Sparse Grid Discontinuous Galerkin (DG) Methods for High Dimensional PDEs

Yingda Cheng

Michigan State University

Midwest NA Day, 2022

Joint work with Wei Guo, Zhanjing Tao, Juntao Huang, Yan Jiang, Yuan Liu

Outline

Motivation

- We are interested in computing a class of high dimensional PDEs.
- Example includes: high dimensional kinetic transport problem (Vlasov, Boltzmann) in plasma, semiconductor device simulations, high dimensional Hamilton-Jacobi equations.
- Conventional deterministic numerical solvers runs into the curse of dimensionality.

The discontinuous Galerkin method

This is a class of finite element method using piecewise "discontinuous" approximation space, and is widely used in many applications.

- Invented by Reed and Hill (73) for neutron transport. First analysis by Lesaint and Raviart (74).
- Runge-Kutta discontinuous Galerkin (RKDG) method by Cockburn and Shu (89, 90,...) for general conservation laws.
- DG methods for elliptic equations and parabolic equations, see review paper Arnold, Cockburn, Brezzi, Marini (02).
- Suitable for calculating transport problems.
- Flexibility with the mesh (hanging nodes, nonconforming mesh);
- Flexibility with choice of approximation space;
- Compact scheme, highly parallelizable;
- Provable convergence properties.
- Adaptivity

x Large number of degrees of freedom.

Sparse grid method: breaking the curse of dimensionality

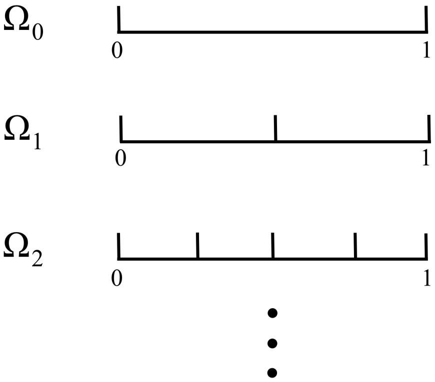
- Sparse grid is first introduced in the quadrature context Smolyak (63), introduced by Zenger (91), developed by Griebel (91,98,05...), widely used in UQ framework Xiu, Hesthaven (05...).
- When solving high-dimensional PDEs, sparse grid method has been incorporated in
  - Finite difference/volume/element methods: Griebel (98); Griebel, Zumbusch (99). Hemker (95); Bungartz, Griebel (04); Schwab, Suli, Todor (08).
  - Spectral methods: Griebel (07); Gradinaru (07); Shen, Wang (10); Shen, Yu (10, 12).
  - **DG methods**: Wang *et al* JCP, 2016, Guo, Cheng, SISC, 2016, 2017, Tao *et al* JCP, SISC, 2019, Liu *et al*, JCP 2019, Tao *et al*, JCP, 2020, Huang *et al*, SISC 2020.

Outline

Hierarchical decomposition of piecewise polynomial spaces in one dimension

Consider $\Omega = [0,1]$ and define n-th level grid

$$\Omega_n = \{I_n^j = (2^{-n}j, 2^{-n}(j+1)], j = 0, \ldots, 2^n - 1\}$$



Hierarchical decomposition of piecewise polynomial spaces in one dimension

Conventional approximation space on the n-th level grid Ω_n

$$V_n^k = \{v : v \in P^k(I_n^j),\ \forall j = 0, \ldots, 2^n - 1\}$$

$$dim(V_n^k) = 2^n(k+1)$$

Nested structure

$$V_0^k \subset V_1^k \subset V_2^k \subset V_3^k \subset \cdots$$

W_n^k: orthogonal complement of V_{n-1}^k in V_n^k, for $n > 1$, represents the finer level details when the mesh is refined, satisfying

$$V_{n-1}^k \oplus W_n^k = V_n^k$$

$$W_n^k \perp V_{n-1}^k$$

Let $W_0^k := V_0^k$, then

$$V_N^k = \bigoplus_{0 \le n \le N} W_n^k$$

$$dim(W_n^k) = \left\lceil 2^{n-1} \right\rceil (k+1)$$

Background for multiwavelet in DG context

- Haar wavelet Haar (1910).
- L^2 orthogonal multiwavelet bases Alpert (1993).
- Adaptive multiresolution DG schemes Calle et al. (2005), Archibald et al. (2011), Hovhannisyan et al. (2014), Gerhard et al. (2015)...
- Multiwavelet trouble cell indicator Vuik, Ryan (2014)...

Hierarchical orthonormal bases: Alpert's multiwavelet

Bases in W_0^k: scaled orthonormal Legendre polynomials.

Bases in W_1^k:

$$h_i(x) = 2^{1/2} f_i(2x-1), \quad i = 1, \dots, k+1$$

The orthonormal, vanishing-moment functions $\{f_i(x)\}_k$ (Alpert 93), which are supported on $(-1,1)$ and depend on k, will be defined later.

