AN ADAPTIVE SAVITSKY–GOLAY FILTER
FOR SMOOTHING FINITE ELEMENT COMPUTATION

TEODORO COLLIN\textsuperscript{*}, GORDON KINDLMANN\textsuperscript{†}, AND L. RIDGWAY SCOTT\textsuperscript{‡}

\textbf{Key words.} Finite Element Methods, smoothness-increasing accuracy-conserving filter, Savitzky and Golay filter, smoothing

\textbf{AMS subject classifications.} 65N30, 65D10, 65N15

\textbf{Abstract.} The smoothing technique of Savitzky and Golay is extended to data defined on multidimensional meshes. A smoothness-increasing accuracy-conserving (SIAC) filter is defined that is suitable for use with finite-element computation.

1. Introduction. The approach of Savitzky and Golay \cite{15, 16} to data smoothing involves fitting data via least squares to a polynomial in a window of fixed size. It is applied to data values $y_i$ corresponding to time (or other variable) points $x_i$ that are spaced uniformly, that is, $x_i = ih$ for some $h > 0$. Using the method of least squares, a polynomial $P$ of degree $k$ is chosen so that the expression

$$\sum_{i=1}^{r} (y_i - P(x_i))^2$$

is minimized over all polynomials of degree $k$. This polynomial can be used in various ways: providing an approximation to the data at some point in the window $x_1, \ldots, x_r$, or a derivative at such a point, or a second derivative, and so forth. A major result of \cite{15} was the identification of the least squares processes and subsequent evaluation of approximations as being equivalent to a convolution with a discrete kernel of finite extent in each case. Tables of many such kernels are provided in \cite{15}. The least-squares approach was compared with interpolation in \cite{19}. See also \cite{20}.

The SIAC (smoothness-increasing accuracy-conserving) filters \cite{7, 10} were developed

1. to smooth discontinuities in functions or their derivatives and
2. to extract extra accuracy associated with error oscillation in certain settings.

These are typically implemented using convolution with a localized kernel. When meshes are irregular \cite{10}, this can pose certain difficulties. Moreover, in many problems, the error oscillation is limited (or nonexistent) due to a lack of regularity in the problem. In such problems, high accuracy can be obtained only via adaptive mesh refinement \cite{17}.

The SIAC filters derive from the work of Bramble and Schatz \cite{1} and Thomée \cite{18} who realized that higher-order local accuracy could be extracted by averaging finite element approximations. This was further developed in the context of finite element approximations of hyperbolic equations \cite{4}. It was also realized that this approach could be extended to compute accurate approximations to derivatives even when the original finite elements are discontinuous \cite{14}.

Here we utilize the original idea of Savitzky and Golay \cite{15} to smooth or differentiate data defined via a finite element representation on a multidimensional mesh $M_h$, where $h$ measures in some way the size of the mesh elements. This achieves goal 1. of SIAC filters, without any assumption of error oscillation. This is illustrated in Figure 1 where we apply our approach to smooth the derivative of a standard finite-element approximation.

Our filter does not appear to achieve goal 2., but it does apply with highly refined meshes. It is “accuracy conserving” in a certain sense, so the term SIAC still is applicable, but it is not “accuracy improving” in the sense of \cite{1, 18, 4}. It may be that a different acronym is more appropriate for the technique introduced here, such as SISG, for smoothness increasing Savitzky–Golay.

1.1. Sobolev spaces. To quantify the notions of smoothness and accuracy in SIAC, Sobolev spaces are used, defined as follows. Let $\Omega \subset \mathbb{R}^n$ be a domain of interest, and define

$$\|u\|_{L^2(\Omega)} = \left( \int_{\Omega} |u(x)|^2 \, dx \right)^{1/2}$$

$^*$ The University of Chicago, Chicago, Illinois 60636, USA; email: teocollin@uchicago.edu
$^†$ Department of Computer Science, The University of Chicago, Chicago, Illinois 60637, USA; email: glk@uchicago.edu
$^‡$ The University of Chicago, Chicago, Illinois 60637, USA; email: ridg@uchicago.edu
for a scalar or vector-valued function $u$. Let $\nabla^m$ denote the tensor of $m$-th order partial derivatives. Then

