Discretization by Machine Learning

The Finite-Element with Discontiguous Support Method

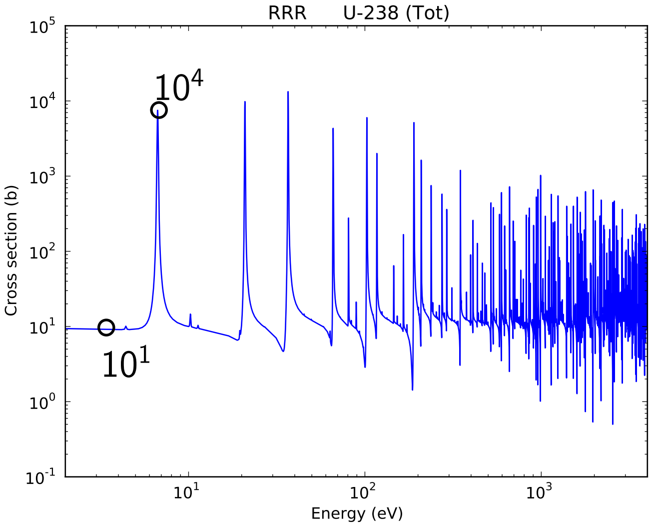
Andrew T. Till

Nicholas C. Metropolis Postdoc Fellow
Computational Physics and Methods (CCS-2)
Los Alamos National Laboratory
till@lanl.gov

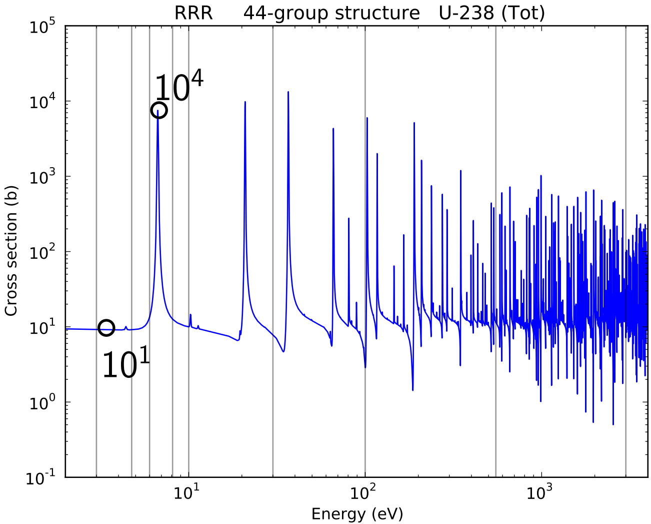
July 27, 2016
More at http://goo.gl/HMWtu
LA-UR-16-25320

How would you represent this image efficiently?

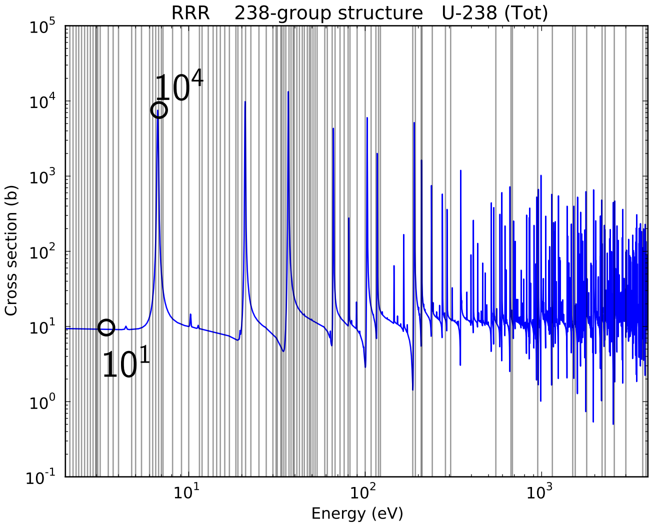
How would you represent this data efficiently?