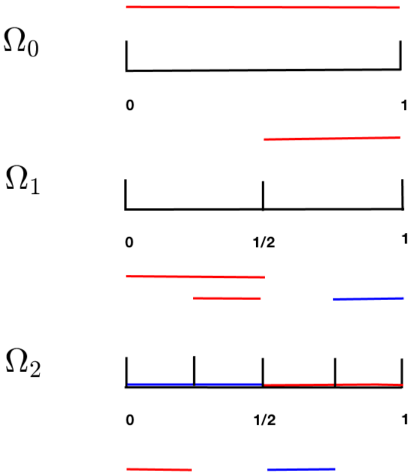
Bases in W_n^k, $n \geq 1$

$$v_{i,n}^j(x) = 2^{(n-1)/2}\, h_i(2^{n-1}x - j), \quad i = 1, \dots, k+1,\ j = 0, \dots, 2^{n-1}-1$$

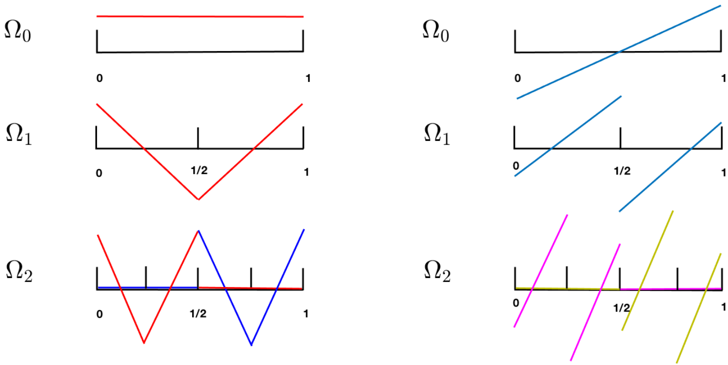
Orthonormality of multiwavelet bases across different hierarchical levels

$$\int_0^1 v_{i,n}^j(x) v_{i',n'}^{j'}(x)\, dx = \delta_{ii'}\delta_{nn'}\delta_{jj'}$$

Bases on different levels for $k = 0$



Bases on different levels for $k = 1$



Approximation space in multi-dimensions

Consider 2D case, $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and multi-index $\mathbf{l} = (l_1, l_2) \in \mathbb{N}_0^2$
The standard rectangular grid $\Omega_{\mathbf{l}}$ with mesh size

$$h_{\mathbf{l}} := (2^{-l_1}, 2^{-l_2})$$
$$h := \min\{2^{-l_1}, 2^{-l_2}\}$$

For each $I_{\mathbf{l}}^{\mathbf{j}} = \{(x_1, x_2) : x_i \in (2^{-l_i} j_i, 2^{-l_i}(j_i + 1)]\}$, the traditional tensor-product polynomial space is

$$\mathbf{V}_{\mathbf{l}}^k = \{\mathbf{v} : \mathbf{v}(\mathbf{x}) \in P^k(I_{\mathbf{l}}^{\mathbf{j}}),\ \mathbf{0} \leq \mathbf{j} \leq 2^{\mathbf{l}} - \mathbf{1}\}$$

P^k denotes polynomial of degree at most k in each dimension.

Approximation space in multi-dimensions

Consider 2D case, $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and multi-index $\mathbf{l} = (l_1, l_2) \in \mathbb{N}_0^2$
The standard rectangular grid $\Omega_{\mathbf{l}}$ with mesh size

$$h_{\mathbf{l}} := (2^{-l_1}, 2^{-l_2})$$
$$h := \min\{2^{-l_1}, 2^{-l_2}\}$$

For each $I_{\mathbf{l}}^{\mathbf{j}} = \{(x_1, x_2) : x_i \in (2^{-l_i} j_i, 2^{-l_i}(j_i + 1)]\}$, the traditional tensor-product polynomial space is

$$\mathbf{V}_{\mathbf{l}}^k = \{\mathbf{v} : \mathbf{v}(\mathbf{x}) \in P^k(I_{\mathbf{l}}^{\mathbf{j}}),\ \mathbf{0} \le \mathbf{j} \le 2^{\mathbf{l}} - \mathbf{1}\}$$

P^k denotes polynomial of degree at most k in each dimension. Uniform grid: $l_1 = l_2 = N$, $\mathbf{V}_{\mathbf{l}}^k = \mathbf{V}_N^k$, then

$$\mathbf{V}_N^k := V_{N,x_1}^k \times V_{N,x_2}^k = \bigoplus_{|\mathbf{l}|_\infty \le N} \mathbf{W}_{\mathbf{l}}^k$$

where

$$\mathbf{W}_{\mathbf{l}}^k := W_{l_1,x_1}^k \times W_{l_2,x_2}^k$$

The basis functions for $\mathbf{W}_{\mathbf{l}}^k$ can be defined by a tensor product

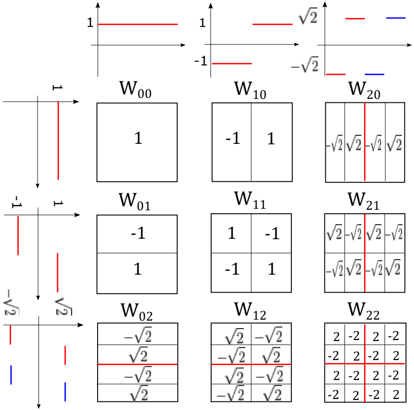
$$v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x}) := \prod_{t=1}^{2} v_{i_t,l_t}^{j_t}(x_t), \quad j_t = 0, \ldots, \max(0, 2^{l_t - 1} - 1), \quad i_t = 1, \ldots, k+1$$

Full grid approximation space

Full grid space:

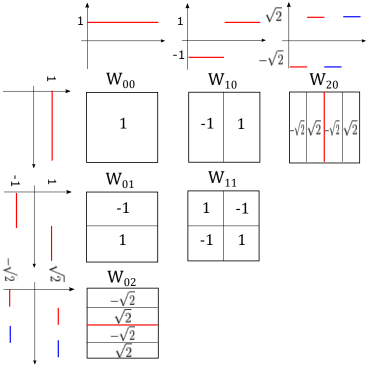
$$\mathbf{V}_N^k = \bigoplus_{|\mathbf{l}|_\infty \leq N} \mathbf{W}_\mathbf{l}^k$$

