Novel Meshes for Multivariate Interpolation and Approximation

Thomas C. H. Lux, Layne T. Watson, Tyler H. Chang, Jon Bernard, Bo Li, Xiaodong Yu, Li Xu, Godmar Back, Ali R. Butt, Kirk W. Cameron, Yili Hong, Danfeng Yao

Virginia Polytechnic Institute and State University

VIRGINIA TECH™

This work was supported by the National Science Foundation Grant CNS-1565314.

Motivation

Regression and interpolation are problems of considerable importance that find applications across many fields of science.

Motivation

Regression and interpolation are problems of considerable importance that find applications across many fields of science.

- Pollution and air quality analysis
- Energy consumption management
- Student performance prediction

Motivation

Regression and interpolation are problems of considerable importance that find applications across many fields of science.

Pollution and air quality analysis
Energy consumption management
Student performance prediction

These techniques are applied here to:

Motivation

Regression and interpolation are problems of considerable importance that find applications across many fields of science.

- Pollution and air quality analysis
- Energy consumption management
- Student performance prediction

These techniques are applied here to:

- High performance computing file input/output (HPC I/O)
- Parkinson's patient clinical evaluations
- Forest fire risk assessment

VIRGINIA TECH.

Problem Formulation

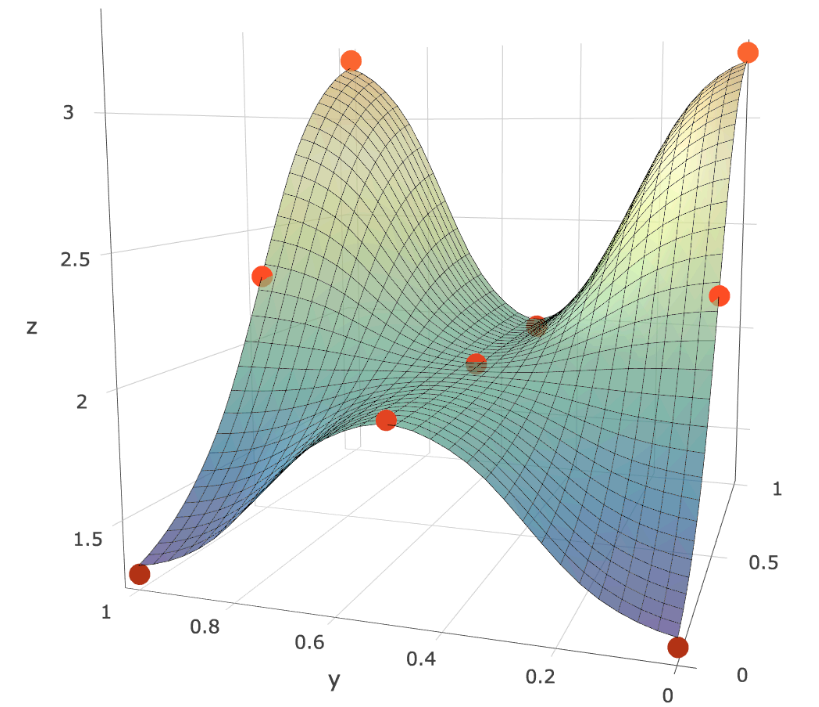
Given

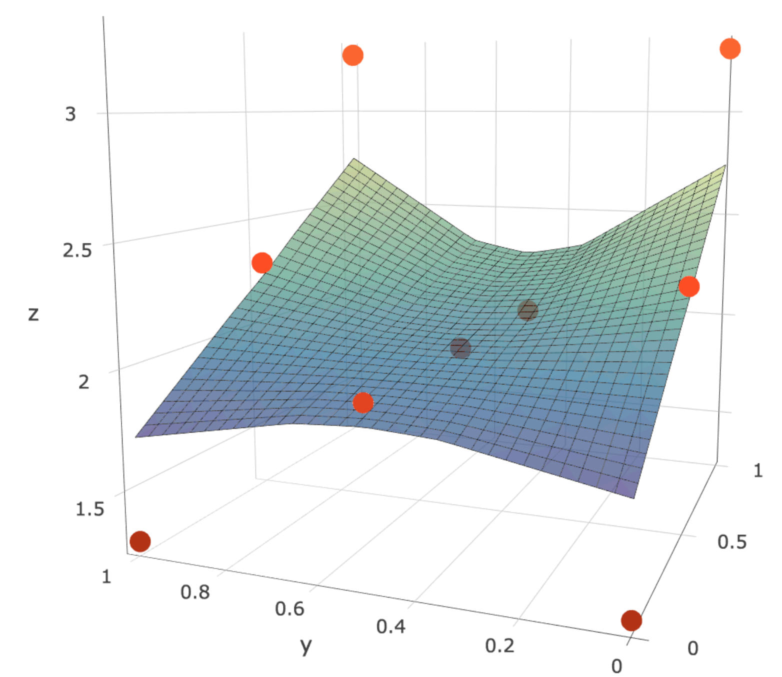
underlying function $f : \mathbb{R}^{d} \rightarrow \mathbb{R}$

data matrix $X^{n \times d}$ with row vectors $x^{(i)} \in \mathbb{R}^{d}$

response values $f(x^{(i)})$ for all $x^{(i)}$

matrix $f(X)$ has rows $f(x^{(i)})$





VIRGINIA TECH

Problem Formulation

Given

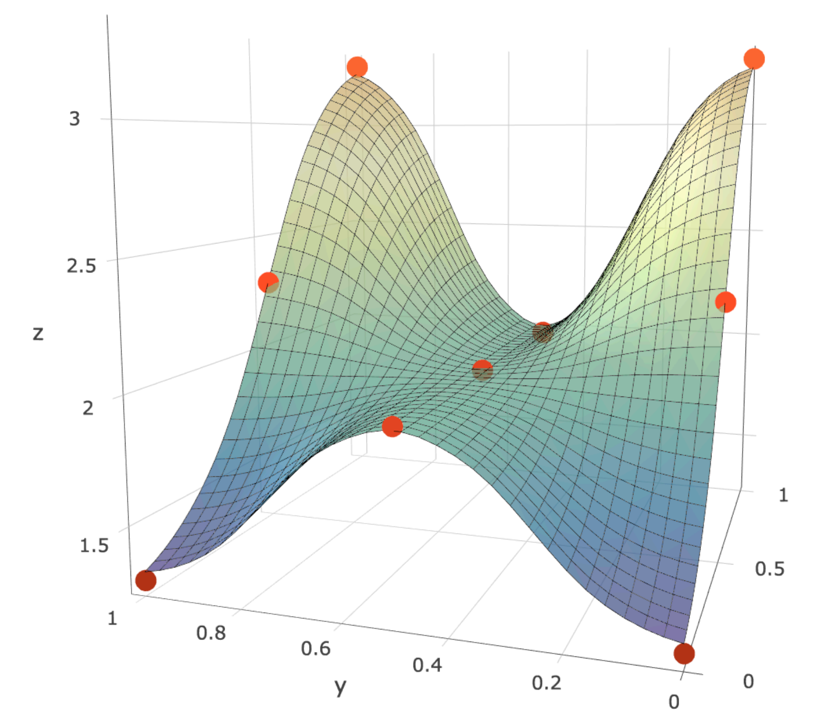
underlying function $f : \mathbb{R}^d \rightarrow \mathbb{R}$

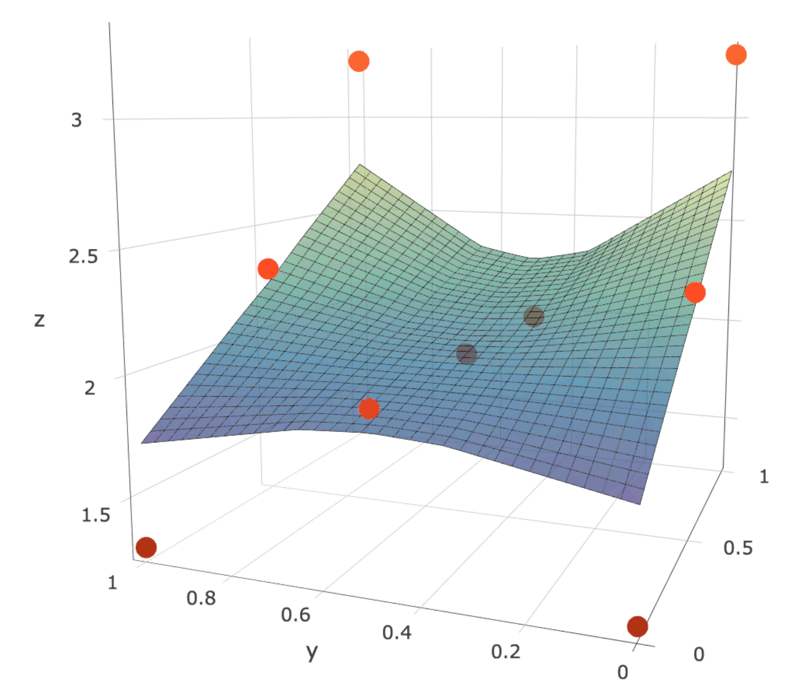
data matrix $X^{n \times d}$ with row vectors $x^{(i)} \in \mathbb{R}^d$

response values $f(x^{(i)})$ for all $x^{(i)}$

matrix $f(X)$ has rows $f(x^{(i)})$

Generate a function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ such that:





Problem Formulation

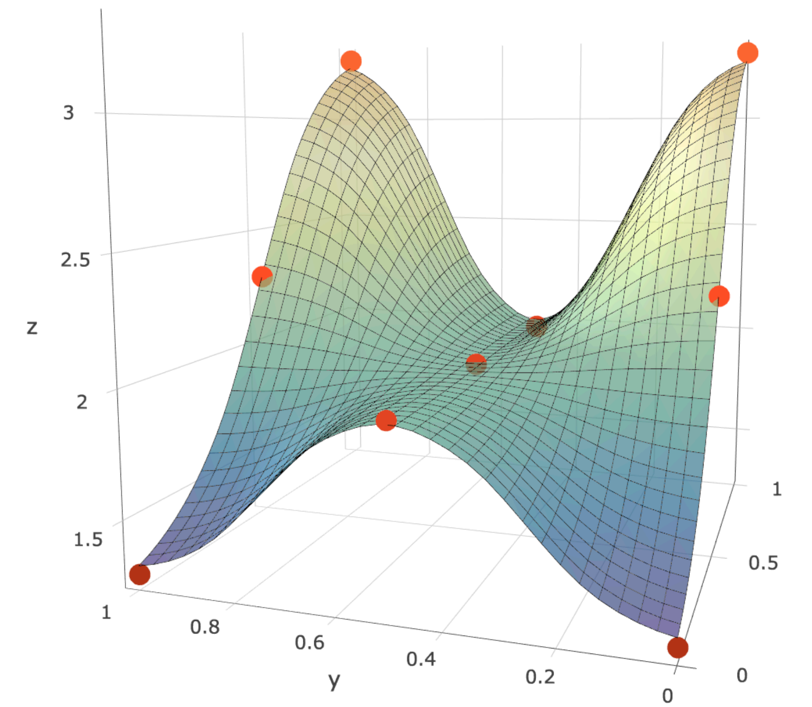
Given

underlying function $f: \mathbb{R}^d \rightarrow \mathbb{R}$

data matrix $X^{n \times d}$ with row vectors $x^{(i)} \in \mathbb{R}^d$

response values $f(x^{(i)})$ for all $x^{(i)}$

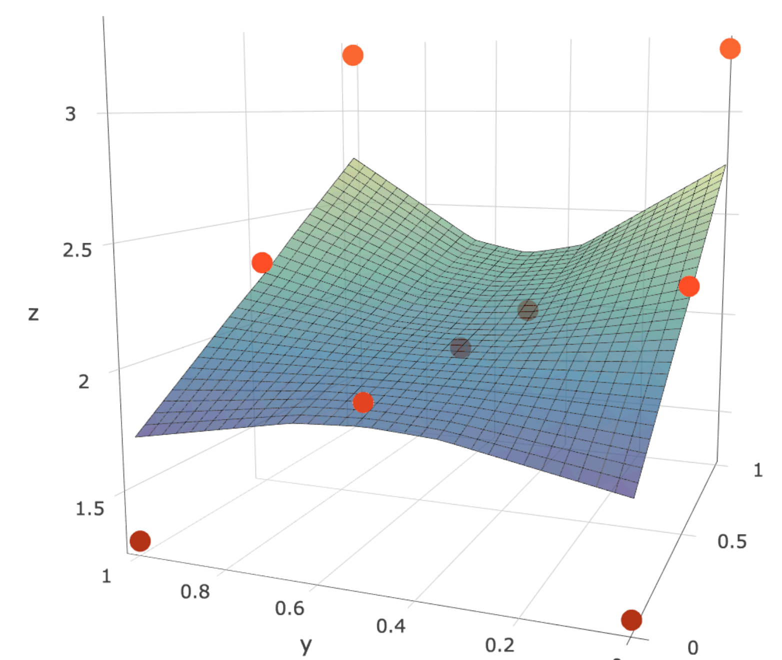
matrix $f(X)$ has rows $f(x^{(i)})$



Generate a function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ such that:

Interpolation

$g(x^{(i)})$ equals $f(x^{(i)})$ for all $x^{(i)}$



Problem Formulation

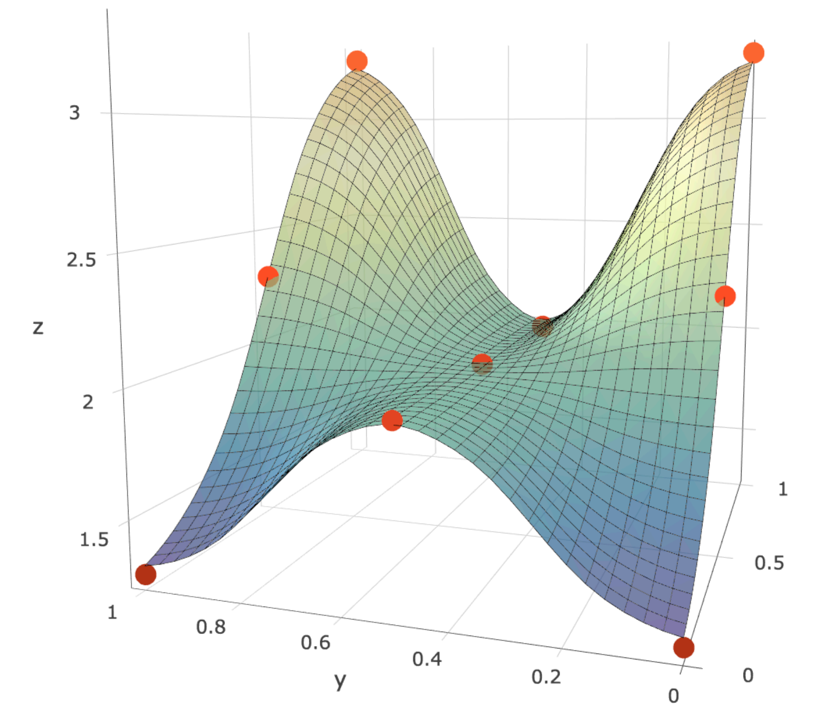
Given

underlying function $f: \mathbb{R}^d \rightarrow \mathbb{R}$

data matrix $X^{n \times d}$ with row vectors $x^{(i)} \in \mathbb{R}^d$

response values $f(x^{(i)})$ for all $x^{(i)}$

matrix $f(X)$ has rows $f(x^{(i)})$



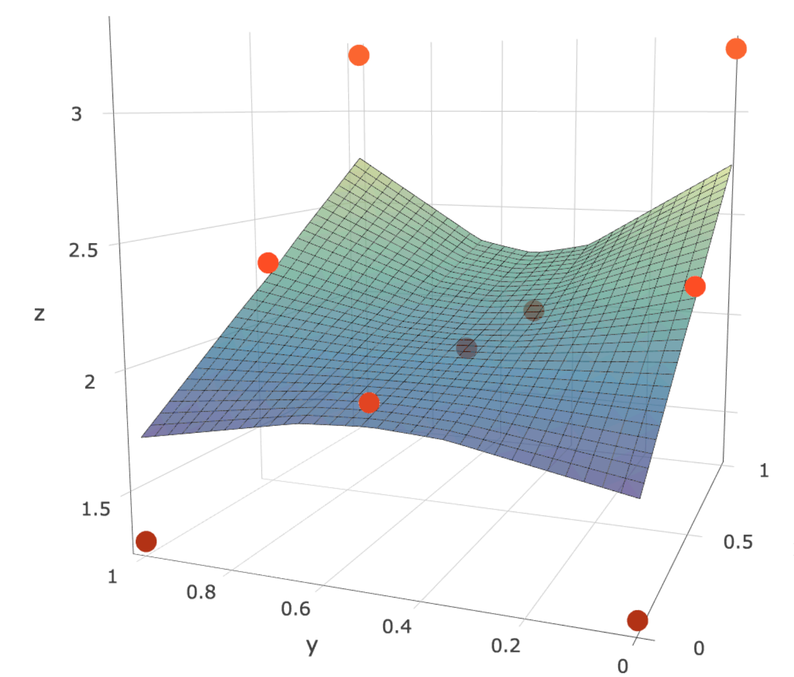
Generate a function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ such that:

Interpolation

$g(x^{(i)})$ equals $f(x^{(i)})$ for all $x^{(i)}$

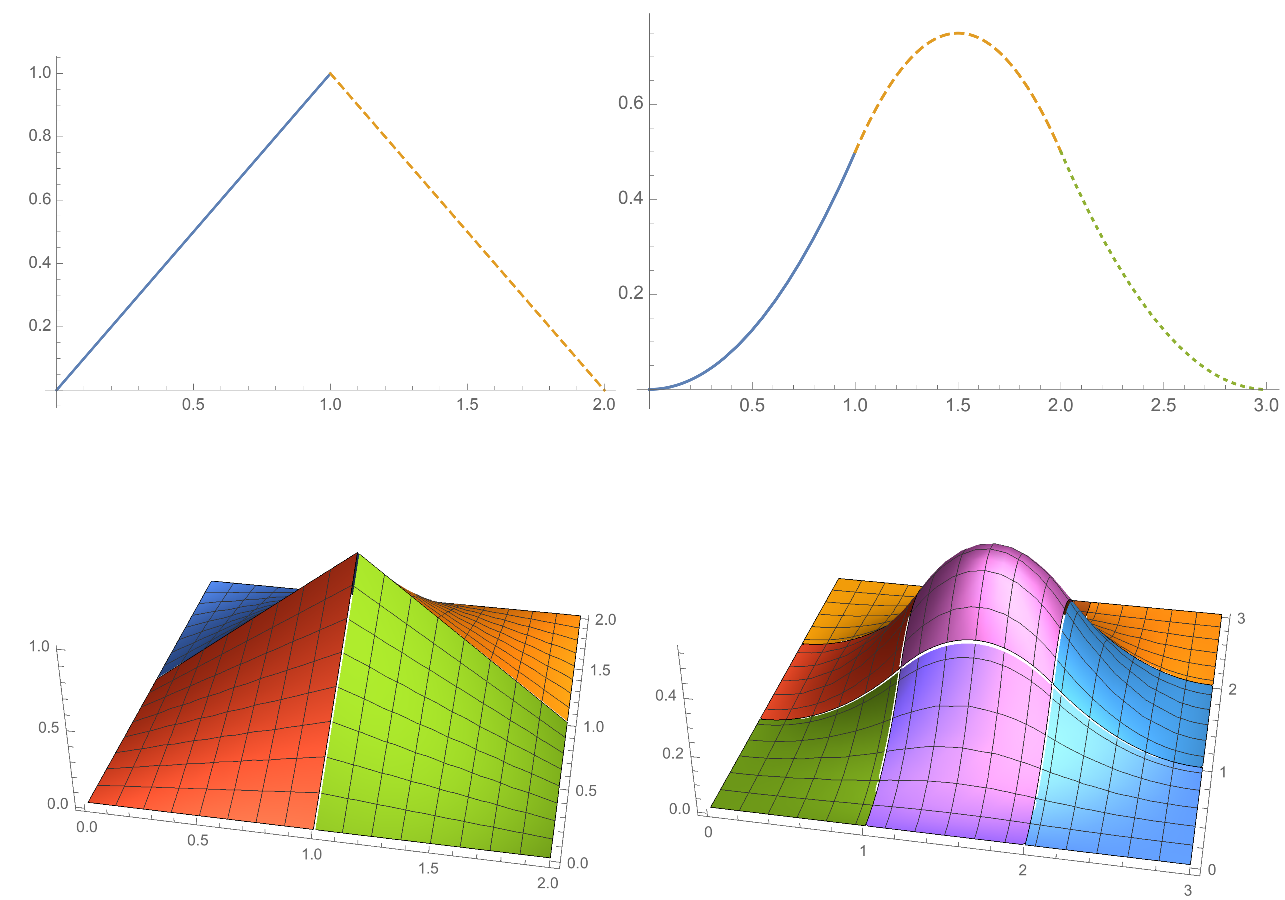
Approximation

g has parameters P and is the solution to $\min_P \| f(X) - g(X) \|$



Box Splines

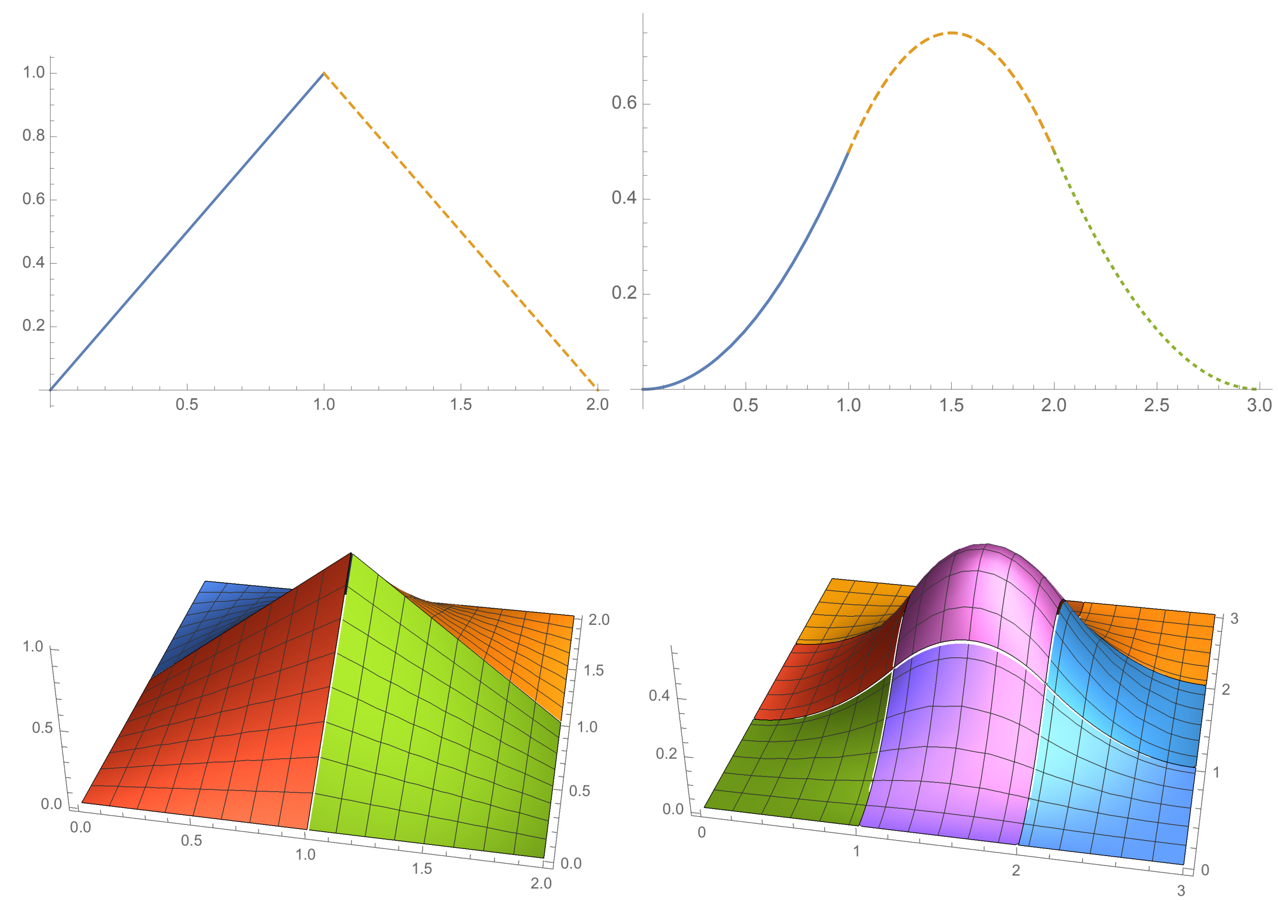
Proposed by C. de Boor as an extension of B-Splines into multiple dimensions (without using tensor products).



Box Splines

Proposed by C. de Boor as an extension of B-Splines into multiple dimensions (without using tensor products).

Can be shifted and scaled without losing smoothness.

