# Efficient Implementation of Smoothness-Increasing Accuracy Conserving (SIAC) Filters for Discontinuous Galerkin (DG) Solutions

Hanieh Mirzaee
University of Utah

Jennifer K. Ryan
Delft University of Technology

Robert M. Kirby
University of Utah

## Why SIAC Filtering

- The discontinuous Galerkin method continues to maintain heightened levels of interest within the simulation community because of its discretization flexibility.
- This flexibility causes a plethora of difficulties in simulation post-processing such as streamlining.
- SIAC filtering **enhances the smoothness** of the field by eliminating the discontinuity between elements.
- High-order **accuracy is preserved** and in many cases increased.

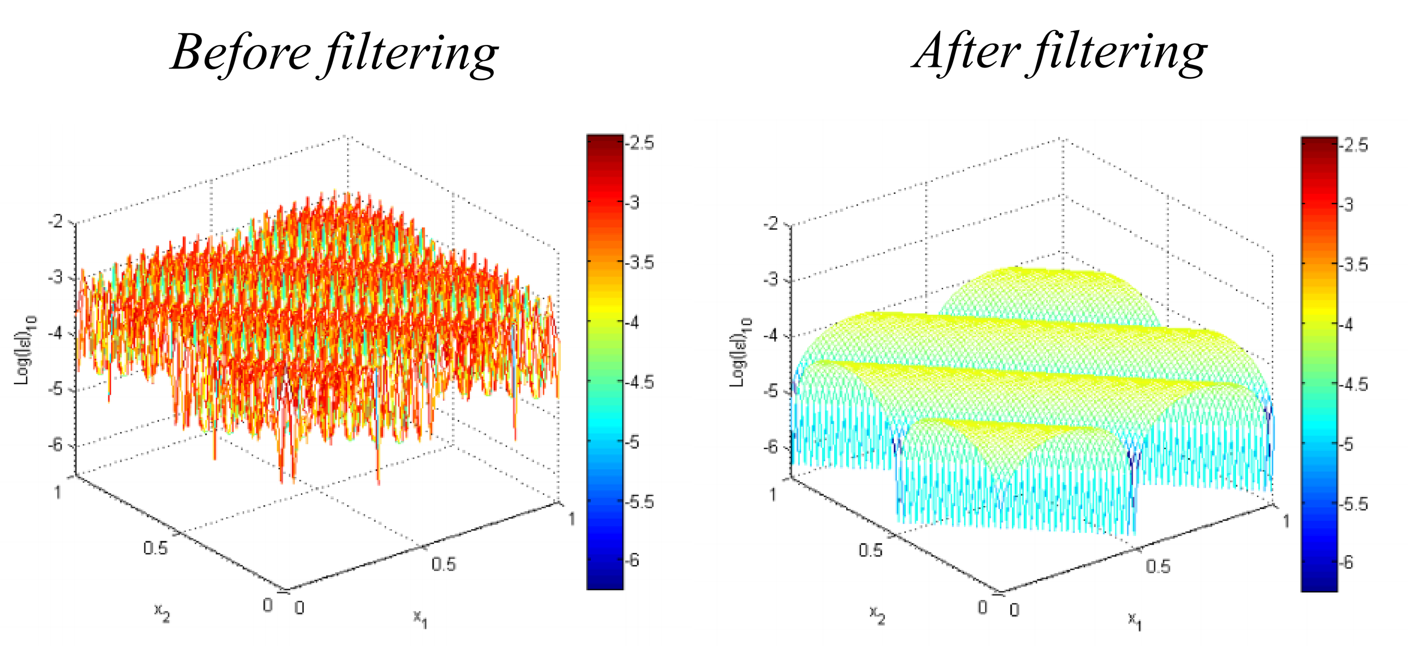


*Figure 1. Point-wise errors in log scale for constant coefficient advection equation over a structured DG triangular mesh.*

## Implementation

- The filtered solution is the result of the **convolution** of the DG approximation and a B-spline based kernel.
In 1D:

$$u^{\star}(x) = \frac{1}{h}\int_{-\infty}^{\infty} K\left(\frac{y-x}{h}\right) u_h(y)\,dy$$

- In higher dimensions the kernel is formed by tensor products of one-dimensional kernels.