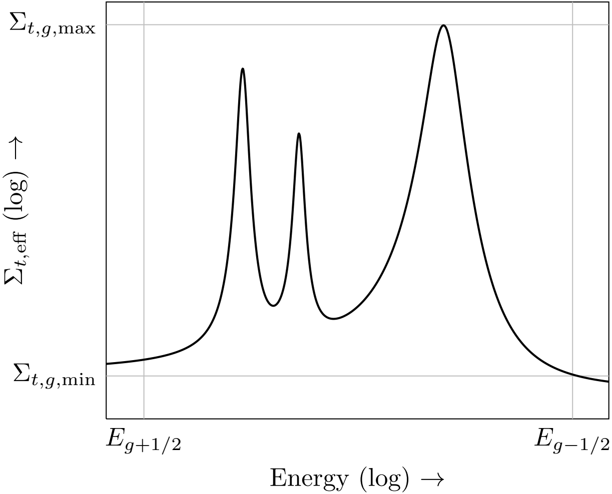
Partitioning in energy requires many unknowns to resolve the resonances



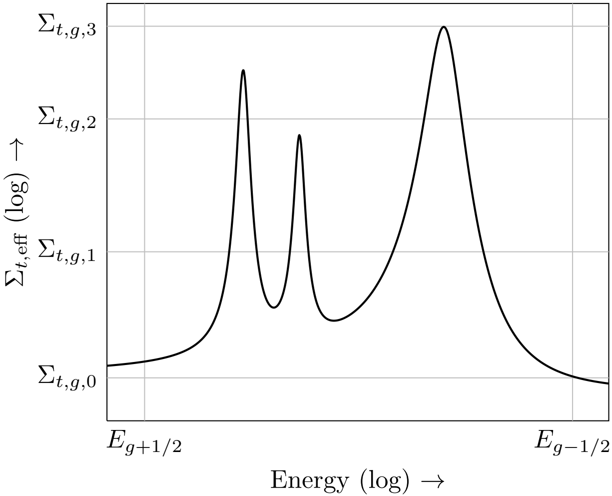
Partitioning in energy requires many unknowns to resolve the resonances



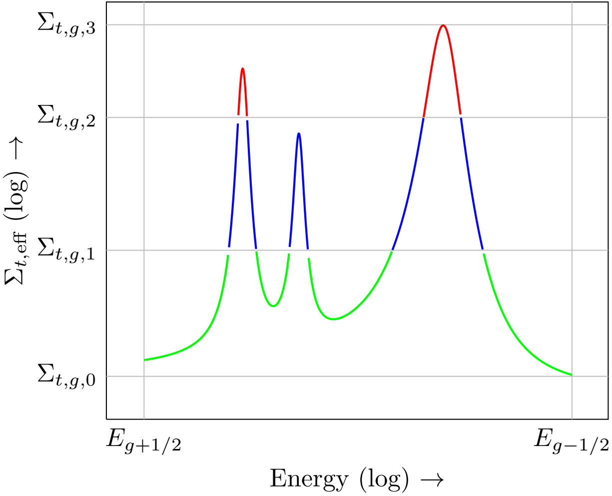
Partitioning in the dependent variable requires fewer unknowns, but results in a non-contiguous partitioning in energy



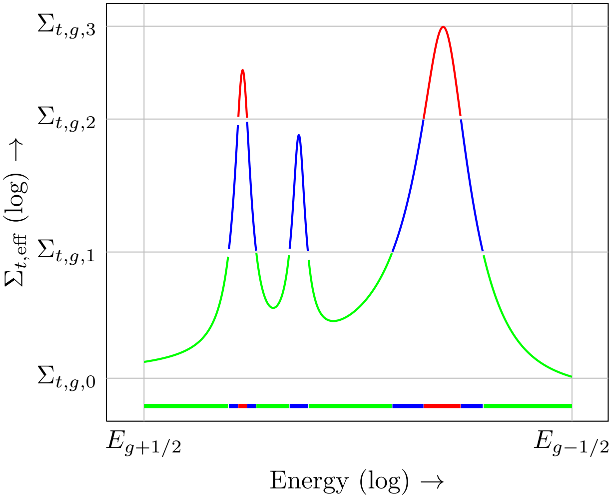
Partitioning in the dependent variable requires fewer unknowns, but results in a non-contiguous partitioning in energy