$$\|u\|_s = \sum_{m=0}^{s} \|\nabla^m u\|_{L^2(\Omega)},$$

where $|T|$ denotes the Euclidean norm of a tensor $T$ of arity $\alpha$, thought of as a vector of dimension $n^\alpha$. Define $H^s(\Omega)$ to be the set of functions whose derivatives of order up to $s$ are square integrable, that is, such that $\|u\|_s < \infty$. Note that $\|u\|_0 = \|u\|_{L^2(\Omega)}$. More generally, we write

$$\|T\|_s = \sum_{m=0}^{s} \|\nabla^m T\|_{L^2(\Omega)},$$

for a tensor-valued function $T$. Note that for a tensor $T$ of arity $\alpha$, $\nabla^m T$ is a tensor of arity $\alpha + m$. These norms can be extended to allow non-integer values of $s$ [2].

1.2. Approximation on meshes. Suppose that $\Omega \subset \mathbb{R}^n$ is subdivided in some way by a mesh $M_h$ (e.g., a triangulation, quadrilaterals, prisms, etc.). Suppose further that $W_h$ is a finite element space defined on $M_h$ and that $u_h \in W_h$ is some approximation to a function $u \in H^s(\Omega)$. In many cases [2], an error estimate holds of the form

$$(1.1) \quad \|u - u_h\|_{-t} \leq C h^{s+t} \|u - u_h\|_s, \quad 0 \leq t \leq t_0,$$

where $\|v\|_s$ denotes the norm in $H^s(\Omega)$. When $s$ is negative, the norm in $H^s$ is defined by duality [2].

The approximation $u_h$ could be defined by many techniques, including Galerkin approximations to solutions of partial differential equations [10]. The SIAC objective is to create an operator $\Pi_h$ that maps $u_h$ into a smoother space $V_h$ in a way that maintains this accuracy:

$$(1.2) \quad \|u - \Pi_h u_h\|_{-t} \leq C h^{s+t} \|u - u_h\|_s, \quad 0 \leq t \leq t_0,$$

where the limiting value $t_0$ may change. Moreover, provided that $V_h \subset H^\tau(\Omega)$ for $\tau > 0$, we will also be able to show that

$$(1.3) \quad \|u - \Pi_h u_h\|_{\tau} \leq C h^{s-\tau} \|u - u_h\|_s.$$

This means that $\Pi_h u_h$ can provide optimal-order approximations of derivatives of $u$, even though the space $W_h$ from which $u_h$ comes may harbor discontinuous functions.

We will define $\Pi_h$ as the $L^2(\Omega)$ projection of $W_h$ onto $V_h$. In this way, our proposed SIAC is very similar to the unified Stokes algorithm (USA) proposed in [12].

2. Savitzky-Golay as a projection. Given uniformly spaced points $x_i$ and a fixed window size $r$, define inner-products

$$(2.1) \quad (f, g)_r = \sum_{i=1}^{r} f_i g_i$$
defined on sequences of real numbers \( f \) and \( g \) of length \( r \). We can extend this to be an inner-product on the space of polynomials of degree \( k \) via

\[
(P, Q)_r = \sum_{i=1}^{r} P(x_i)Q(x_i)
\]

To be an inner-product on polynomials of degree \( k \), we must have \( r > k \), for otherwise a nonzero polynomial of degree \( k \) could vanish at all of the grid points.

The method of least squares can be cast as a minimization problem involving these inner-products since

\[
\sum_{i=1}^{r} (y_i - P(x_i))^2 = (P - y, P - y)_r.
\]

More precisely, the inner-products are defined on \( \mathbb{R}^r \), and we think of the space of polynomials of degree \( k \) as a \( k+1 \) dimensional subspace via the identification \( f_i = P(x_i) \) for all \( i \). Minimizing (2.1) is equivalent to projecting \( y \) onto polynomials of degree \( k \) as a subspace of \( \mathbb{R}^r \).

3. Least-squares projections for finite elements. We can define the projection \( \Pi_h \) from \( L^2(\Omega) \) to \( V_h \) by

\[
(\Pi_h u, v)_{L^2(\Omega)} = (u, v)_{L^2(\Omega)} \quad \forall v \in V_h.
\]

This finite dimensional system of equations is easily defined and solved in automated systems [13, 17]. Although this requires solution of a global system, it allows the use of general meshes, including highly graded meshes. It is convenient to restrict to ones that are nondegenerate, by which we mean that each element \( e \) of the mesh (simplex, cube, prism, etc.) has the following properties.