$d = 2,\ N = 2,\ k = 0$



Sparse grid approximation space

We consider the sparse grid space: $\hat{\mathbf{V}}_N^k := \bigoplus_{|\mathbf{l}|_1 \le N} \mathbf{W}_\mathbf{l}^k$



A viewpoint without using multiwavelet space: $\hat{\mathbf{V}}_N^k = \bigoplus_{|\mathbf{l}|_1 \le N} \mathbf{V}_\mathbf{l}^k$.

$$dim(\hat{\mathbf{V}}_N^k) = O(2^N N^{d-1}(k+1)^d) \quad \text{or} \quad O(h^{-1}|\log_2 h|^{d-1})$$

DG method on sparse grids: linear transport problems

Consider the linear transport equation with variable coefficient

$$\begin{cases} u_t + \nabla \cdot (\boldsymbol{\alpha}(\mathbf{x}, t)\, u) = 0, & \mathbf{x} \in \Omega = [0,1]^d, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), \end{cases} \tag{1}$$

The semi-discrete DG formulation for (1) is defined as follows: find $u_h \in \hat{\mathbf{V}}_N^k$, such that

$$\int_\Omega (u_h)_t\, v_h \, d\mathbf{x} = \int_\Omega u_h \boldsymbol{\alpha} \cdot \nabla v_h \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \widehat{\boldsymbol{\alpha} u_h} \cdot [v_h] \, ds, \tag{2}$$

$$\doteq A(u_h, v_h)$$

for $\forall\, v_h \in \hat{\mathbf{V}}_N^k$, where $\widehat{\boldsymbol{\alpha} u_h}$ defined on the element interface denotes a monotone numerical flux.

Stability (constant coefficient case)

Theorem (L^2 stability)

The DG scheme (2) for (1) is L^2 stable when $\boldsymbol{\alpha}$ is a constant vector, i.e.

$$\frac{d}{dt}\int_\Omega (u_h)^2\,d\mathbf{x} = -\sum_{e\in\Gamma}\int_e \frac{|\boldsymbol{\alpha}\cdot\mathbf{n}|}{2}|[u_h]|^2 ds \leq 0. \tag{3}$$

Error estimate (constant coefficient case)

Similar to Schwab, Suli, Todor (08), we can establish error estimate in L^2 norm for the L^2 projection operator, combining with an estimate for DG method, we get

Theorem (L^2 error estimate)

Let u be the exact solution, and u_h be the numerical solution to the semi-discrete scheme (2) *with numerical initial condition $u_h(0) = \mathbf{P} u_0$. For $k \geq 1$, $u_0 \in \mathcal{H}^{p+1}(\Omega)$, $1 \leq q \leq \min\{p, k\}$, $N \geq 1$, $d \geq 2$, we have for all $t \geq 0$,*

$$\|u_h - u\|_{L^2(\Omega_N)} \leq \left(2\sqrt{C_d \|\boldsymbol{\alpha}\|_2 t}\, C_\star(k, q, d, N) + (\bar{\bar{c}}_{k,0,q} + B_0(k, q, d)\kappa_0(k, q, N)^d)2^{-N/2}\right) 2^{-N(q+1/2)} |u_0|_{\mathcal{H}^{q+1}(\Omega)},$$

where C_d is a generic constant with dependence only on d, $C_\star(k, q, d, N) = \max_{s=0,1}\left(\bar{\bar{c}}_{k,s,q} + B_s(k, q, d)\kappa_s(k, q, N)^d\right)$. The constants $\bar{\bar{c}}_{k,s,q}$, $B_s(k, q, d)$, $\kappa_s(k, q, N)$ are defined in L^2 projection error estimates.

Convergence rate $O((\log h)^d h^{k+1/2})$.

Linear advection: sparse grid DG

We consider the following linear advection problem

$$\begin{cases} u_t + \sum_{m=1}^{d} u_{x_m} = 0, \quad \mathbf{x} \in [0,1]^d, \\ u(0,\mathbf{x}) = \sin\left(2\pi \sum_{m=1}^{d} x_m\right), \end{cases} \tag{4}$$

subject to periodic boundary conditions.
In the simulation, we compute the numerical solutions up to two periods in time, meaning that we let final time $T = 1$ for $d = 2$, $T = 2/3$ for $d = 3$, and $T = 0.5$ for $d = 4$.

Table: L^2 errors and orders of accuracy at $T = 1$ when $d = 2$, $T = 2/3$ when $d = 3$, and $T = 0.5$ when $d = 4$. N is the number of mesh levels, h_N is the size of the smallest mesh in each direction, k is the polynomial order, d is the dimension. DOF denotes the degrees of freedom of the sparse approximation space $\hat{V}_N^k$. L^2 order is calculated with respect to h_N.