## Implementation *(continued)*

- The convolution can be evaluated **exactly** to machine precision:
  - Solving a geometric intersection problem.
  - Up to several numerical integrations within a DG element.
- The computational cost can be **reduced** by evaluating the convolution **approximately** by disregarding the kernel-mesh intersections.

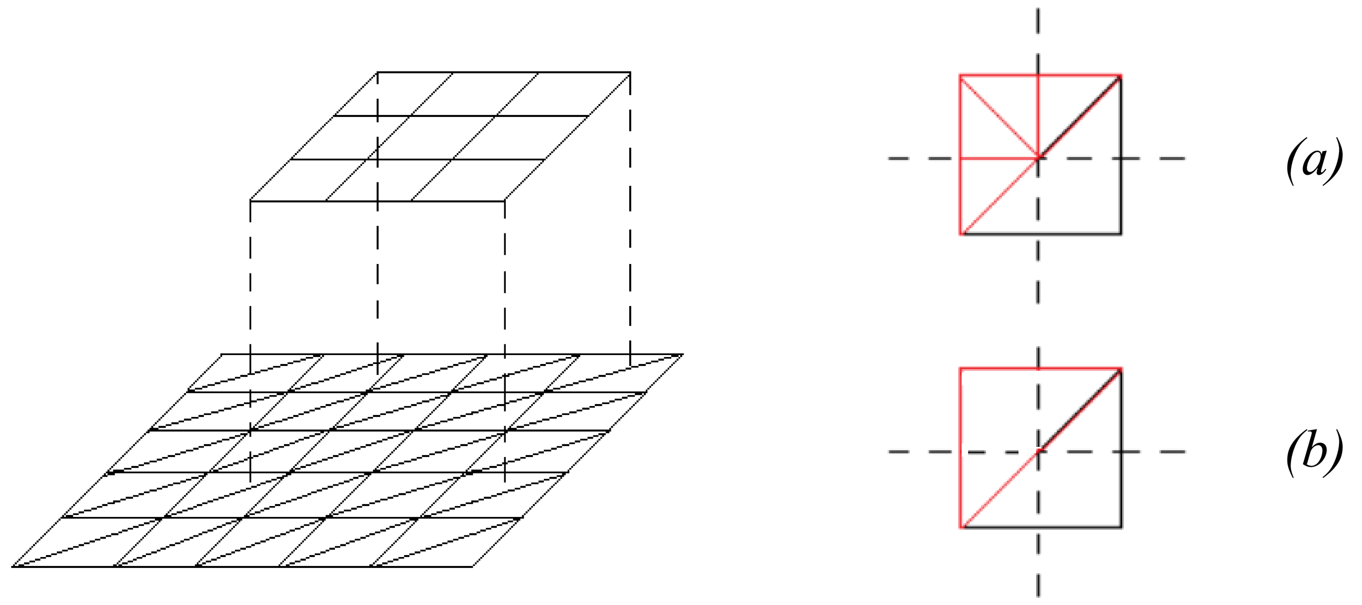


*Figure 3. Kernel-mesh intersection. (a) integration regions in the exact evaluation. (b) integration region in the approximate evaluation.*

- Only **one integration region** is considered in the approximate evaluation, however, more quadrature points might be required.

- Near the boundaries a (partly) **one-sided** form of the kernel is used.