3.1. Mesh assumptions.

**Definition 3.1.** Let \( D \) be a finite, positive integer. Let \( M_h \) be a family of meshes consisting of elements \( e \in M_h \). For each \( e \in M_h \), let \( \rho_e \) be the diameter of the largest sphere contained in \( e \), and let \( h_e \) be the diameter of the smallest sphere containing \( e \). We say that the family of meshes \( M_h \) is **nondegenerate** if there is a constant \( \gamma \) such that, for all \( h \) and \( e \),

\[
h_e \leq \gamma \rho_e
\]

and such that \( e \) is a union of at most \( D \) domains that are star-shaped [5] with respect to a ball of radius \( h_e/\gamma \).

In most cases [2], \( D = 1 \), that is, the elements are star-shaped with respect to a ball. This holds, for example, if the elements are all convex. Thus the usual definition [2] is based just on (3.2). But we have allowed \( D > 1 \) for generality.

Note that the parameter \( h \) is not a single mesh size but rather a function defined on the mesh that prescribes the local mesh size. If the quantifiers guarding (3.2) are rearranged, so that

\[
\max_e h_e \leq \gamma \min_e \rho_e,
\]

then we call the mesh quasi-uniform [2], and we can define a single mesh size \( h = \max_e h_e \).

3.2. Approximation theory. For a nondegenerate family of meshes \( M_h \), there is a constant \( C \) depending only on \( \gamma \) and \( D \), such that, for all \( e \) and \( h \), there is a polynomial \( P_e \) of degree \( k \) with the property

\[
\|\nabla^m (u - P_e)\|_{L^2(e)} \leq C h_e^{k+1-m} \|\nabla^{k+1} u\|_{L^2(e)}.
\]

This is a consequence of the Bramble-Hilbert lemma [5]. It in particular demonstrates adaptive approximation using discontinuous Galerkin methods.

An estimate similar to (3.4) involving an approximation operator \( I_h \), called an interpolant, is well known for essentially all finite-element methods. Thus we make the following **hypothesis**:

\[
\sum_{m=0}^{l} \sum_e h_e^{2m} \|\nabla^m (u - I_h u)\|^2_{L^2(e)} \leq C \sum_e h_e^{2(k+1)} \|\nabla^{k+1} u\|^2_{L^2(e)}.
\]
Note that we have apportioned the quantity $h^{k+1-m}$ in (3.4) partly on the left and partly on the right in (3.5). This balancing act is arbitrary and could be done in a different way. But the thinking in (3.5) is that we have chosen the mesh so that the terms on the right-hand side of the inequality in (3.5) are balanced in some way. That is, we make the mesh size small where the derivatives of $u$ are large.

For discontinuous Galerkin methods, we take $I_h u|_e = P_e$ on each element $e \in M_h$, where $P_e$ comes from (3.4). On a quasi-uniform [2] mesh of size $h$, (3.5) simplifies to
\[
\sum_{m=0}^{t} h_e^m \| \nabla^m (u - I_h u) \|_0 \leq C h^{k+1} \| u \|_{k+1}.
\]

The following can be found in [2] and other sources.

**Lemma 3.2.** Let $V_h$ consist of piecewise polynomials that include complete polynomials of degree $k$ on each element of a nondegenerate subdivision of $\Omega$, such as
- $(t = 0)$ Discontinuous Galerkin (DG) ($k \geq 0$),
- $(t = 1)$ Lagrange ($k \geq 1$), Hermite ($k \geq 3$), tensor-product elements ($k \geq 1$),
- $(t = 2)$ Argyris ($k \geq 5$ in two dimensions),
as well as many others. Then the hypothesis (3.5) holds with $t$ and $k$ as specified.

### 3.3. Estimates for the projection.

**Theorem 3.3.** Under the hypothesis (3.5), and in particular for the spaces $V_h$ listed in Lemma 3.2, there is a constant $C$ such that
\[
(3.6) \quad \sum_{m=0}^{t} h_e^m \| \nabla^m (u - \Pi_h u) \|_{L^2(e)}^2 \leq C \sum_{e} h_e^{2(k+1)} \| \nabla^{k+1} u \|_{L^2(e)}^2
\]
for any nondegenerate family of meshes. The constant $C$ depends only on $\gamma$ and $D$ in Definition 3.1.