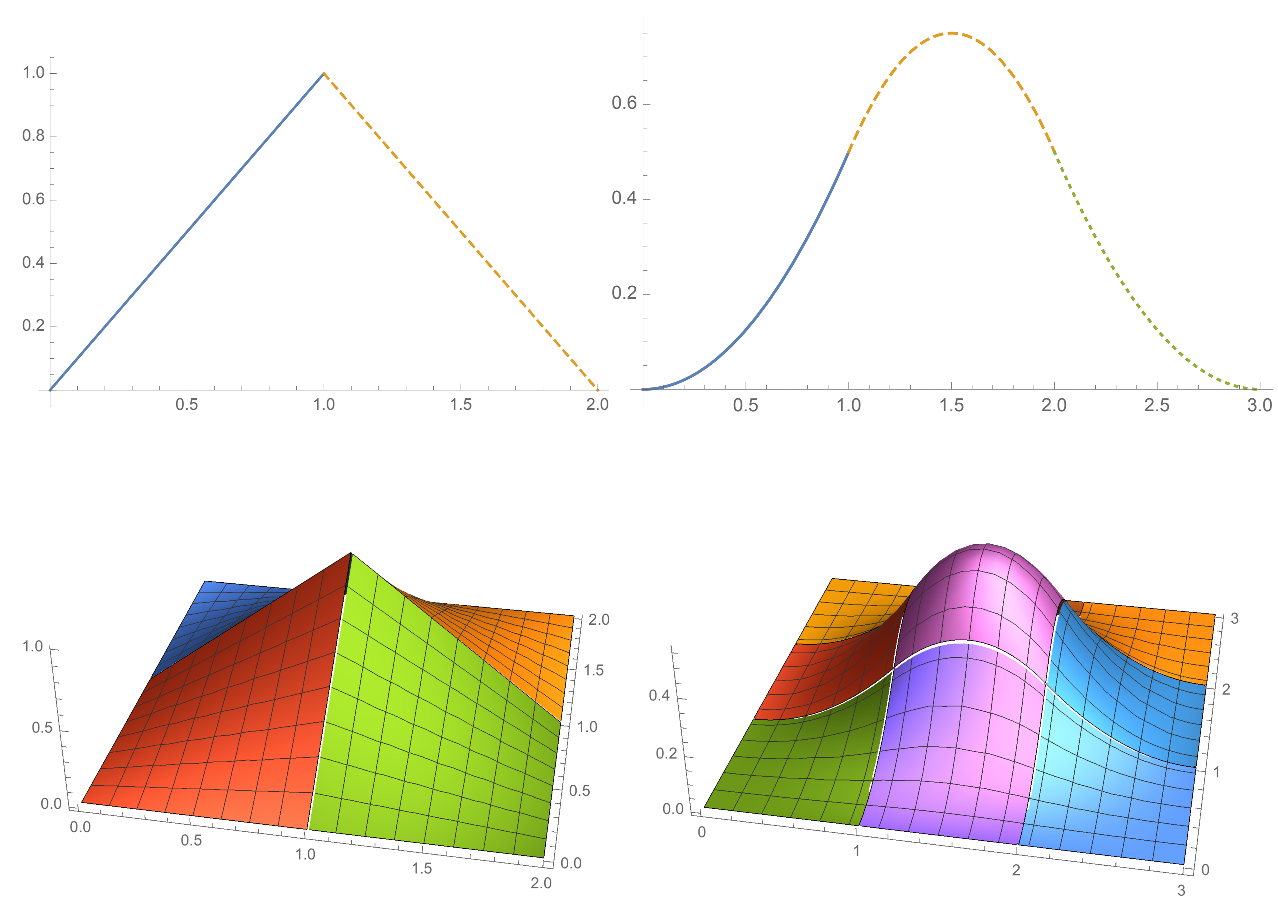


Box Splines

Proposed by C. de Boor as an extension of B-Splines into multiple dimensions (without using tensor products).

Can be shifted and scaled without losing smoothness.

Computationally tractable.

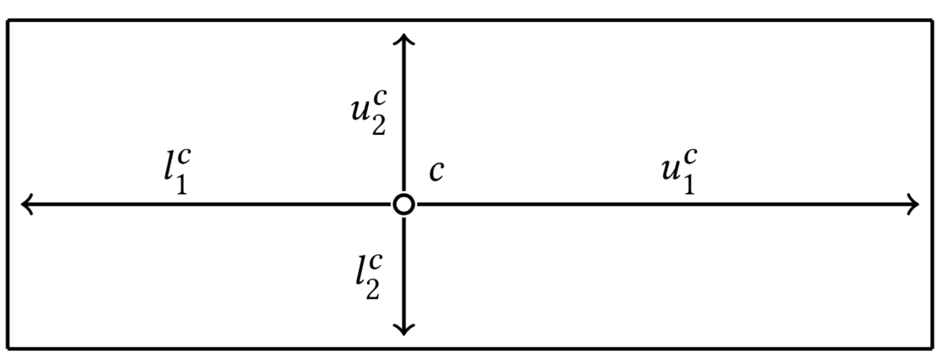


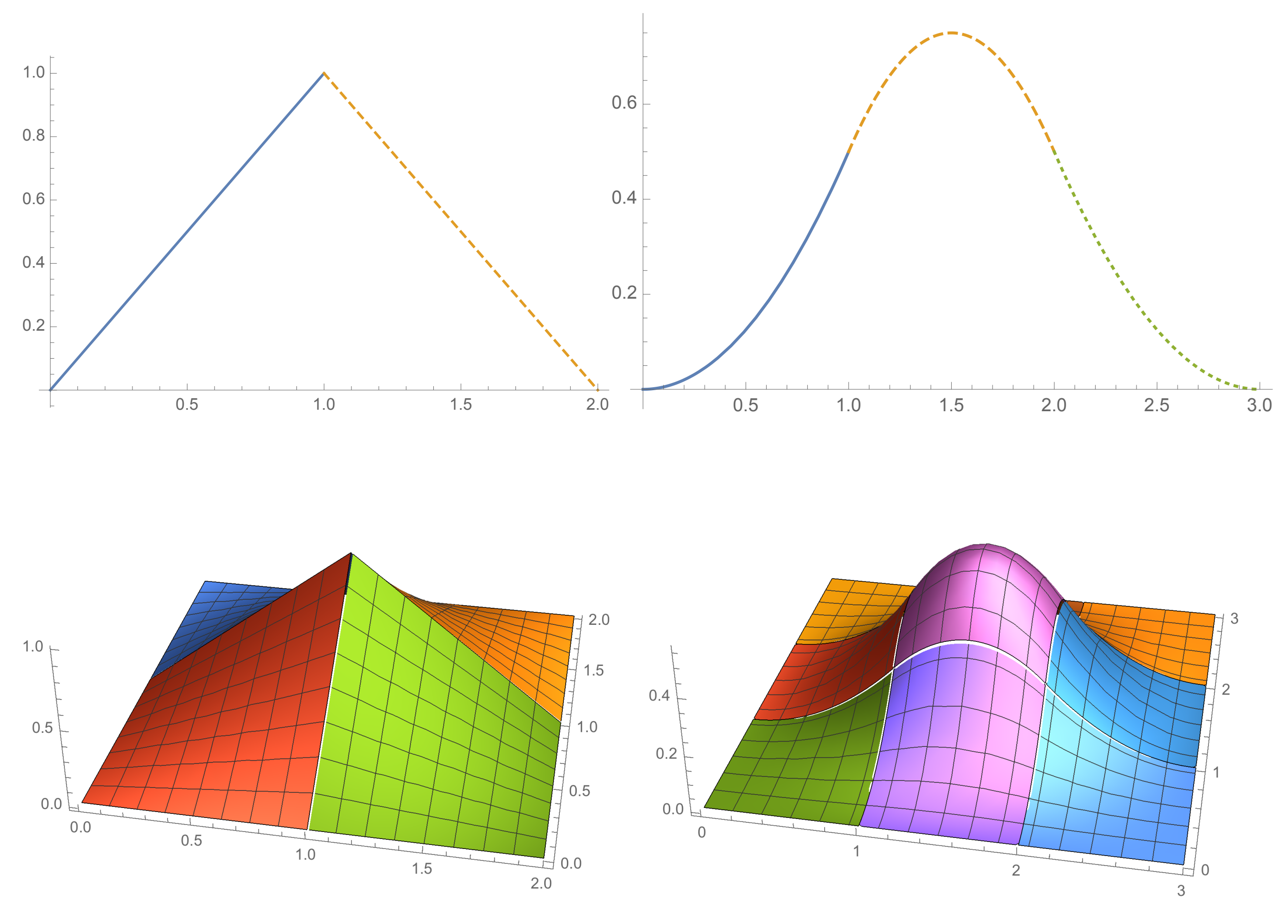
Box Splines

Proposed by C. de Boor as an extension of B-Splines into multiple dimensions (without using tensor products).