Partitioning in the dependent variable requires fewer unknowns, but results in a non-contiguous partitioning in energy



Partitioning in the dependent variable requires fewer unknowns, but results in a non-contiguous partitioning in energy

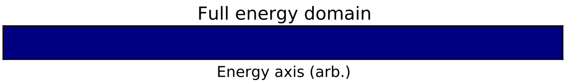


Non-contiguous discretizations can be advantageous

This talk will be on the Finite-Element with Discontiguous Support (FEDS) method

FEDS

- Preserves fine-scale features with few unknowns
- Uses machine learning to determine non-contiguous grid
- Applied to nuclear reactor analysis

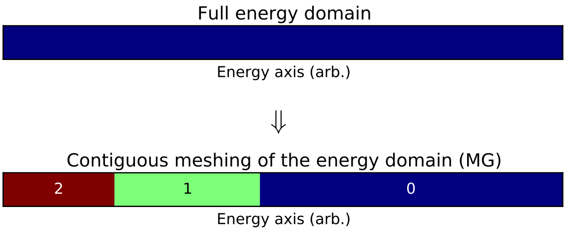


Non-contiguous discretizations can be advantageous

This talk will be on the Finite-Element with Discontiguous Support (FEDS) method

FEDS

- Preserves fine-scale features with few unknowns
- Uses machine learning to determine non-contiguous grid
- Applied to nuclear reactor analysis

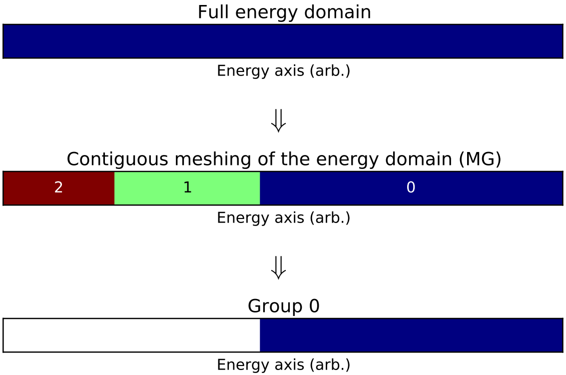


Non-contiguous discretizations can be advantageous

This talk will be on the Finite-Element with Discontiguous Support (FEDS) method

FEDS

- Preserves fine-scale features with few unknowns
- Uses machine learning to determine non-contiguous grid
- Applied to nuclear reactor analysis

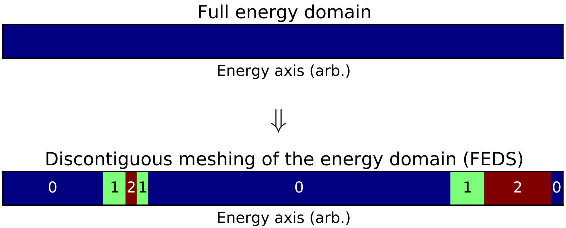


Non-contiguous discretizations can be advantageous

This talk will be on the Finite-Element with Discontiguous Support (FEDS) method

FEDS

- Preserves fine-scale features with few unknowns
- Uses machine learning to determine non-contiguous grid
- Applied to nuclear reactor analysis

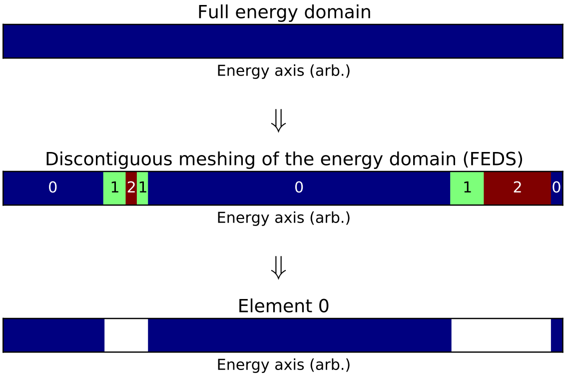


Non-contiguous discretizations can be advantageous

This talk will be on the Finite-Element with Discontiguous Support (FEDS) method

FEDS

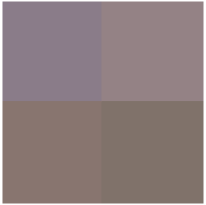
- Preserves fine-scale features with few unknowns
- Uses machine learning to determine non-contiguous grid
- Applied to nuclear reactor analysis