**Proof.** First of all, consider the case $t = 0$. In this case, the result to be proved is
\[
(3.7) \quad \sum_{e} \| u - \Pi_h u \|_{L^2(e)}^2 \leq C \sum_{e} h_e^{2(k+1)} \| \nabla^{k+1} u \|_{L^2(e)}^2.
\]
Since the $L^2$ projection provides optimal approximation in $L^2$, we find
\[
\sum_{e} \| u - \Pi_h u \|_{L^2(e)}^2 = \| u - \Pi_h u \|_{L^2(\Omega)}^2 \leq \| u - I_h u \|_{L^2(\Omega)}^2.
\]
Then the hypothesis (3.5) implies (3.7), which completes the proof in the case $t = 0$.

Using the triangle inequality and standard inverse estimates [2] for nondegenerate meshes, we have
\[
\| \nabla^m (u - \Pi_h u) \|_{L^2(e)} \leq \| \nabla^m (u - I_h u) \|_{L^2(e)} + \| \nabla^m (I_h u - \Pi_h u) \|_{L^2(e)}
\leq \| \nabla^m (u - I_h u) \|_{L^2(e)} + Ch_e^{-m} \| I_h u - \Pi_h u \|_{L^2(e)}
\leq \| \nabla^m (u - I_h u) \|_{L^2(e)} + Ch_e^{-m} (\| u - I_h u \|_{L^2(\Omega)} + \| u - \Pi_h u \|_{L^2(\Omega)}).
\]

Squaring and summing this over $e$, and using the hypothesis (3.5) yields
\[
\sum_{e} h_e^{2m} \| \nabla^m (u - \Pi_h u) \|_{L^2(e)}^2 \leq C \left( \sum_{e} h_e^{2(k+1)} \| \nabla^{k+1} u \|_{L^2(e)}^2 + \| u - \Pi_h u \|_{L^2(\Omega)}^2 \right),
\]
with a possibly larger constant $C$. Using (3.7) and summing over $m$ completes the proof. QED

### 3.4. New ambition.
Based on Theorem 3.3, we now replace our assumption (1.1) with something appropriately ambitious. Thus we assume that $u_h \in W_h$ satisfies
\[
(3.9) \quad \| u - u_h \|_{L^2(\Omega)}^2 \leq C \sum_{e} h_e^{2(k+1)} \| \nabla^{k+1} u \|_{L^2(e)}^2.
\]
The following theorem justifies the term SIAC for $\Pi_h$. 
Many standard finite element approximations have become available in automated finite-element systems. Thus visualization of derivatives of finite-element approximations requires continuous (but not \(C\)) forced to deal with visualization of discontinuous objects. We consider three examples in which the choice of the approximation space \(W\) has the exact solution \(u\) of squares divided into two right triangles, using continuous piecewise polynomials of degree 3, the resulting approximation being denoted by \(\Pi_h\). Depicted in Figure 1 is the (discontinuous) derivative \(\partial_x u_h\), together with the smoothed version \(\Pi_h(\partial_x u_h)\).

**Theorem 3.4.** Suppose \(V_h\) is a space as in Lemma 3.2. Under the assumption (3.9), we have

\[
\sum_{m=0}^{k} \sum_{e} h^{-m} e \left\| \nabla^m (u - \Pi_h u) \right\|_{L^2(e)}^2 \leq C \sum_{e} h^{-2(k+1)} e \left\| \nabla^{k+1} u \right\|_{L^2(e)}^2.
\]

**Proof.** The proof is similar to that of Theorem 3.3. We write

\[
u - \Pi_h u_h = u - \mathcal{I}_h u + \mathcal{I}_h u - \Pi_h u_h.
\]

The required estimate for \(u - \mathcal{I}_h u\) is hypothesis (3.5). Using inverse estimates [2] for nondegenerate meshes, we find

\[
h^{-m} e \left\| \nabla^m (\mathcal{I}_h u - \Pi_h u) \right\|_{L^2(e)} \leq C \left\| \mathcal{I}_h u - \Pi_h u \right\|_{L^2(\Omega)} \leq C \left( \|u - \mathcal{I}_h u\|_{L^2(e)} + \|u - \Pi_h u\|_{L^2(e)} \right).
\]