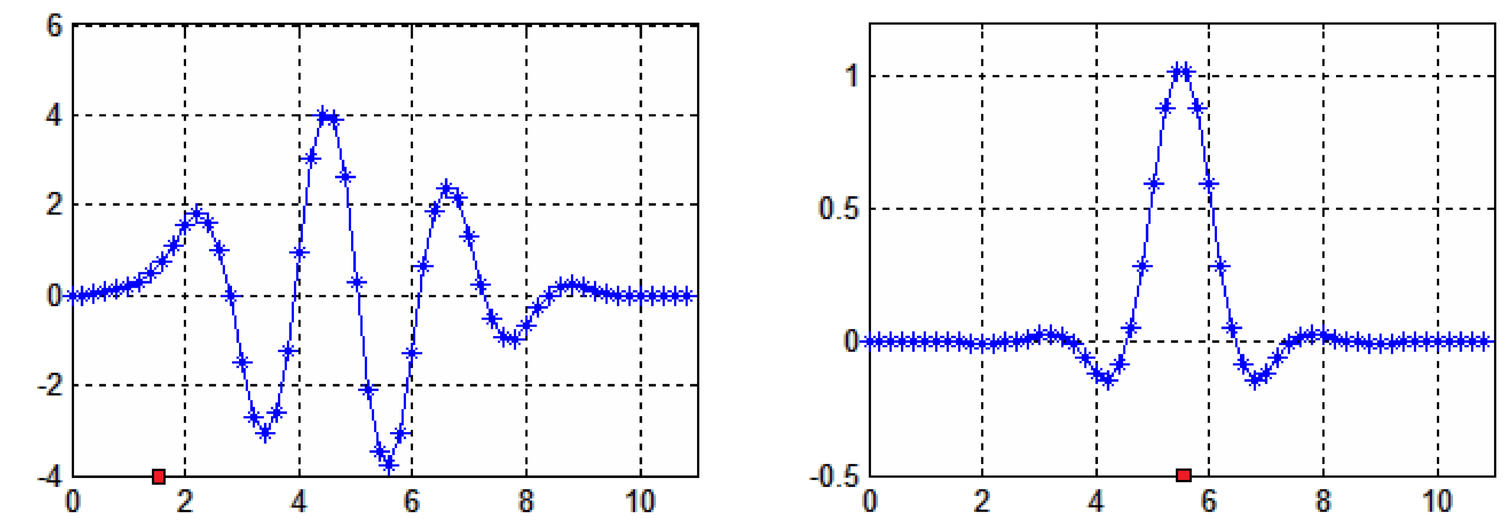

*Figure 4. One-sided vs. Symmetric filtering.*

- A switch between one-sided and symmetric will take place as soon as possible to avoid extra computational costs.

## Parallelization

- Evaluating the post-processed solution at one point is independent of the other.
- Only a few compiler directives are required to gain close to **perfect scaling** on a shared-memory multiprocessors machine using OpenMP.

| | $\mathbb{P}^2$ | | | | |
|---|---|---|---|---|---|
| mesh | $th=1$ | $th=2$ | $th=4$ | $th=8$ | $th=16$ |
| $20^2\times 2$ | 17.64 | 8.81 | 4.49 | 2.23 | 1.17 |
| $40^2\times 2$ | 67.41 | 33.73 | 16.90 | 8.54 | 4.43 |
| $80^2\times 2$ | 266.86 | 133.66 | 66.95 | 33.65 | 17.17 |
| | $\mathbb{P}^3$ | | | | |
| $20^2\times 2$ | 93.27 | 46.19 | 22.90 | 11.49 | 5.82 |
| $40^2\times 2$ | 344.85 | 172.60 | 86.26 | 44.25 | 21.93 |
| $80^2\times 2$ | 1358.81 | 681.31 | 339.55 | 170.48 | 85.24 |

*Figure 5. Timings results in seconds for filtering a DG projection problem over a smoothly-varying triangular mesh. th represents the number of threads.*