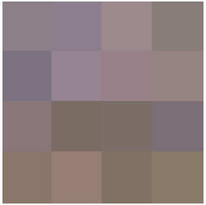
Let us compress this image using non-contiguous unknowns

Traditional discretization schemes correspond to smearing / averaging the image

1×1



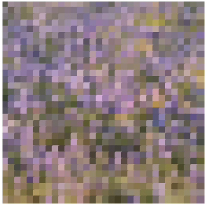
2×2



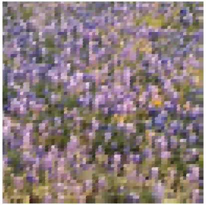
4×4

8×8

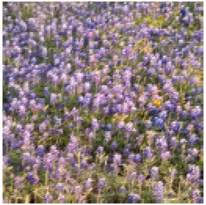
16×16



32×32



64×64



128×128

256×256

512×512

FEDS corresponds to clustering the image, which we do here by clustering like colors


1 color


2 colors


4 colors


8 colors


16 colors


32 colors

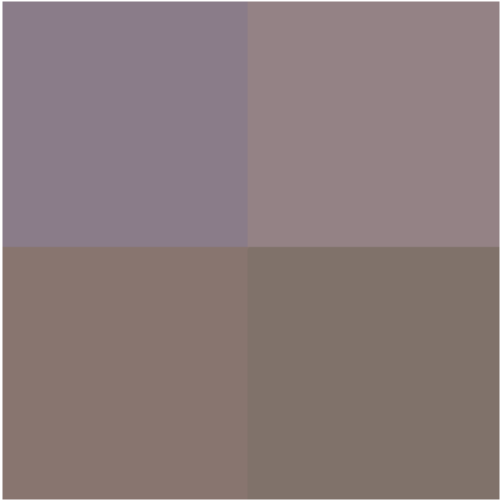

64 colors


128 colors


Original

Clustering preserves underlying physical structure while averaging does not

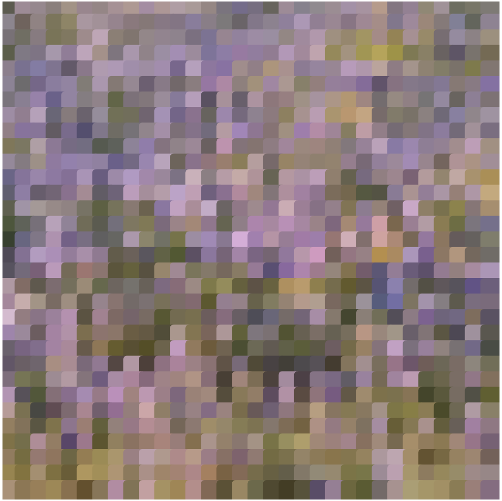
2 colors



2×2 pixels

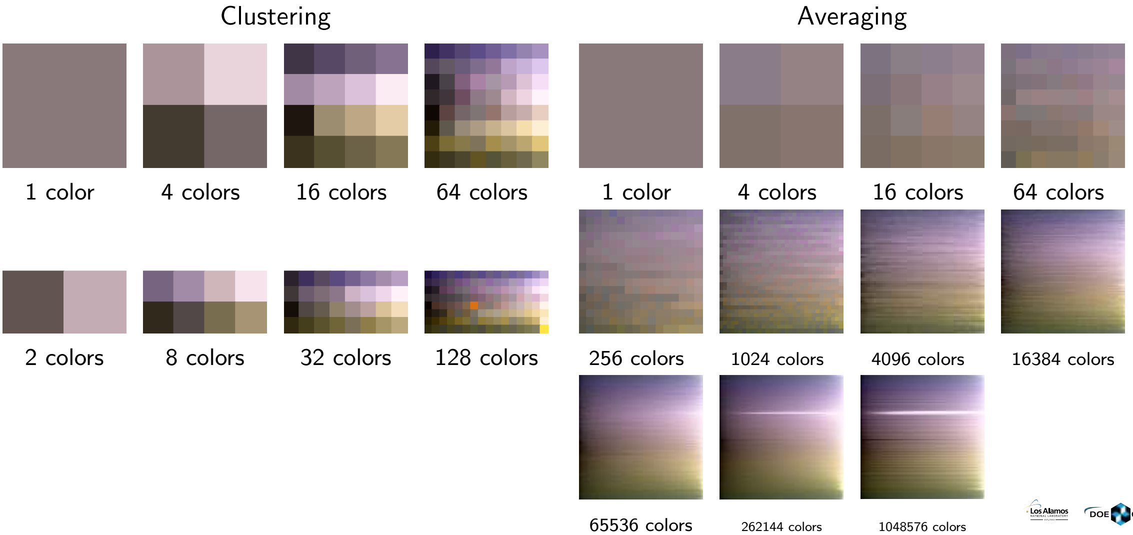
Clustering preserves underlying physical structure while averaging does not