N	h_N	DOF	L^2 error	order	DOF	L^2 error	order	DOF	L^2 error	order
		$k=1,\ d=2$			$k=1,\ d=3$			$k=1,\ d=4$		
4	1/16	192	9.17E-02	–	832	3.72E-01	–	3072	4.99E-01	–
5	1/32	448	1.90E-02	2.27	2176	1.19E-01	1.64	8832	2.40E-01	1.06
6	1/64	1024	4.81E-03	1.98	5504	2.96E-02	2.01	24320	9.84E-02	1.28
7	1/128	2304	1.27E-03	1.92	13568	8.85E-03	1.74	64768	3.21E-02	1.62
		$k=2,\ d=2$			$k=2,\ d=3$			$k=2,\ d=4$		
4	1/16	432	2.13E-03	–	2808	1.10E-02	–	15552	2.80E-02	–
5	1/32	1008	4.39E-04	2.28	7344	1.79E-03	2.63	44712	5.82E-03	2.27
6	1/64	2304	4.45E-05	3.30	18576	3.97E-04	2.17	123120	1.37E-03	2.09
7	1/128	5184	7.68E-06	2.54	45792	5.14E-05	2.95	327888	2.58E-04	2.41
		$k=3,\ d=2$			$k=3,\ d=3$			$k=3,\ d=4$		
3	1/8	320	6.36E-04	–	2432	2.10E-03	–	16128	4.09E-03	–
4	1/16	768	8.93E-05	2.83	6656	2.37E-04	3.14	49152	6.06E-04	2.75
5	1/32	1792	4.07E-06	4.46	17408	2.49E-05	3.25	141312	6.85E-05	3.14
6	1/64	4096	3.47E-07	3.55	44032	1.83E-06	3.76	389120	7.19E-06	3.25
7	1/128	9216	1.97E-08	4.14	108544	2.03E-07	3.18	1036288	6.36E-07	3.50

Adaptivity

To resolve fine local structures/accelerate the computation

- Adaptive wavelet methods.
- Adaptive DG methods.
- Adaptive sparse grid schemes. Zenger (90), Griebel (98), Bokanowksi et al. (12)...
- Multiresolution finite difference/finite volume methods for hyperbolic PDEs. Harten (95), Bihari, Harten (97), Dahmen et al. (01), Cohen et al. (03)
- Adaptive multiresolution DG schemes Calle et al. (2005), Archibald et al. (2011), Hovhannisyan et al. (2014), Gerhard et al. (2015)

Adaptive projection algorithm: parents and children

If a element $V_{\mathbf{l}'}^{\mathbf{j}'}$ satisfies the following conditions:

- There exists an integer m such that $1 \leq m \leq d$ and $\mathbf{l}' = \mathbf{l} + \mathbf{e}_m$, where $\mathbf{e}_m$ denotes the unit vector in x_m direction, and the support of $V_{\mathbf{l}'}^{\mathbf{j}'}$ is within the support of $V_{\mathbf{l}}^{\mathbf{j}}$.
- $|\mathbf{l}'|_\infty \leq N$,

then it is called a child element of $V_{\mathbf{l}}^{\mathbf{j}}$. Accordingly, element $V_{\mathbf{l}}^{\mathbf{j}}$ is called a parent element of $V_{\mathbf{l}'}^{\mathbf{j}'}$.

We use the hash table as the underlying data structure.

Refinement criteria

For a function $u(\mathbf{x}) \in \mathcal{H}^{p+1}(\Omega)$, we can show that $u(\mathbf{x}) = \sum_{\mathbf{l} \in \mathbb{N}_0^d} \sum_{\mathbf{j} \in B_{\mathbf{l}}, \mathbf{1} \le \mathbf{i} \le \mathbf{k}+\mathbf{1}} u_{\mathbf{i},\mathbf{l}}^{\mathbf{j}} v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x})$, where the hierarchical coefficient is $u_{\mathbf{i},\mathbf{l}}^{\mathbf{j}} = \int_\Omega u(\mathbf{x}) v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x}) d\mathbf{x}$.

An element $V_{\mathbf{l}}^{\mathbf{j}} := \{v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}, \mathbf{1} \le \mathbf{i} \le \mathbf{k}+\mathbf{1}\}$ is considered important if

$$\sum_{\mathbf{1} \le \mathbf{i} \le \mathbf{k}+\mathbf{1}} |u_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}| \| v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x}) \|_{L^1(\Omega)} > \varepsilon, \quad \text{if} \quad s = 1 \tag{5}$$

$$\left(\sum_{\mathbf{1} \le \mathbf{i} \le \mathbf{k}+\mathbf{1}} |u_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}|^2 \right)^{\frac{1}{2}} > \varepsilon, \quad \text{if} \quad s = 2 \tag{6}$$

$$\sum_{\mathbf{1} \le \mathbf{i} \le \mathbf{k}+\mathbf{1}} |u_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}| \| v_{\mathbf{i},\mathbf{l}}^{\mathbf{j}}(\mathbf{x}) \|_{L^\infty(\Omega)} > \varepsilon, \quad \text{if} \quad s = \infty, \tag{7}$$

where ε is a prescribed error threshold.

A similar coarsening criteria can be defined.

Adaptive evolution algorithm

Input: Hash table H and leaf table L at t^n, numerical solution $u_h^n \in \mathbf{V}_{N,H}^k$.
Parameters: Maximum level N, polynomial degree k, error constants ε, η, CFL constant.
Output: Hash table H and leaf table L at t^{n+1}, numerical solution $u_h^{n+1} \in \mathbf{V}_{N,H}^k$.

- **Prediction.** Given a hash table H that stores the numerical solution u_h at time step t^n, calculate Δt. Predict the solution by the DG scheme using space $\mathbf{V}_{N,H}^k$ and the forward Euler time stepping method. Generate the predicted solution $u_h^{(p)}$.

Adaptive evolution algorithm

- **Refinement.** Based on the predicted solution $u_h^{(p)}$, screen all elements in the hash table H. If for element $V_{\mathbf{l}}^{\mathbf{j}}$, the refining criteria hold, then add its children elements to H and L provided they are not added yet, and set the associated detail coefficients to zero. We also need to make sure that all the parent elements of the newly added element are in H (i.e., no "hole" is allowed in the hash table) and increase the number of children for all its parent elements by one. This step generates the updated hash table $H^{(p)}$ and leaf table $L^{(p)}$.