Can be shifted and scaled without losing smoothness.

Computationally tractable.





Max Box Mesh (MBM)

Properties

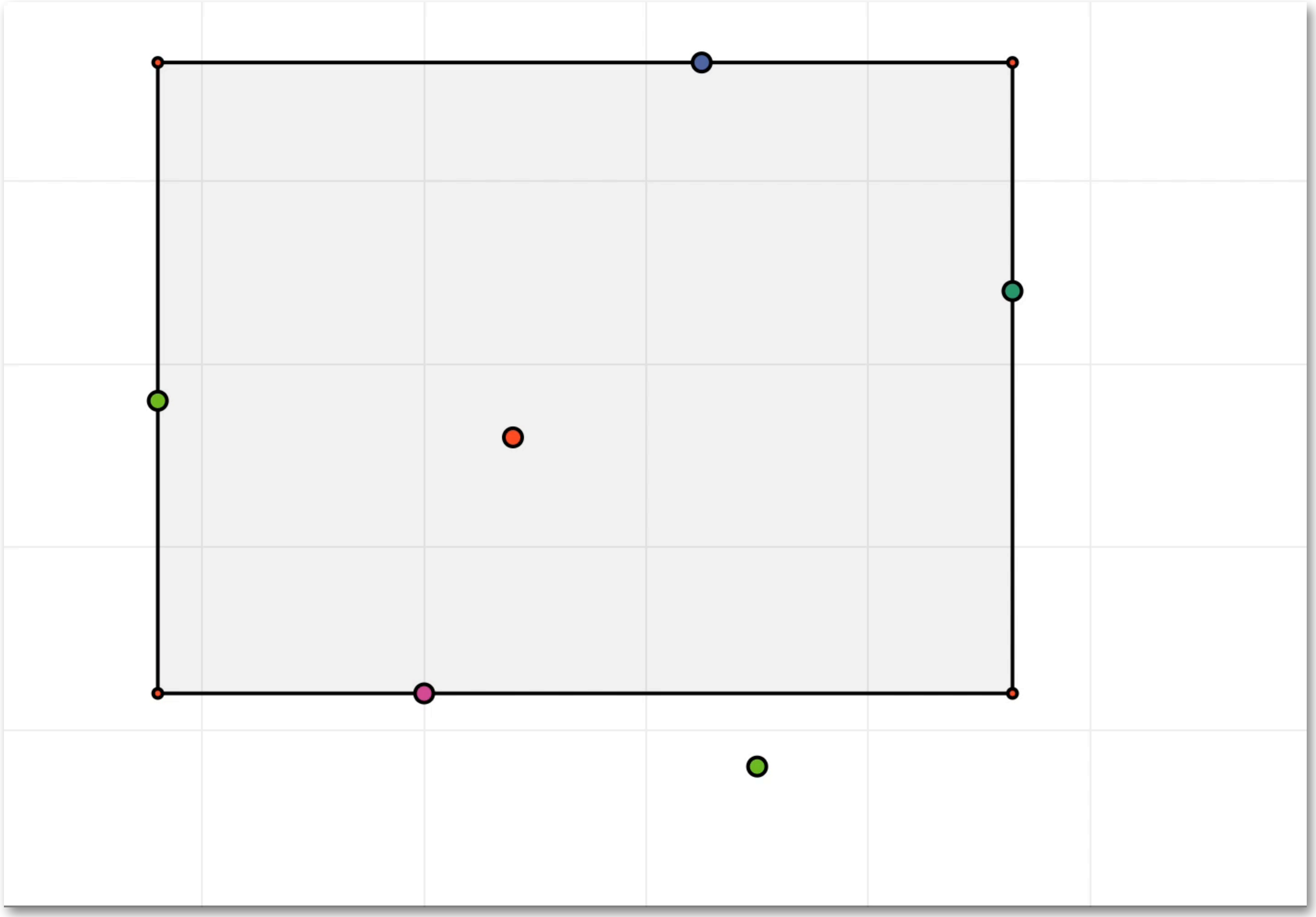
Largest max norm distance from anchor to edge of support.

Not always a covering for the space.

Construction Complexity

For each box (n)
  Distance to all (nd)
  For each dimension (2d)
    Sort distances (n log n)

$\mathcal{O}(n^2 d \log n)$



Iterative Box Mesh (IBM)

Properties

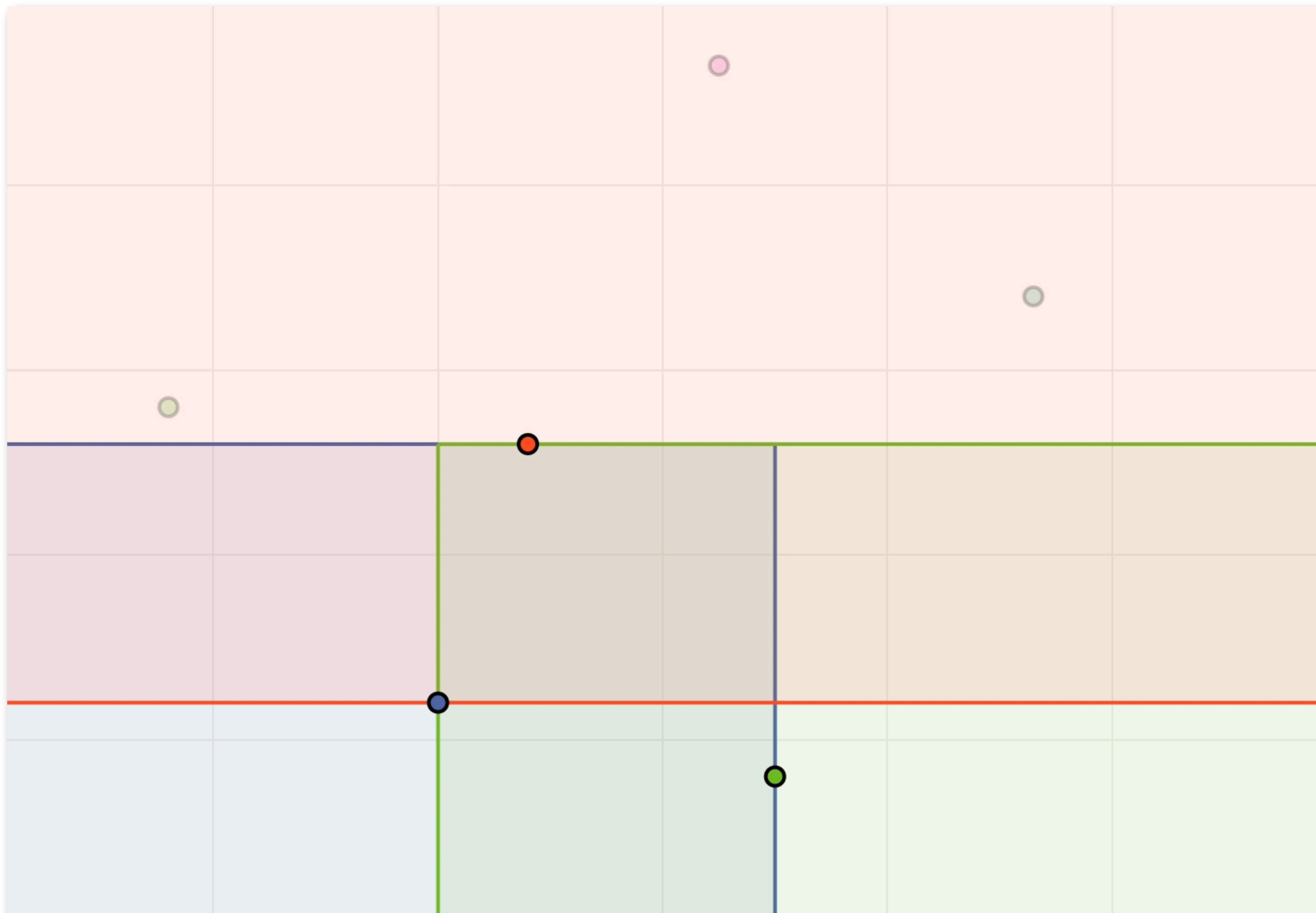
Built-in bootstrapping used to guide construction.

Always a covering for the space by construction.