8 colors



32×32 pixels

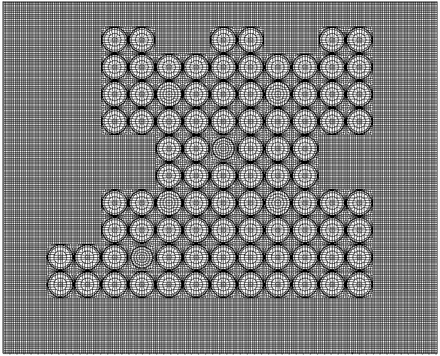
Clustering requires substantially less data when only the colors need to be stored, not their locations



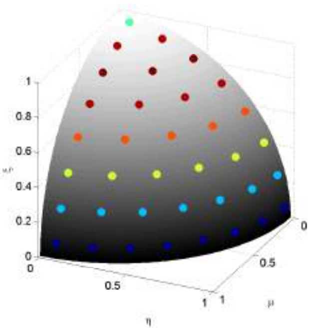
Nuclear reactor analysis involves solving the six-dimensional Boltzmann transport equation

Unknown is $\psi(\mathbf{r}, \mathbf{\Omega}, E)$

- $\mathbf{r}$ is space (3D: x, y, z), $\mathbf{\Omega}$ is neutron direction (2D), E is neutron kinetic energy (1D)



Spatial mesh

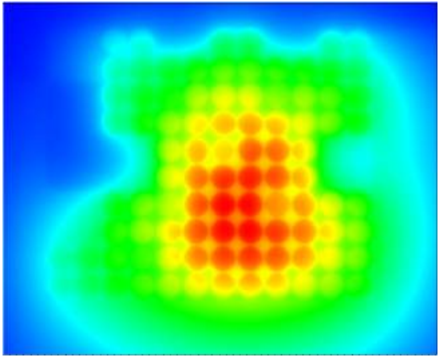


Directional mesh

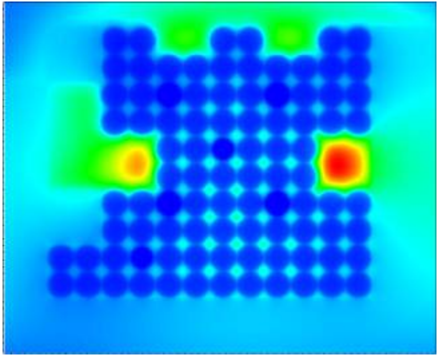
Nuclear reactor analysis involves solving the six-dimensional Boltzmann transport equation

Unknown is $\psi(\mathbf{r}, \mathbf{\Omega}, E)$

- $\mathbf{r}$ is space (3D: x, y, z), $\mathbf{\Omega}$ is neutron direction (2D), E is neutron kinetic energy (1D)