Squaring and summing over \(e\), and applying (3.5), the triangle inequality, and (3.6), yields

\[
\sum_{e} h^{-2m} e \left\| \nabla^m (\mathcal{I}_h u - \Pi_h u) \right\|_{L^2(e)}^2 \leq 2C' \left( \sum_{e} \|u - \mathcal{I}_h u\|_{L^2(e)}^2 + \sum_{e} \|u - \Pi_h u\|_{L^2(e)}^2 \right).
\]

\[
\leq C'' \left( \sum_{e} h^{-2(k+1)} e \left\| \nabla^{k+1} u \right\|_{L^2(e)}^2 + \sum_{e} \|u - \Pi_h u\|_{L^2(e)}^2 \right),
\]

for appropriate constants \(C'\) and \(C''\). Since \(\Pi_h\) is the \(L^2(\Omega)\) projection,

\[
\sum_{e} \|\Pi_h (u - u_h)\|_{L^2(e)}^2 = \|\Pi_h (u - u_h)\|_{L^2(\Omega)} \leq \|u - u_h\|_{L^2(\Omega)}.
\]

Thus applying (3.9) completes the proof. QED

**4. Applications.** There are situations in which a finite-element approximation, or its derivative, is naturally discontinuous. The SIAC-like operator proposed here is then essential if we want to visualize the corresponding approximations accurately. We consider three examples in which the choice of the approximation space \(W_h\) is dictated by the structure of the problem. Switching to a smoother space may give suboptimal results, so we are forced to deal with visualization of discontinuous objects.

**4.1. Derivatives of standard Galerkin approximations.** Many standard finite element approximations [2, 17] use continuous (but not \(C^1\)) finite elements. It is only recently [9] that \(C^1\) elements have become available in automated finite-element systems. Thus visualization of derivatives of finite-element approximations requires dealing with discontinuous piecewise polynomials.

We take as an example the equation

\[-\Delta u = 32\pi^2 \cos(4\pi x) \sin(4\pi y) \quad \text{in} \quad [0, 1]^2, \quad u = 0 \quad \text{on} \quad \partial [0, 1]^2,
\]

which has the exact solution \(u(x, y) = -\cos(4\pi x) \sin(4\pi y)\). This was approximated on an \(8 \times 8\) regular mesh of squares divided into two right triangles, using continuous piecewise polynomials of degree 3, the resulting approximation being denoted by \(u_h\). Depicted in Figure 1 is the (discontinuous) derivative \(\partial_x u_h\), together with the smoothed version \(\Pi_h(\partial_x u_h)\).
4.2. Visualizing the pressure in Stokes. Most finite-element methods for solving the Stokes, Navier-Stokes, or non-Newtonian flow equations [17, 6] involve a discontinuous approximation of the pressure. The unified Stokes algorithm (USA) proposed in [12] uses the projection method described here to smooth the pressure. In Figure 2, we present the example from [12]. In this case, $\Pi_h p_h$ is exactly the USA pressure.

4.3. DG for mixed methods. One approach to approximating flow in porous media with discontinuous physical properties is to use mixed methods [17] and discontinuous finite elements. This approach is called discontinuous Galerkin (DG). Thus visualization of the primary velocity variable is a challenge due to its discontinuity. DG methods are also widely applied to hyperbolic PDEs [4]. In Figure 3, we depict the solution of the problem in [17, (18.19)] using BDM elements of order 1.

4.4. Highly refined meshes. Highly refined meshes are used to resolve solution singularities in many contexts. For example, they can be required to resolve singularities in data [17, Section 6.2]. Even when data is smooth, singularities can arise due to domain geometry or changes in boundary condition types [17, Section 6.1]. Such singularities are common, and physical quantities are often associated with derivatives of solutions to such problems. For example, the function $g$ defined in polar coordinates by

$$g(r, \theta) = r^{1/2} \sin(\frac{1}{2}\theta)$$

arises naturally in this context. Consider the boundary value problem to find a function $u$ satisfying the partial differential equation

$$-\Delta u = 1$$

in the domain $\Omega := [-\frac{1}{2}, \frac{1}{2}] \times [0, 1]$, together with the boundary conditions

$$u = 0 \text{ on } \{(x, 0) : 0 \leq x \leq \frac{1}{2}\}, \quad \frac{\partial u}{\partial n} = 0 \text{ on } \{(x, 0) : -\frac{1}{2} \leq x \leq 0\},$$

with $u = g - \frac{1}{2}y^2$ on the remainder of $\partial \Omega$. Note that $g$ is a smooth function on this part of the boundary. Then the exact solution is $u = g - \frac{1}{2}y^2$.

