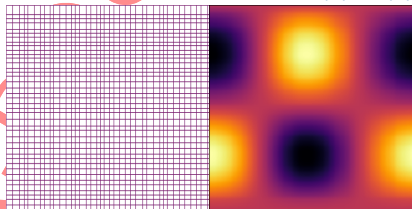
$$u = e^{-x^2 - y^2}$$



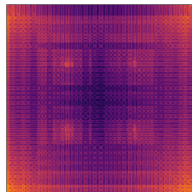
$p = 2$

Mesh: 40×40

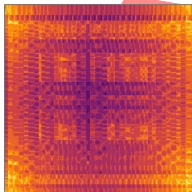
$$u = \sin(x) \cos(y)$$



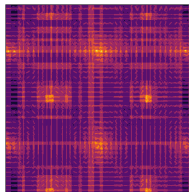
Error contours (log)



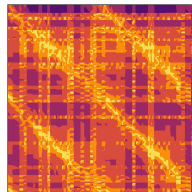
L^2 -projection



Tensor Filter



L^2 -projection



Line Filter

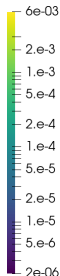
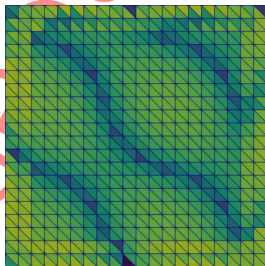
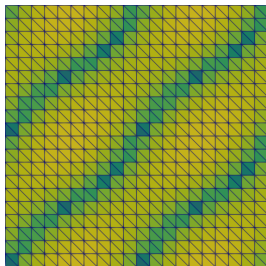
LSIAC filtering: triangular meshes and Paraview

Error contours ($p = 1, n = 20^2$) for a sine wave (DG at $t = 0$)

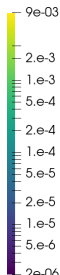
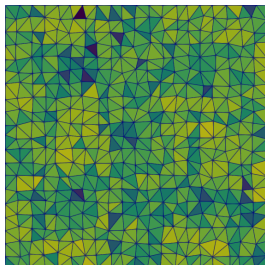
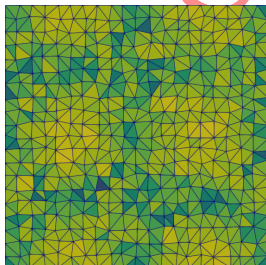
L^2 projection

LSIAC filter

Structured



Unstructured



Wrapping up: hopefully you will remember...

SIAC filters: increase smoothness and reduce errors

→ $2p + 1$ superconvergence

moment conditions + Bsplines + L^2 -initialization

→ Theoretical estimates

equations (linearity), mesh type & domain boundaries

→ Multidimensional data

tensor filter (accuracy) vs. line filter (CPU efficiency)

The Julia Package: currently supporting 1D & 2D data

Mesh						
	Kernel		Uniform		Non-uniform	
Filter	Symmetric	Shifted	Quads	Triangles	Quads	Triangles
Tensor						
Line						



Theory & implemented



Implemented but no theory

The **MSIAC tool** is a joint project with Prof. Jennifer K. Ryan

Visit us!

https://siac_magic.gitlab.io/siac-magic-tools



Acknowledgements

- BSC supercomputing facilities: **MareNostrum4**
- This project is funded through The European Union's Horizon 2020 research and innovation programme under the **Marie Skłodowska-Curie grant agreement No 893378**

Gracias !!