Construction Complexity

Until all points are added (n)
Identify boxes containing new anchor to add (n)
Shrink boxes containing new anchor along all dimensions (d)

$$\mathcal{O}(n^2 + nd)$$



VIRGINIA TECH

Voronoi Mesh (VM)

Properties

Naturally shaped geometric regions (not forcibly axis aligned)

Always a covering for the space by construction.

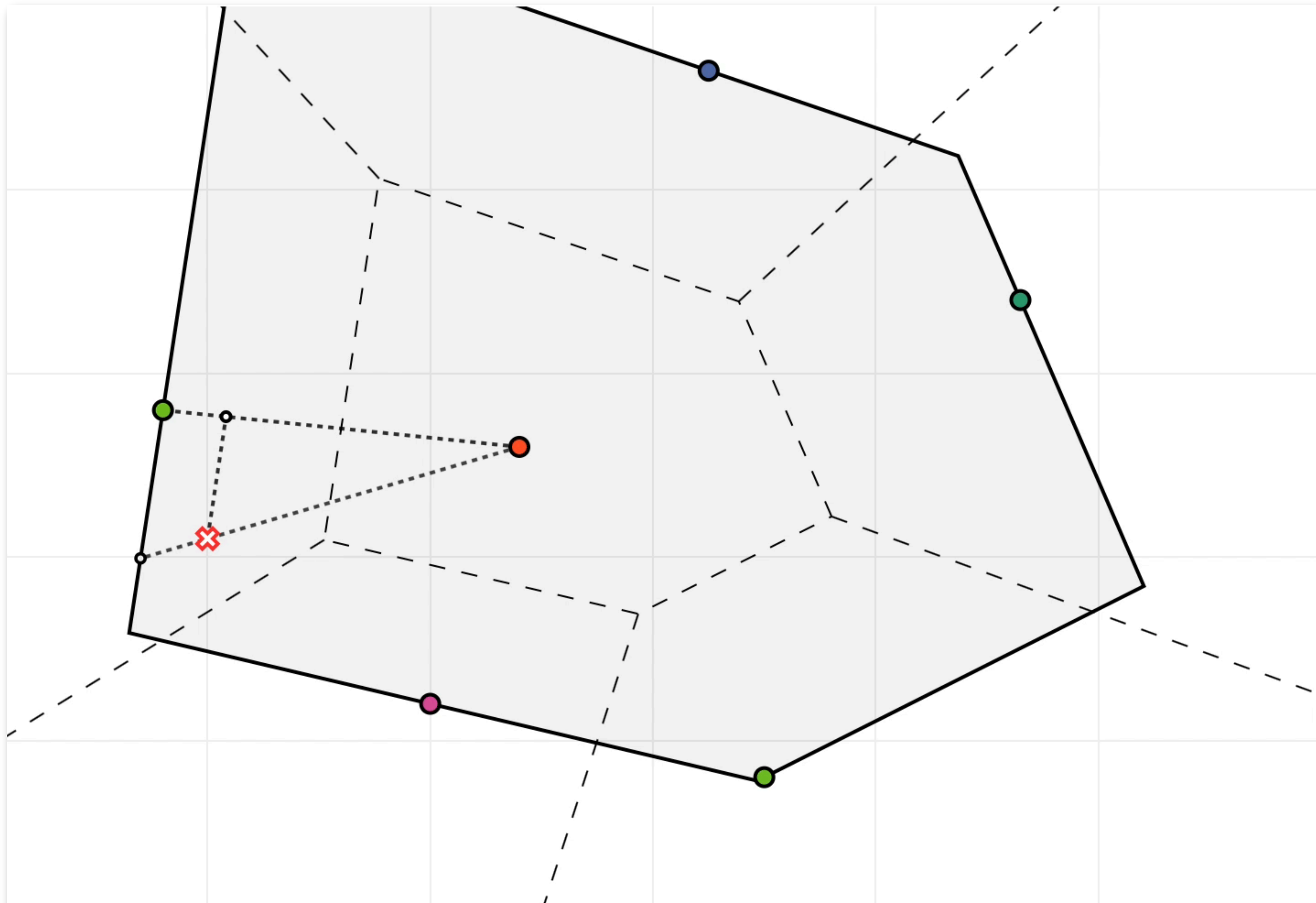
Construction Complexity

$$\mathcal{O}(n^2 d)$$

Prediction Complexity

For each cell anchor (n)
  For each other anchor, compute distance (nd)

$$\mathcal{O}(n^2 d)$$



Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When “c” is the number of control points used for a mesh, using an (n × c) matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(\mathrm{X})$ with cost $\mathcal{O}(nc^2 + c^3)$.

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When "c" is the number of control points used for a mesh, using an (n × c) matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(\mathrm{X})$ with cost $\mathcal{O}(nc^2 + c^3)$.

Bootstrapping

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When "c" is the number of control points used for a mesh, using an (n × c) matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(\mathrm{X})$ with cost $\mathcal{O}(nc^2 + c^3)$.

Bootstrapping

Initialize mesh only using the most central point

VIRGINIA TECH™

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When “c” is the number of control points used for a mesh, using an $(n \times c)$ matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(\mathrm{X})$ with cost $\mathcal{O}(\mathrm{nc}^2 + \mathrm{c}^3)$.

Bootstrapping

Initialize mesh only using the most central point
Fit mesh and evaluate error at all other points

VIRGINIA TECH.

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When "c" is the number of control points used for a mesh, using an $(n \times c)$ matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f(\mathrm{X})$ with cost $\mathcal{O}(nc^2 + c^3)$.

Bootstrapping

Initialize mesh only using the most central point
  Fit mesh and evaluate error at all other points
  Add (batch of) point(s) with largest error to mesh

VIRGINIA TECH.

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When “c” is the number of control points used for a mesh, using an $(n \times c)$ matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(X)$ with cost $\mathcal{O}(nc^2 + c^3)$.

Bootstrapping

Initialize mesh only using the most central point
  Fit mesh and evaluate error at all other points
  Add (batch of) point(s) with largest error to mesh
  If average error is not below ***error tolerance***, repeat

VIRGINIA TECH.

Fitting and Bootstrapping

Fitting

Evaluate all basis functions in the mesh at all points *n*.

When "c" is the number of control points used for a mesh, using an (n × c) matrix *A* of basis function evaluations at all points, solve the least squares problem $A\,x = f\,(\mathrm{X})$ with cost $\mathcal{O}(\mathrm{nc}^2 + \mathrm{c}^3)$.

Bootstrapping

Initialize mesh only using the most central point
Fit mesh and evaluate error at all other points
Add (batch of) point(s) with largest error to mesh
If average error is not below ***error tolerance***, repeat

Increased cost up to $\mathcal{O}(\mathrm{n})$

VIRGINIA TECH

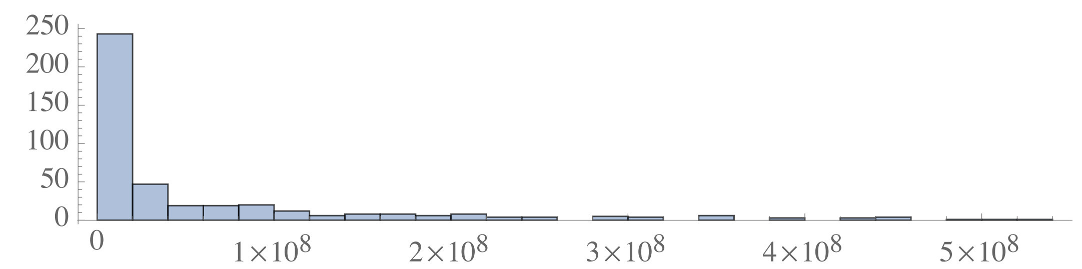
Testing and Evaluation: Data

Testing and Evaluation: Data

High Performance Computing File I/O

$n = 532, d = 4$

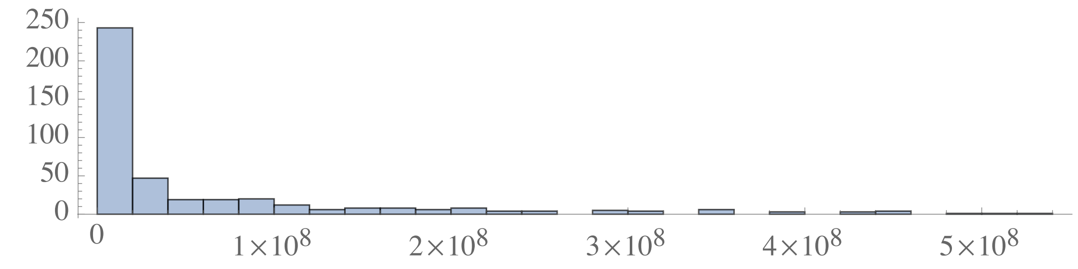
predicting *file I/O throughput*



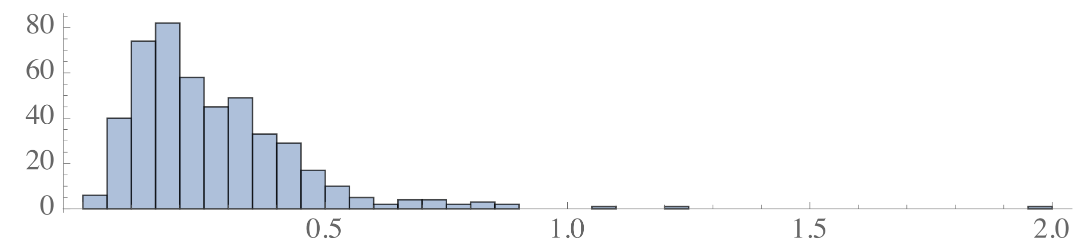
VIRGINIA TECH

Testing and Evaluation: Data

High Performance Computing File I/O
$n = 532, d = 4$
predicting *file I/O throughput*