This problem has a singularity at $(0,0)$; the derivative $u_x$ is infinite there. The solution $u$ of this problem is shown in [17, Figure 16.1], computed using automatic, goal-oriented refinement and piecewise-linear approximation. Here we focus instead on $u_x$. In this case, the derivative of the piecewise linear approximation is a piecewise constant, and so the SISG approach seems warranted. Thus we project $u_x$ onto continuous piecewise linear functions on the same mesh, and the result is depicted in Figure 4. This approach is actually the default approach taken in DOLFIN [11]. We see that SISG is quite effective in representing a very singular, discontinuous computation in a comprehensible way.
To make this result more quantitative, we compare with the exact solution derivative $u_x$. Note that we can write
$$g(x, y) = (x^2 + y^2)^{1/4} \sin \left( \frac{1}{3} \arctan(y/x) \right),$$
so that
$$g_x(x, y) = \frac{1}{2} (x^2 + y^2)^{-3/4} \left( x \sin \left( \frac{1}{3} \arctan(y/x) \right) - y \cos \left( \frac{1}{3} \arctan(y/x) \right) \right).$$
To avoid the singularity at the origin in the computations, we introduce $\epsilon > 0$ and replace $g_x$ by
$$g^\epsilon_x(x, y) = \frac{1}{2} (\epsilon + x^2 + y^2)^{-3/4} \left( x \sin \left( \frac{1}{3} \arctan(y/x) \right) - y \cos \left( \frac{1}{3} \arctan(y/x) \right) \right).$$
This introduces a small error that we minimize with respect to $\epsilon$. More precisely, we compute
\begin{equation}
\|\Pi(u_h, x) - g^\epsilon_x\|_{L^2(\Omega)}.
\end{equation}
The computational results are given in Table 1. The goal of the adaptivity was to minimize
\begin{equation}
\|\nabla(u_h - u)\|_{L^2(\Omega)},
\end{equation}
and the data in Table 1 indicates that this was successful. The initial mesh consisted of eight 45° right-triangles, with 9 vertices, in all of the computations. For tolerances greater than 0.03, no adaptation occurs.

The $H^1(\Omega)$ error
\begin{equation}
\left( \|u_h - u\|_{L^2(\Omega)}^2 + \|\nabla(u_h - u)\|_{L^2(\Omega)}^2 \right)^{1/2}
\end{equation}
is closely related to (4.2), and the quantity (4.3) is presented in the 4-th column in Table 1. These values were computed via the DOLFIN function `errornorm` without any regularization of the exact solution $u(x, y) = g(x, y) - \frac{1}{2} y^2$. This quantity is a non-SISG error, in that the gradients are treated as discontinuous piecewise-defined functions. By contrast the SISG error (4.1) corresponds to part of the norm in (4.3) (the $x$-derivative), but this error is a measure of the accuracy of the SISG projection.

Both (4.2) and (4.1) decay approximately like $CN^{-1/2}$, where $N$ is the number of vertices in the mesh (including boundary vertices) after adaptation. For a smooth $u$ and a regular mesh of size $h$, we would expect the quantities (4.2) and (4.1) to be $O(h)$, and $N = O(h^{-2})$ in this case. So the observed convergence $CN^{-1/2}$ in Table 1 is best possible. Note that the tolerance value is associated with the square of the quantity (4.2).
\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
N & tolerance & SISG error (4.1) & H1 error & $\epsilon$ \\
\hline
9 & 0.03 & 1.64e-01 & 4.21e-01 & 1.00e-03 \\
29 & 0.01 & 9.68e-02 & 2.36e-01 & 1.00e-03 \\
146 & 0.002 & 3.53e-02 & 9.69e-02 & 1.00e-05 \\
482 & 0.001 & 1.70e-02 & 5.42e-02 & 1.00e-06 \\
4477 & 0.0001 & 4.11e-03 & 1.64e-02 & 1.00e-08 \\
41481 & 0.00001 & 1.06e-03 & 5.22e-03 & 1.00e-11 \\
427169 & 0.000001 & 2.51e-04 & 1.65e-03 & 1.00e-13 \\
\hline
\end{tabular}
\caption{Errors (4.1) for the SISG technique. The initial mesh size was 2 for all of the computations. $N$ denotes the number of vertices in the mesh (including boundary vertices) after adaptation, the second column indicates the tolerance used for adaptivity, the third column denotes the error quantity defined in (4.1), “H1 error” denotes $\|u - u_h\|_{H^1(\Omega)}$, and the last column gives the value of $\epsilon$ used in (4.1).}
\end{table}

5. Methods. Except for Figure 4, the images presented here were produced using Firedrake [13] and Diderot [3, 8]. Figure 4 was done with DOLFIN [11].