Fast neutrons' flux



Slow neutrons' flux

Consider an example advection-reaction problem

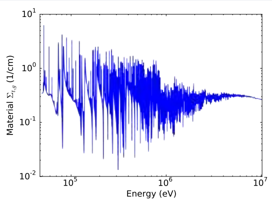
- Unknown solution is $\psi(\mathbf{x}, t, E)$ and satisfies
  $\left[\frac{\partial}{\partial t} + \nabla \cdot \mathbf{v}(E) + \sigma(\mathbf{x}, t, E)\right] \psi(\mathbf{x}, t, E) = q(\mathbf{x}, t, E)$
- Has fine-scale dependence on variable E
  (More generally, E **could be anything that has microstructure**)

Consider an example advection-reaction problem

- Unknown solution is $\psi(\mathbf{x}, t, E)$ and satisfies
  $\left[\frac{\partial}{\partial t} + \nabla \cdot \mathbf{v}(E) + \sigma(\mathbf{x}, t, E)\right] \psi(\mathbf{x}, t, E) = q(\mathbf{x}, t, E)$
- Has fine-scale dependence on variable E
  (More generally, E **could be anything that has microstructure**)

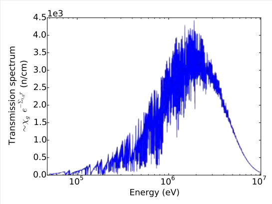
Fine structure in coefficients

$\sigma(E)$ vs. E



Causes fine structure in solution

$\psi(\mathbf{x}_R, t, E)$ vs. E

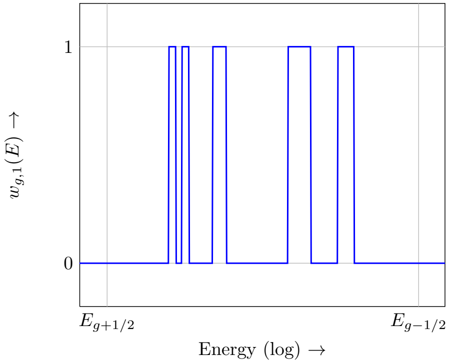


Our method is the Finite-Element with Discontiguous Support (FEDS) method

FEDS is a Petrov-Galerkin finite element method

This is our *only* approximation

$$\psi_{\text{exact}}(\mathbf{x}, t, E) \simeq \psi_{\text{FEDS}}(\mathbf{x}, t, E) \equiv \sum_k b_k(\mathbf{x}, E)\Psi_k(\mathbf{x}, t),$$



Our weight functions (left)

$$w_k(E) = \begin{cases} 1 & \text{if } E \in \mathbb{E}_k, \\ 0 & \text{otherwise,} \end{cases}$$

Our basis functions

$$b_k(\mathbf{x} \in V_i, E) = \begin{cases} C_{i,k}\, f_i(E) & E \in \mathbb{E}_k, \\ 0 & \text{otherwise,} \end{cases}$$

where the $f_i(E)$ have fine-scale features and $C_{i,k}$ normalizes

Apply FEDS to our example advection-reaction problem

We seek a weak form of the equations in energy

1. Multiply by weight function and integrate over all E

   $$\int_0^\infty \mathrm{d}E\, w_j(E) \left\{ \left[\frac{\partial}{\partial t} + \nabla \cdot \mathbf{v}(E) + \sigma(\mathbf{x}, t, E) \right] \psi(\mathbf{x}, t, E) - q(\mathbf{x}, t, E) \right\} = 0$$

2. Expand ψ into basis function representation (our only approximation) and group terms
3. Use orthonormality of weight and basis functions

   $$\int_0^\infty \mathrm{d}E\, w_j(E) b_k(\mathbf{x}, \mathbf{E}) = \delta_{j,k}\ \forall \mathbf{x}$$

4. Get weak form

   $$\left[\frac{\partial}{\partial t} + \nabla \cdot \mathbf{v}_{i,k} + \sigma_k(\mathbf{x}, t) \right] \Psi_k(\mathbf{x}, t) = q_k(\mathbf{x}, t),$$