Forest Fire
$n = 517, d = 12$
predicting *area burned*

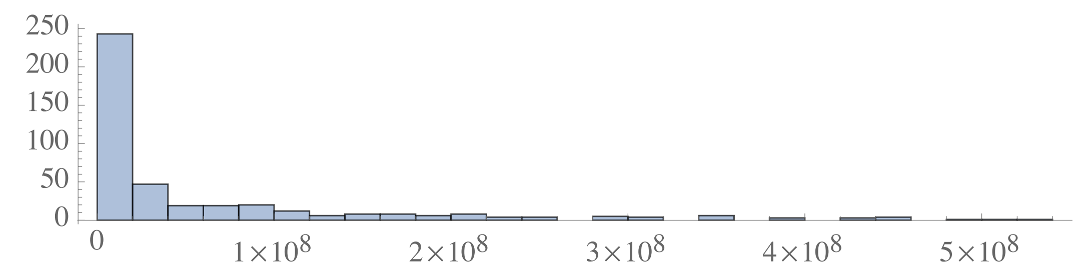


Testing and Evaluation: Data

High Performance Computing File I/O

$n = 532, d = 4$

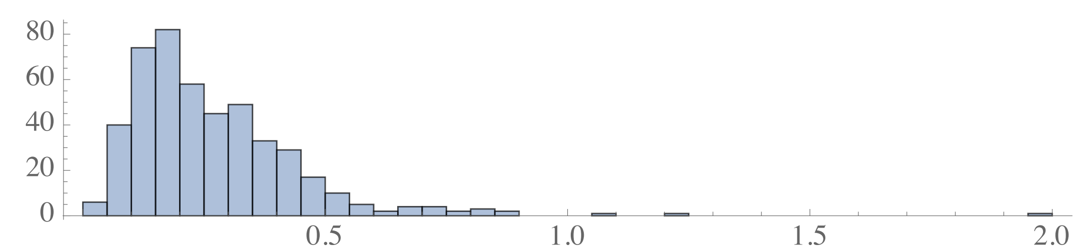
predicting *file I/O throughput*



Forest Fire

$n = 517, d = 12$

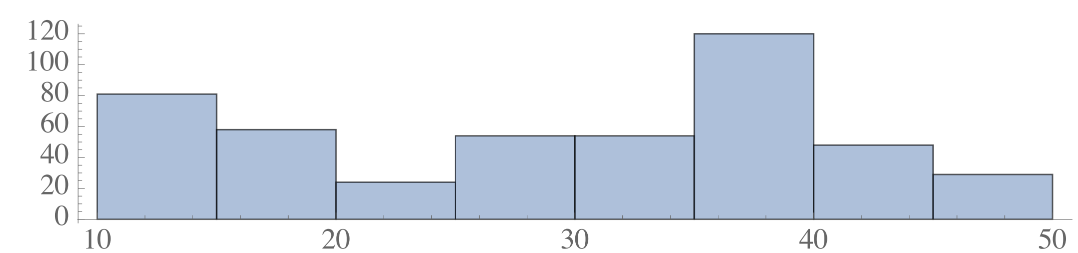
predicting *area burned*



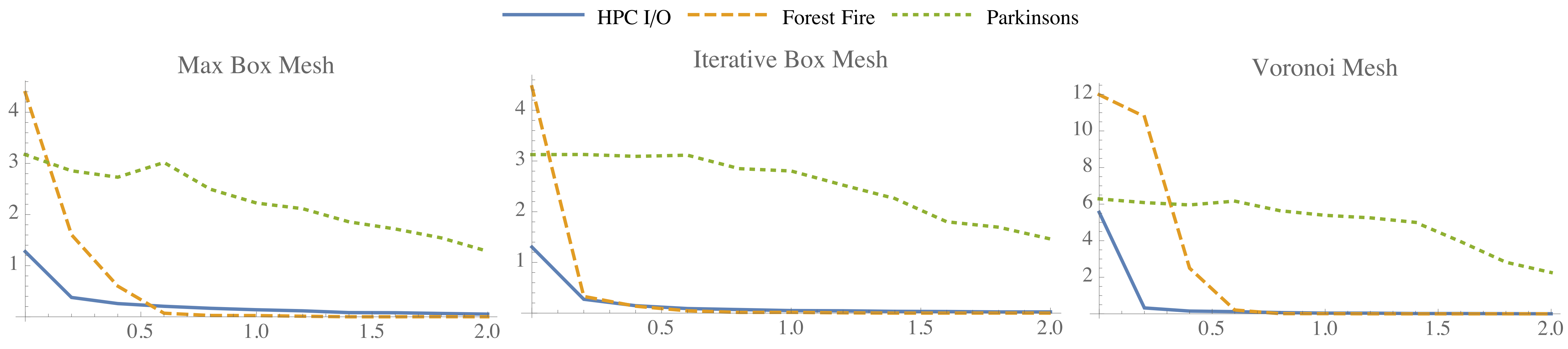
Parkinson's Clinical Evaluation

$n = 468, d = 16$

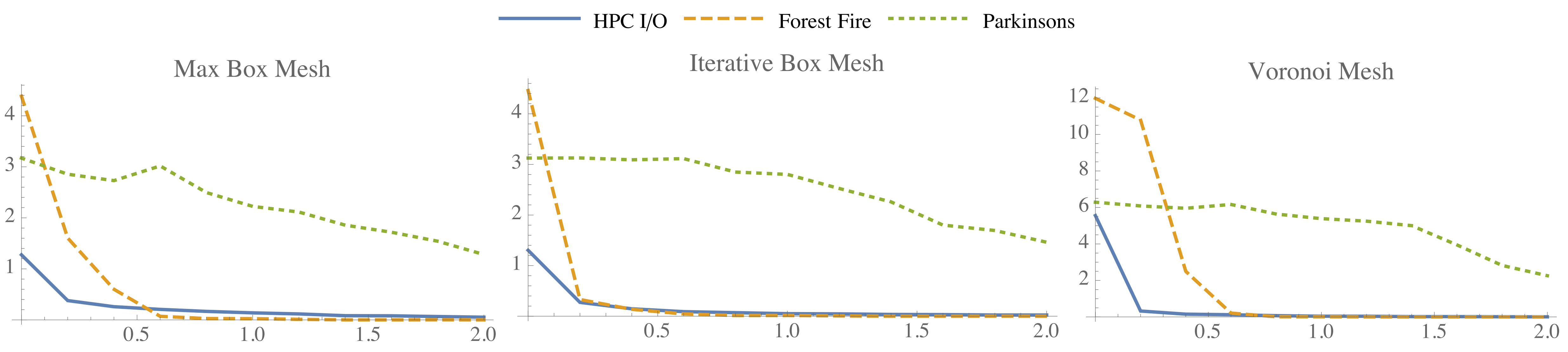
predicting *total clinical "UPDRS" score*



Time to Fit (y-axis) versus Error Tolerance (x-axis)