Adaptive evolution algorithm

- **Evolution.** Given the predicted table $H^{(p)}$ and the leaf table $L^{(p)}$, we evolve the solution from t^n to t^{n+1} by the DG scheme using space $\mathbf{V}^k_{N,H^{(p)}}$ and the third order Runge-Kutta time stepping method. This step generates the pre-coarsened numerical solution $\tilde{u}_h^{n+1}$.
- **Coarsening.** For each element in the leaf table, if the coarsening criteria hold, then remove the element from table $H^{(p)}$ and $L^{(p)}$. For each of its parent elements in $H^{(p)}$, we decrease the number of children by one. If the number becomes zero, i.e, the element has no child, then it will be added to leaf table $L^{(p)}$. Repeat the coarsening procedure until no element can be removed from the leaf list. Denote the resulting hash table and leaf table by H and L respectively, and the compressed numerical solution $u_h^{n+1} \in \mathbf{V}^k_{N,H}$.

Linear advection: adaptive sparse grid DG

We test the convergence of adaptive scheme with smooth initial $u(0,\mathbf{x}) = \prod_{m=1}^{d} \sin^4(\pi x_m)$.
For smooth case, we fix $N = 7$, and calculate

$$\text{convergence rate with respect to } \varepsilon \quad R_{\varepsilon_l} = \frac{\log(e_{l-1}/e_l)}{\log(\varepsilon_{l-1}/\varepsilon_l)}$$

$$\text{convergence rate with respect to DOF} \quad R_{\text{DOF}_l} = \frac{\log(e_{l-1}/e_l)}{\log(\text{DOF}_l/\text{DOF}_{l-1})},$$

Table: Numerical error and convergence rate. $N = 7$. $T = 1$. L^2 norm based criteria.

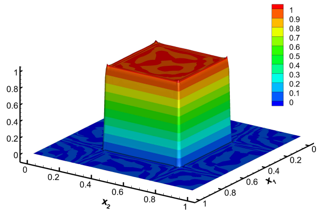
ε	DOF	L^2 error	R_{DOF}	R_ε	DOF	L^2 error	R_{DOF}	R_ε	DOF	L^2 error	R_{DOF}	R_ε
	$k = 1,\ d = 2$				$k = 1,\ d = 3$				$k = 1,\ d = 4$			
1E-03	312	1.47E-02			1168	2.62E-02			2592	2.87E-02		
5E-04	404	8.90E-03	1.93	0.72	1840	1.87E-02	0.75	0.49	4512	2.32E-02	0.39	0.31
1E-04	1148	1.70E-03	1.59	1.03	3920	7.26E-03	1.25	0.59	14976	9.49E-03	0.75	0.56
5E-05	1688	1.04E-03	1.28	0.71	6440	4.16E-03	1.12	0.80	23776	6.60E-03	0.79	0.53
1E-05	3588	2.42E-04	1.93	0.90	18624	8.83E-04	1.46	0.96	62368	2.13E-03	1.17	0.70
5E-06	4636	1.37E-04	2.23	0.82	25496	5.10E-04	1.75	0.79	111424	1.18E-03	1.02	0.86
	$k = 2,\ d = 2$				$k = 2,\ d = 3$				$k = 2,\ d = 4$			
5E-05	774	3.61E-04			4428	1.30E-03			26244	1.48E-03		
1E-05	1584	8.78E-05	1.97	0.88	9585	2.58E-04	2.10	1.01	51840	5.30E-04	1.51	0.64
5E-06	1998	4.58E-05	2.80	0.94	13716	1.74E-04	1.09	0.57	69012	2.60E-04	2.49	1.03
1E-06	4023	1.43E-05	1.67	0.73	27081	4.15E-05	2.11	0.89	168723	9.46E-05	1.13	0.63
5E-07	5157	7.20E-06	2.76	0.99	40446	2.45E-05	1.32	0.76	226719	4.89E-05	2.23	0.95
1E-07	9072	1.80E-06	2.46	0.86	77463	7.06E-06	1.91	0.77	531684	1.24E-05	1.61	0.85
	$k = 3,\ d = 2$				$k = 3,\ d = 3$				$k = 3,\ d = 4$			
1E-05	1120	3.71E-05			10496	5.72E-05			58368	1.26E-04		
5E-06	1184	2.92E-05	4.32	0.35	12032	4.91E-05	1.12	0.22	97280	7.53E-05	1.01	0.74
1E-06	2208	9.87E-06	1.74	0.67	18688	1.31E-05	3.00	0.82	129024	3.73E-05	2.49	0.44
5E-07	2864	4.85E-06	2.73	1.03	25984	1.09E-05	0.56	0.27	204800	1.34E-05	2.21	1.47
1E-07	3968	1.31E-06	4.02	0.82	43840	2.71E-06	2.66	0.86	409600	6.14E-06	1.13	0.49
5E-08	5760	7.88E-07	1.36	0.73	57472	1.50E-06	2.20	0.86	521216	2.79E-06	3.27	1.14

Linear advection: discontinuous profile

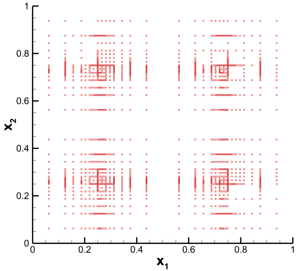
We consider

$$u(0,\mathbf{x}) = \begin{cases} 1 & (x_1, x_2) \in [\frac{1}{2} - \frac{\sqrt{6}}{2}, \frac{1}{2} + \frac{\sqrt{6}}{2}]^2. \\ 0 & \text{otherwise}, \end{cases} \quad (8)$$

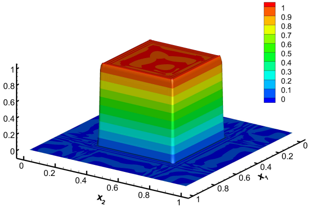
We fix $N = 7, \varepsilon = 10^{-5}$ and compare the performance of the scheme with L^1, L^2 and L^∞ based refinement/coarsening criteria up to final time $T = 1$.