Apply FEDS to our example advection-reaction problem

We seek a weak form of the equations in energy

1. Multiply by weight function and integrate over all E

   $$\int_0^\infty \mathrm{d}E\, w_j(E)\left\{\left[\frac{\partial}{\partial t} + \nabla\cdot\mathbf{v}(E) + \sigma(\mathbf{x},t,E)\right]\psi(\mathbf{x},t,E) - q(\mathbf{x},t,E)\right\} = 0$$

2. Expand ψ into basis function representation (our only approximation) and group terms
3. Use orthonormality of weight and basis functions

   $$\int_0^\infty \mathrm{d}E\, w_j(E) b_k(\mathbf{x},\mathbf{E}) = \delta_{j,k}\ \forall \mathbf{x}$$

4. Get weak form

   $$\left[\frac{\partial}{\partial t} + \nabla\cdot\mathbf{v}_{i,k} + \sigma_k(\mathbf{x},t)\right]\Psi_k(\mathbf{x},t) = q_k(\mathbf{x},t),$$

   with coefficients

   $$\mathbf{v}_{i,k} = C_{i,k}\int_{\mathbb{E}_k} \mathrm{d}E\, \mathbf{v}(E) f_i(E),\ \mathbf{x}\in V_i,$$
   $$\sigma_k(\mathbf{x},t) = C_{i,k}\int_{\mathbb{E}_k} \mathrm{d}E\, \sigma(\mathbf{x},t,E) f_i(E),\ \mathbf{x}\in V_i,$$
   $$q_k(\mathbf{x},t) = \int_{\mathbb{E}_k} \mathrm{d}E\, q_k(\mathbf{x},t).$$

Solve for $\Psi_k(\mathbf{x},t)$, which determines $\psi(\mathbf{x},t,E) = \sum_k b_k(\mathbf{x},E)\Psi_k(\mathbf{x},t)$.

Our finite element method has two free parameters

We must choose the $f_i(E)$ (shape of basis functions)

We must choose the $\mathbb{E}_k$ (support of weight/basis functions)

Our finite element method has two free parameters

We must choose the $f_i(E)$ (shape of basis functions)

- Reasonable first approximation is $f_i(E) = 1$
- Using an analytic model or subproblem to compute $f_i(E)$ has been found to decrease solution error constant, but not rate of convergence

We must choose the $\mathbb{E}_k$ (support of weight/basis functions)

Our finite element method has two free parameters

We must choose the $f_i(E)$ (shape of basis functions)

- Reasonable first approximation is $f_i(E) = 1$
- Using an analytic model or subproblem to compute $f_i(E)$ has been found to decrease solution error constant, but not rate of convergence

We must choose the $\mathbb{E}_k$ (support of weight/basis functions)

- Should have the property that:
  If E_1 and E_2 are both in the same $\mathbb{E}_k$,
  then $\psi(\mathbf{x}, t, E_1)$ and $\psi(\mathbf{x}, t, E_2)$ should have **similar behavior** in $(\mathbf{x}, t)$

Our finite element method has two free parameters

We must choose the $f_i(E)$ (shape of basis functions)

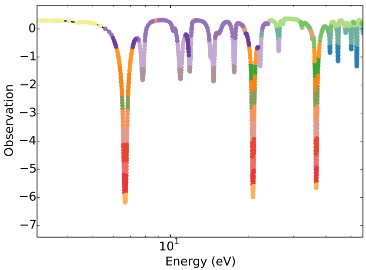
- Reasonable first approximation is $f_i(E) = 1$
- Using an analytic model or subproblem to compute $f_i(E)$ has been found to decrease solution error constant, but not rate of convergence

We must choose the $\mathbb{E}_k$ (support of weight/basis functions)

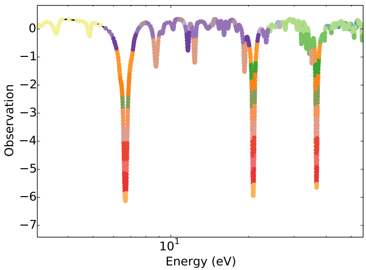
- Should have the property that:
  If E_1 and E_2 are both in the same $\mathbb{E}_k$,
  then $\psi(\mathbf{x}, t, E_1)$ and $\psi(\mathbf{x}, t, E_2)$ should have **similar behavior** in $(\mathbf{x}, t)$ (The ψ may have different behavior if E_1 and E_2 are in different $\mathbb{E}_k$)
- This requires us to know **how the solution behaves on the fine scale** (if only **approximately**)