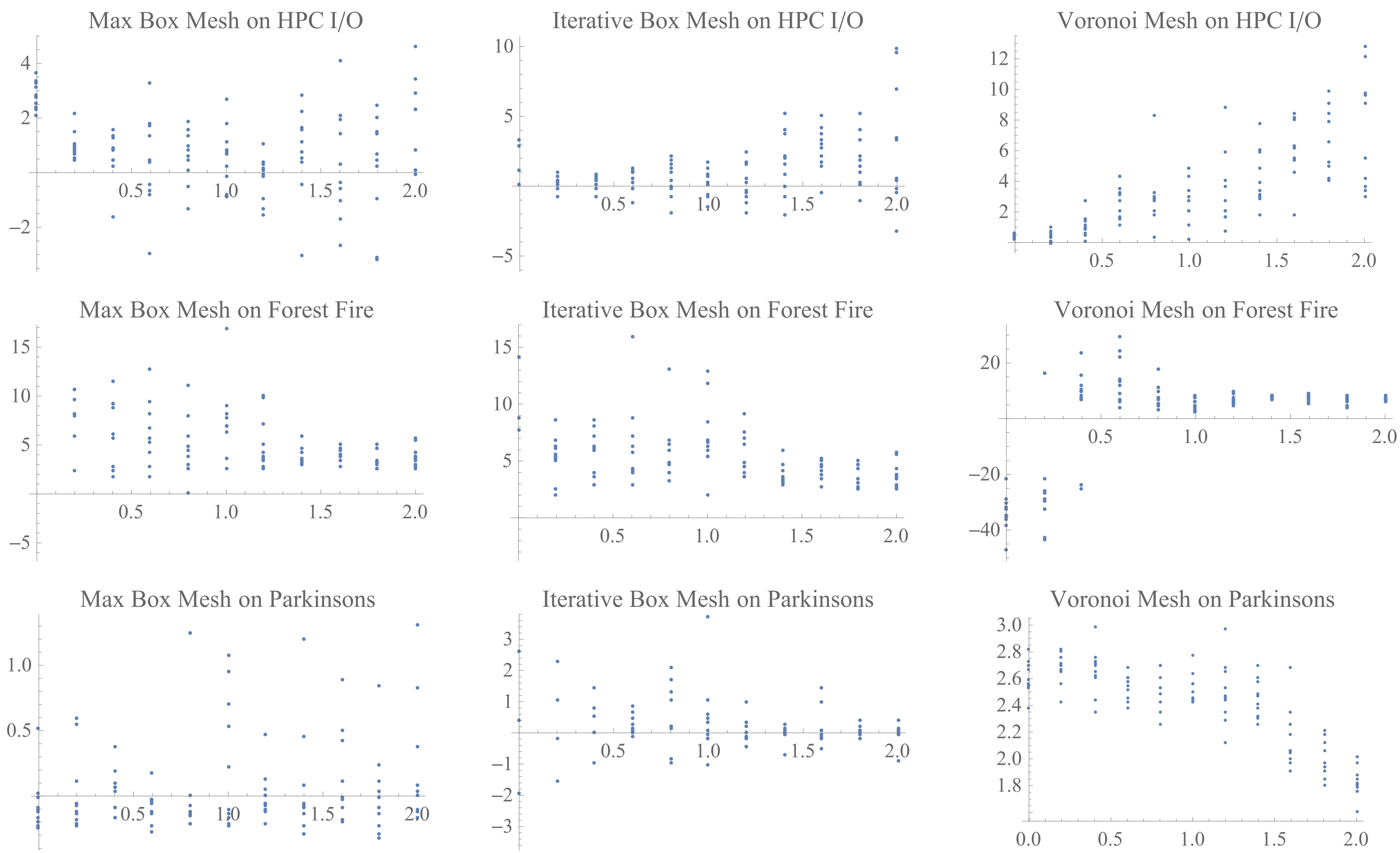
Time to Fit (y-axis) versus Error Tolerance (x-axis)



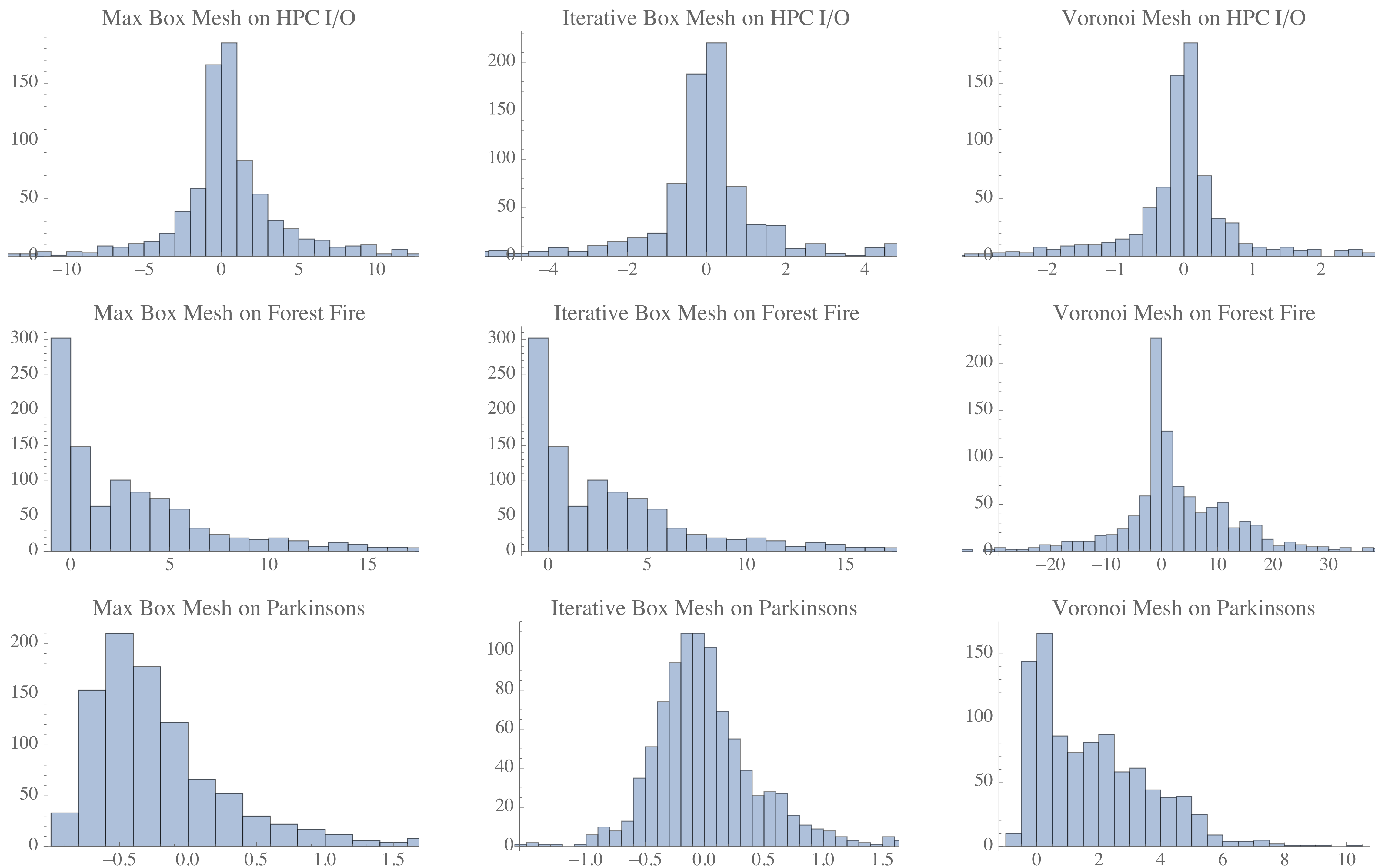
Optimal Tolerance & Accuracy

Data Set	Technique	Tolerance	Average Error
HPC I/O	MBM	1.2	0.597
Forest Fire	MBM	1.8	3.517
Parkinson’s	MBM	0.6	0.114
HPC I/O	IBM	0.4	0.419
Forest Fire	IBM	1.8	3.615
Parkinson’s	IBM	1.8	0.121
HPC I/O	VM	0.2	0.382
Forest Fire	VM	1.0	4.783
Parkinson’s	VM	2.0	1.824

Average Relative Testing Error (y-axis) versus Relative Error Tolerance (x-axis)



Histograms of Signed Relative Error



Summary of Contributions

Three techniques were proposed:
Max Box Mesh, Iterative Box Mesh, and Voronoi Mesh

Each is *theoretically straightforward*, *flexible*, and suitable for applications in *many dimensions*.

Summary of Contributions

Three techniques were proposed:
Max Box Mesh, Iterative Box Mesh, and Voronoi Mesh

Each is *theoretically straightforward*, *flexible*, and suitable for applications in *many dimensions*.

Future Work

Alternative smoothing methods may be preferred that scale better with the number of data points.

Further theoretical and empirical comparisons may be done against more widely used statistical / ML techniques.