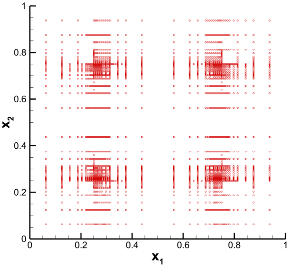
(a) L^1 criteria: solution



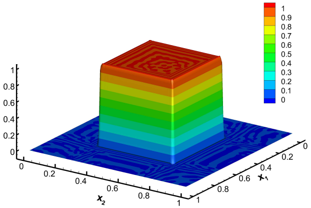
(b) L^1 criteria: active elements



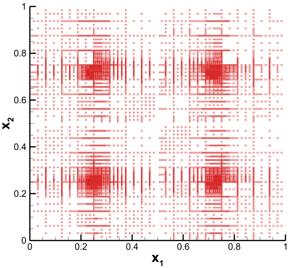
(c) L^2 criteria: solution



(d) L^2 criteria: active elements



(e) L^∞ criteria: solution



(f) L^∞ criteria: active elements

Outline

Nonlinear equations

Nonlinear equations pose simulation challenges. For example, we consider nonlinear conservation law

$$u_t + \nabla \cdot f(u) = 0, \tag{9}$$

The semi-discrete DG formulation is

$$\sum_K \int_K (u_h)_t v_h dx - \sum_K \int_K f(u_h) \cdot \nabla v_h dx + \sum_K \int_{\partial K} \hat{f}(u_h) \cdot n_K v_h ds = 0 \tag{10}$$

Replace terms like $f(u_h)$ by $\mathcal{I} f(u_h)$, where $\mathcal{I}$ is an interpolation operator corresponding to the (adaptive) sparse grid space.

Our work

We introduce a class of **high order local** hierarchical interpolating basis using the following steps:

- locating nested interpolation points, finding associated multiwavelet bases in 1D
- using Smolyak's idea to gain sparsity in high dimensions
- Fast transforms between point values and coefficients are introduced with operation counts of $O(d \cdot \mathrm{DoF})$ even for adaptive algorithms.

We should take into account accuracy and stability when designing the interpolation.

1D: nested points

Consider the domain $I = [0, 1]$, we use the same notation.In addition, we define $k + 1$ distinct points on each cell

$$x^j_{i,n} = 2^{-n} j + 2^{-n} \alpha_i \tag{11}$$

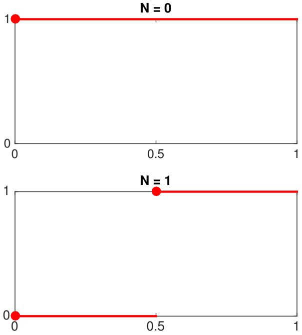
with $\alpha_i \in [0, 1]$, $i = 1, \dots, k + 1$.
In particular, the collection of those points $X^k_n = \{x^j_{i,n}\}$ is called *nested points*, if

$$X^k_0 \subset X^k_1 \subset X^k_2 \subset \cdots . \tag{12}$$

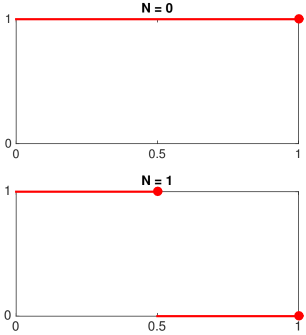
1D - Examples

P^0 case: nested points

- Case 1: $x_0 = 0$;
- Case 2: $x_0 = 1$;



(g) P^0: choice 1.



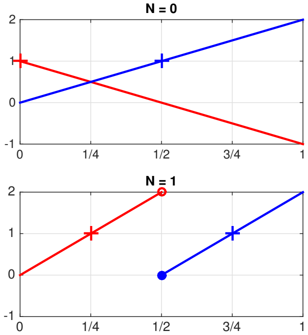
(h) P^0: choice 2.

1D-Example

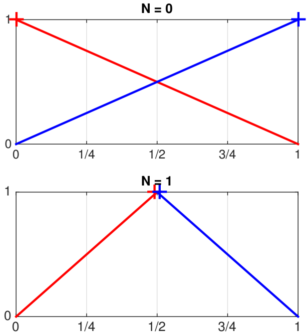
P^1 case:

- Case 1: $x_0 = 0$, $x_1 = 1/2$;
- Case 2: $x_0 = 0$, $x_1 = 1$;
- Case 3: $x_0 = 1/3$, $x_1 = 2/3$;
- Case 4: $x_0 = 1/2$, $x_1 = 1$;

1D-Example



(a) P^1: choice 1.



(b) P^1: choice 2.

Figure: Interpolation points: P^1.

Similarly, we can construct bases based on Hermite interpolation.