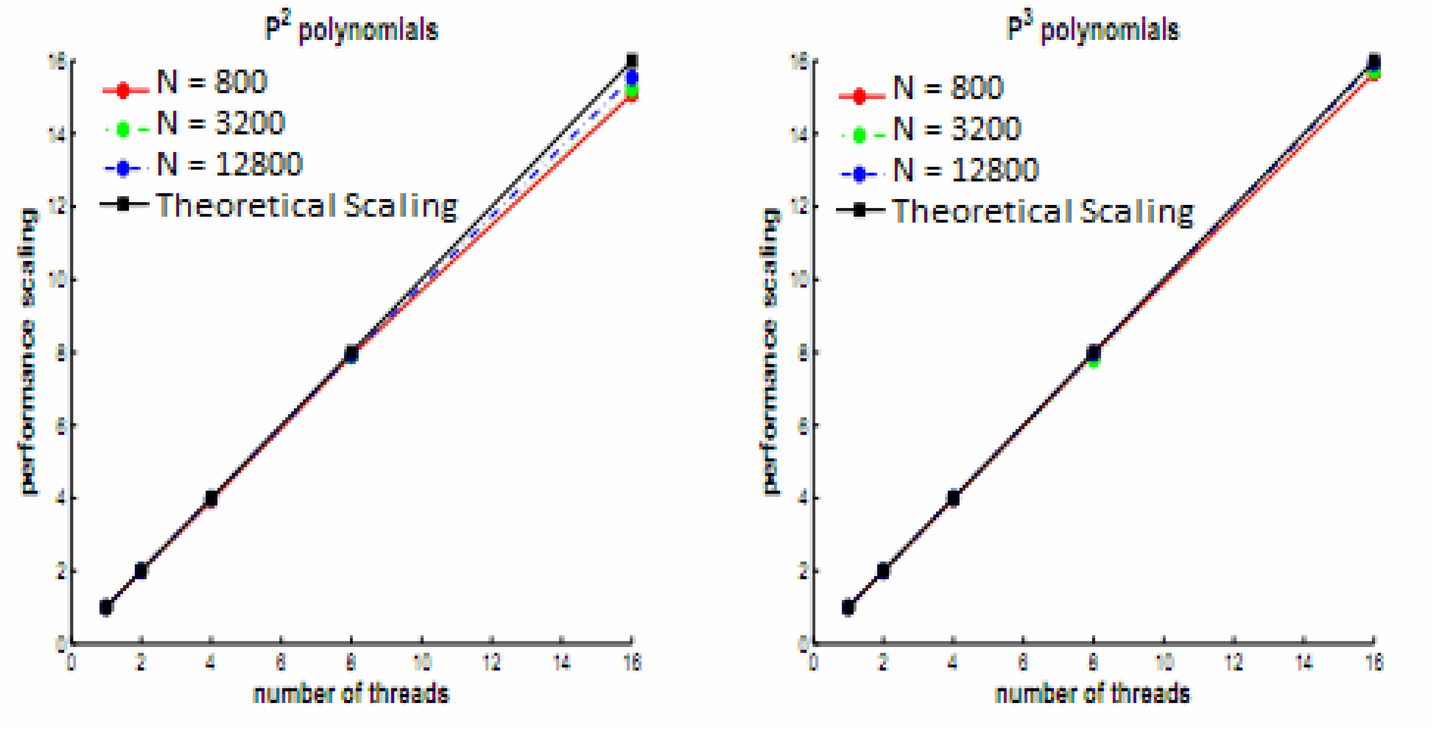


*Figure 6. Performance scaling.*

## Acknowledgments

This project is sponsored in part by the Air Force Office of Scientific Research (AFOSR), Computational Mathematics Program, under grant number FA9550-08-1-0156 and the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-09-1-3055.

## References

H. Mirzaee, J. Ryan, R. Kirby, “Quantification of errors introduced in the numerical approximation and implementation of SIAC filtering of DG field”, Journal of Scientific Computing 45 (2010) 447-470.