6. Conclusions. The smoothing technique of Savitsky and Golay can be extended to finite element methods in a useful way. This allows accurate presentation of derivatives of piecewise-defined functions, even for highly refined meshes. It also allows discontinuous approximations, such as Discontinuous Galerkin, to be visualized in an effective way.

REFERENCES

[1] J. H. Bramble and A. H. Schatz, Higher order local accuracy by averaging in the finite element method, Math. Comp., 31 (1977), pp. 94–111.
[2] S. C. Brenner and L. R. Scott, The Mathematical Theory of Finite Element Methods, Springer-Verlag, third ed., 2008.
[3] C. Chiw, G. Kindlmann, J. Reppy, L. Samuels, and N. Seltzer, Diderot: A parallel DSL for image analysis and visualization, in Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), June 2012, pp. 111–120, https://doi.org/10.1145/2254064.2254079. (alphabetical author order).
[4] B. Cockburn, M. Luskin, C.-W. Shu, and E. Süli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations, Math. Comp., 72 (2003), pp. 577–606.
[5] T. Dupont and L. R. Scott, Polynomial approximation of functions in Sobolev spaces, Math. Comp., 34 (1980), pp. 441–463.
[6] V. Girault and L. R. Scott, Oldroyd models without explicit dissipation, Rev. Roumaine Math. Pures Appl., accepted (2018).
[7] A. Jallepalli, J. Docampo-Sánchez, J. K. Ryan, R. Haimes, and R. M. Kirby, Local derivative post-processing for the discontinuous galerkin method, J. Comput. Appl. Math., 294 (2016), pp. 275–296.
[8] G. Kindlmann, C. Chiw, N. Seltzer, L. Samuels, and J. Reppy, Diderot: a domain-specific language for portable parallel scientific visualization and image analysis, IEEE Transactions on Visualization and Computer Graphics (Proceedings VIS 2015), 22 (2016), pp. 867–876, https://doi.org/10.1109/TVCG.2015.2467449.
[9] A. J. Kitchen and W. G. M. Brown, Oldroyd models without explicit dissipation, Rev. Roumaine Math. Pures Appl., accepted (2018).
[10] X. Li, J. K. Ryan, R. M. Kirby, and C. Vuik, Smoothness-increasing accuracy-conserving (SIAC) filters for derivative approximations of Discontinuous Galerkin (DG) solutions over nonuniform meshes and near boundaries, J. Comput. Appl. Math., 294 (2016), pp. 275–296.
[11] A. Logg and G. N. Wells, DOLFIN: Automated finite element computing, ACM Trans. Math. Software, 37 (2010), p. 20.
[12] H. Morgan and L. R. Scott, Towards a unified finite element method for the Stokes equations, SIAM J. Sci. Comput., 40 (2018), pp. A130–A141.
[13] F. Rathegeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly, Firedrake: automating the finite element method by composing abstractions, ACM Trans. Math. Software, 43 (2017), pp. 24:1–24:27, https://doi.org/10.1145/2998441.
[14] J. K. Ryan and B. Cockburn, Local derivative post-processing for the discontinuous galerkin method, J. Comput. Phys., 228 (2009), pp. 8642–8664.
[15] A. Savitzky and M. J. E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Analytical Chemistry, 36 (1964), pp. 1627–1639.
[16] L. R. Scott and M. V. T. Adams, An efficient method for data smoothing via least-squares polynomial fitting, SIAM J. Numer. Anal., 26 (1989), pp. 681–692.
[17] L. R. Scott, Introduction to Automated Modeling with FEniCS, Computational Modeling Initiative, 2018.
[18] V. Thomée, High order local approximations to derivatives in the finite element method, Math. Comp., 31 (1977), pp. 652–660.
[19] M. Unser and I. Daubechies, On the approximation power of convolution-based least squares versus interpolation, IEEE Trans. Image Process., 45 (1997), pp. 1697–1711.
[20] D. Van De Ville, W. Philips, and I. Lemahieu, Least-squares spline resampling to a hexagonal lattice, Signal Processing: Image Communication, 17 (2002), pp. 393–408.