1D

Since $\{X_n^k\}$ are nested, the points can be rearranged in such a way that

$$X_n^k = X_0^k \cup \widetilde{X}_1^k \cup \cdots \cup \widetilde{X}_n^k, \quad \text{with } \widetilde{X}_n^k = X_n^k / X_{n-1}^k. \tag{13}$$

Moreover, we can now define the subspace W_n^k, $n \geq 1$, as the complement of V_{n-1}^k in V_n^k, in which the piecewise polynomials vanish at all points in X_{n-1}^k,

$$V_n^k = V_{n-1}^k \oplus W_n^k. \tag{14}$$

Thus, we have

$$V_N^k = \bigoplus_{0 \leq n \leq N} W_n^k.$$

1D

We now illustrate the computation of the multiwavelet coefficients based on interpolation. For a given function $f(x) \in C^{k+1}([0,1])$, we define $\mathcal{I}_n^k[f]$ as the standard interpolation on V_n^k. Next, we introduce the increment interpolation operator

$$\widetilde{\mathcal{I}}_n^k := \begin{cases} \mathcal{I}_n^k - \mathcal{I}_{n-1}^k, & n \geq 1 \\ \mathcal{I}_0^k, & n = 0. \end{cases} \tag{15}$$

Then, the interpolation operator $\mathcal{I}_N^k$ can be represented as

$$\mathcal{I}_N^k[f](x) = \sum_{n=0}^{N} \widetilde{\mathcal{I}}_n^k[f](x) = \sum_{n=0}^{N} \sum_{j=0}^{\max(2^{n-1}-1,0)} \sum_{i=1}^{k+1} b_{i,n}^j \varphi_{i,n}^j(x) \tag{16}$$

1D

We can define an operator $\mathcal{F}^{-1}$ mapping from point values $f(x^j_{i,n})$ to hierarchical coefficients $b^j_{i,n}$

$$b^j_{i,n} = \widetilde{\mathcal{I}}^k_n[f](x^j_{i,n}) = \mathcal{F}^{-1}[f] = \begin{cases} f(x^0_{i,0}), & n = 0, \\ f(\tilde{x}^j_{i,n}) - \sum_{l=1}^{k+1} f(x^j_{l,n-1})\phi_l(\tilde{x}_i), & n \geq 1. \end{cases} \tag{17}$$

and similarly

$$f(\tilde{x}^j_{i,n}) = \mathcal{F}[b] = \begin{cases} b^0_{i,0}, & n = 0, \\ b^j_{i,n} + \sum_{l=1}^{k+1} f_h(x^j_{l,n-1})\phi_l(\tilde{x}_i), & n \geq 1, \end{cases} \tag{18}$$

Summary

- This procedure works for arbitrary order, and include the continuous FEM case.
- We can switch from Lagrange to Hermite interpolation as long as the points are nested. This can help construct, e.g., C^1 FEM etc.
- For multi-D, if we use $\hat{\mathbf{V}}_N^k := \bigoplus_{|\mathbf{l}|_1 \leq N} \mathbf{W}_\mathbf{l}^k$, this gives a standard sparse grid method.
- Adaptivity can be incorporated based on thresholding.
- Fast transforms between point values and coefficients are introduced with operation counts of $O(d \cdot \mathrm{DoF})$ by method in Shen, Yu (10, 12).

Choice of interpolation

- For nonlinear conservation laws: we use quadrature with one more degree of accuracy.
- Another consideration is stability: based on numerical experiments, we found the Hermite interpolation is stable, while Lagrangian interpolation is not.

Artificial viscosity

For capturing shock, we add artificial viscosity

$$\sum_K \int_K (u_h)_t v_h d\mathbf{x} - \sum_K \int_K f(u_h)\cdot\nabla v_h d\mathbf{x} + \sum_K \int_{\partial K} \hat{f}(u_h)\cdot n_K v_h ds - \sum_K \int_K \nu(u_h)\nabla u_h\cdot\nabla v_h d\mathbf{x} = 0 \tag{19}$$

where $\nu = \nu(u_h) > 0$ is artificial viscosity depending on u_h. The artificial viscosity is only imposed in the leaf element and is determined in the following approach:

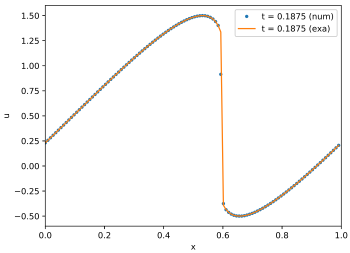
$$\nu = \begin{cases} 0, & \text{if} \quad s_e \leq s_0 + \kappa, \\ \nu_0 h, & \text{otherwise.} \end{cases}$$

where $\nu_0 > 0$ and κ are constants chosen empirically. In the computation, we typically take $\nu_0 = 1$ and $\kappa = 0$. The parameters s_e and s_0 are given as

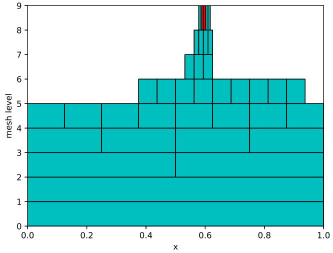
$$s_e = \log_{10}\left(\sum_{\mathbf{1}\leq\mathbf{i}\leq\mathbf{k}+\mathbf{1}} |u^{\mathbf{j}}_{\mathbf{i},\mathbf{l}}|^2 \right)^{\frac{1}{2}}, \quad s_0 = \log_{10}(2^{-(k+1)|\mathbf{l}|_1}). \tag{20}$$

For smooth regions, s_e should be the same order as s_0. In the discontinuous regions, s_e should be much larger than s_0.