We use machine learning to determine the $\mathbb{E}_k$

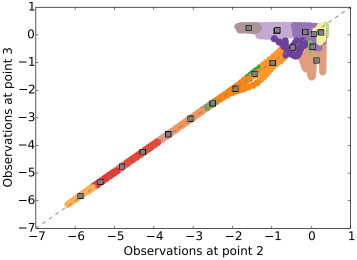
Snapshot of MOX spectrum:



Snapshot of UO_2 spectrum:

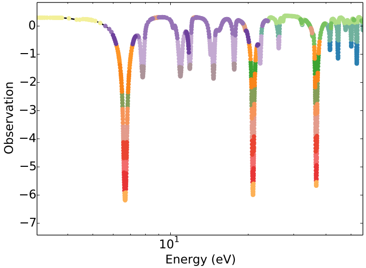


Clustering algorithm view:

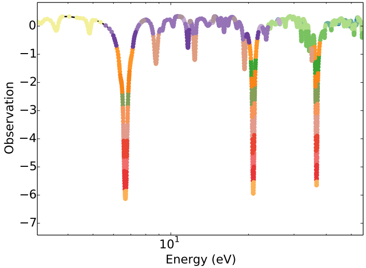


We use machine learning to determine the $\mathbb{E}_k$

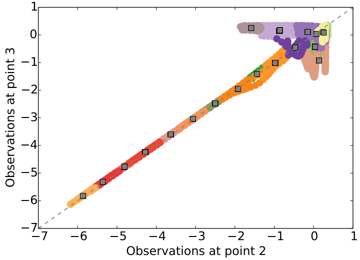
Snapshot of MOX spectrum:



Snapshot of UO_2 spectrum:



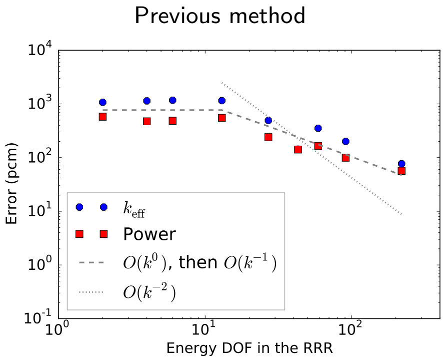
Clustering algorithm view:

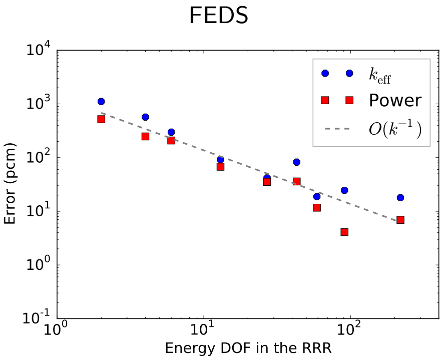


Our process

- We assume we have been given solution-like spectra: $s_i(E_n) \simeq \psi(\mathbf{x}_i, t_i, E_n)$ for representative locations, i, and on a fine grid, n
- We use hierarchical agglomeration to **combine into the same** $\mathbb{E}_k$ **those** E_n **whose** $s_i(E_n)$ **behave similarly** $\forall i$ (above, right)

We find FEDS to be convergent and able to achieve low errors with modest unknown counts





FEDS

- Often first-order convergent
- Uses a non-standard finite element method whose elements have discontiguous support
- Uses machine learning to calculate this support

This work would not have been possible without Krell and the CSGF

Which provided the intellectual freedom to pursue new, interesting problems these past four years

Thank You!

This work was funded by the Department of Energy Computational Science Graduate Fellowship program (DOE CSGF - grant number DE-FG02-97ER25308), which provides strong support to its fellows and their professional development.
LA-UR-16-25320

DOE CSGF

Los Alamos NATIONAL LABORATORY EST. 1943