Numerical results: 1D Burgers' equation



(a) solution



(b) elements

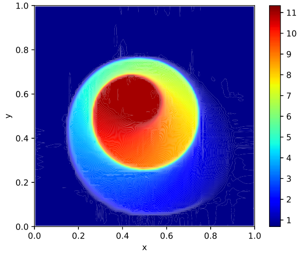
Figure: $t = 0.1875$. $N = 8$ and $\epsilon = 10^{-4}$. $N = 9, k = 2, P^3$ Hermite interpolation. red: elements with artificial viscosity

Numerical results: 2D KPP rotating wave problem

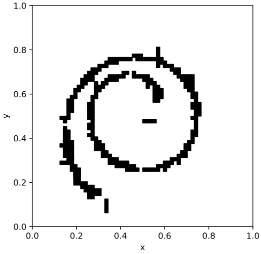
$$u_t + \sin(u)_x + \cos(u)_y = 0.$$

The initial condition is

$$u_0(x,y) = \begin{cases} 3.5\pi, & (x-1/2)^2 + (y-1/2)^2 \le \dfrac{1}{16}, \\ 0.25\pi, & \text{otherwise.} \end{cases}$$



(a) solution

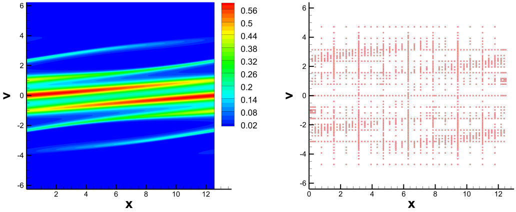


(b) elements with artificial viscosity

Outline

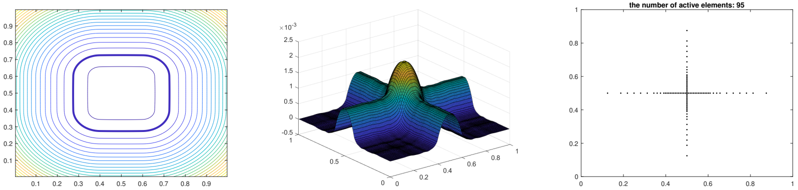
Kinetic equations

Vlasov-Poisson/Vlasov-Maxwell up to 4D. Example: Landau damping
$t = 10$



Hamilton-Jacobi equations

HJ/HJB equations (with LDG solver) up to 4D.

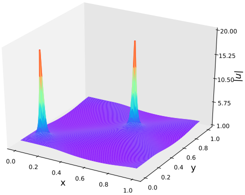


(c) (d) (e)

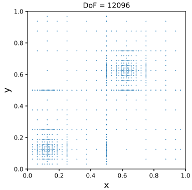
Figure: Example HJB. $T = 0.1$. $k = 2$, $M = 4$. $N = 7$. ϵ=10^{-7}. (a) Contour plot of the numerical solution. (b) Numerical error distribution. (c) Active elements.

NLS equations

NLS (with IPDG solver). 2D NLS



(a)

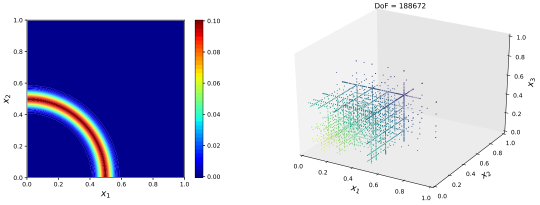


(b)

Figure: Example NLS. $t = 0$ and 1.5813. $N = 7, k = 3, \epsilon = 10^{-4}$ and $\eta = 10^{-5}$. (a) Numerical solution (b) Active elements.

Wave equations

Wave equation (with IPDG solver). 3D expanding wave in homogeneous medium.



(a) (b)

Figure: Expanding wave in homogeneous medium in 3D at $t = 0.5$. $N = 7$ and $\epsilon = 10^{-4}$ (a) numerical solution cut in 2D along $x_3 = 0$ (b) Active elements.

Outline

Conclusions

We design efficient & highly accurate numerical schemes for moderately high dimensional PDEs.

- DG methods: excellent for transport problems.
- Sparse grid DG methods: works well for smooth solutions. Stability and convergence properties can be well understood theoretically.
- Adaptivity is naturally incorporated.
- The schemes can be applied to a large class of PDEs.
- **Source code** `https://github.com/JuntaoHuang/adaptive-multiresolution-DG`

Some Reference

- Z. Wang, Q. Tang, W. Guo and Y. Cheng, Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations, *Journal of Computational Physics*, v314 (2016), pp. 244-263.
- W. Guo and Y. Cheng, A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations, *SIAM Journal on Scientific Computing*, v38 (2016), pp. A3381-A3409.
- W. Guo and Y. Cheng, An adaptive multiresolution discontinuous Galerkin method for time-dependent transport equations in multi-dimensions, *SIAM Journal on Scientific Computing*, v39 (2017), pp. A2962-A2992.
- Z. Tao, W. Guo and Y. Cheng, Sparse grid discontinuous Galerkin methods for Vlasov-Maxwell systems, *Journal of Computational Physics:X*, v3 (2019), 100022.
- Z. Tao, Y. Jiang and Y. Cheng, An adaptive high-order piecewise polynomial based sparse grid collocation method with applications, *Journal of Computational Physics*, 2019.
- J. Huang and Y. Cheng, An adaptive multiresolution discontinuous Galerkin method with artificial viscosity for scalar hyperbolic conservation laws in multidimensions, *SIAM Journal on Scientific Computing*, 2019.
- J. Huang, Y. Liu, W. Guo, Z. Tao and Y. Cheng, An adaptive multiresolution interior penalty discontinuous Galerkin method for wave equations in second order form, arxiv, 2020.
- W. Guo, J. Huang, Z. Tao and Y. Cheng, An adaptive sparse grid local discontinuous Galerkin method for Hamilton-Jacobi equations in high dimensions, arxiv, 2020.
- Z. Tao, J. Huang, Y. Liu, W. Guo and Y. Cheng, An adaptive multiresolution ultra-weak discontinuous Galerkin method for nonlinear Schrödinger equations, arxiv, 2020.

The END!

Thank You!