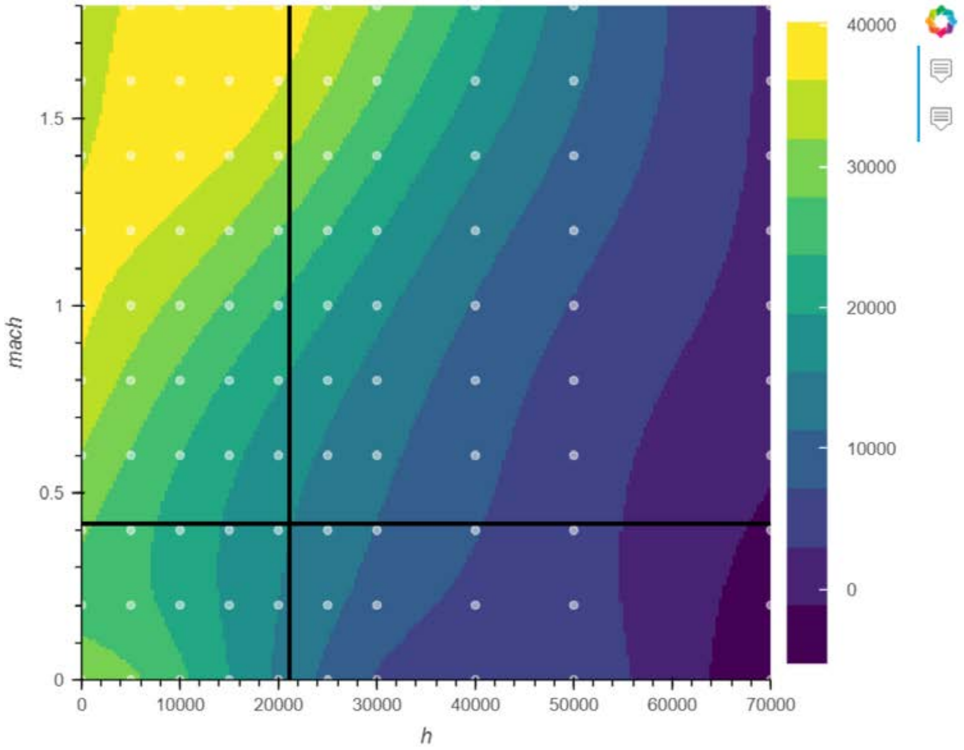
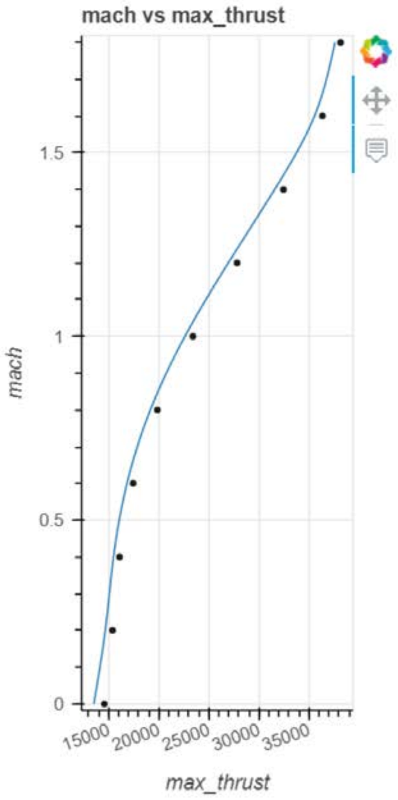
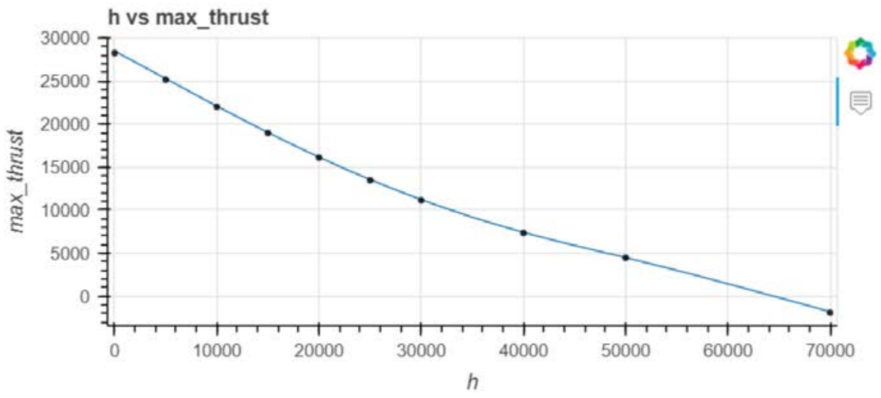






Interpolant improvements and lessons learned

John Jasa
john.jasa@nasa.gov
Banner Quality Management Inc (BQMI)
NASA Glenn Research Center

2022 OpenMDAO Workshop
Cleveland, OH
Oct 24-26, 2022

Using tabular data and interpolants in OpenMDAO

We now have fast fixed interpolants

Be wary of using piecewise linear interpolants

Interpolant accuracy and efficiency matter

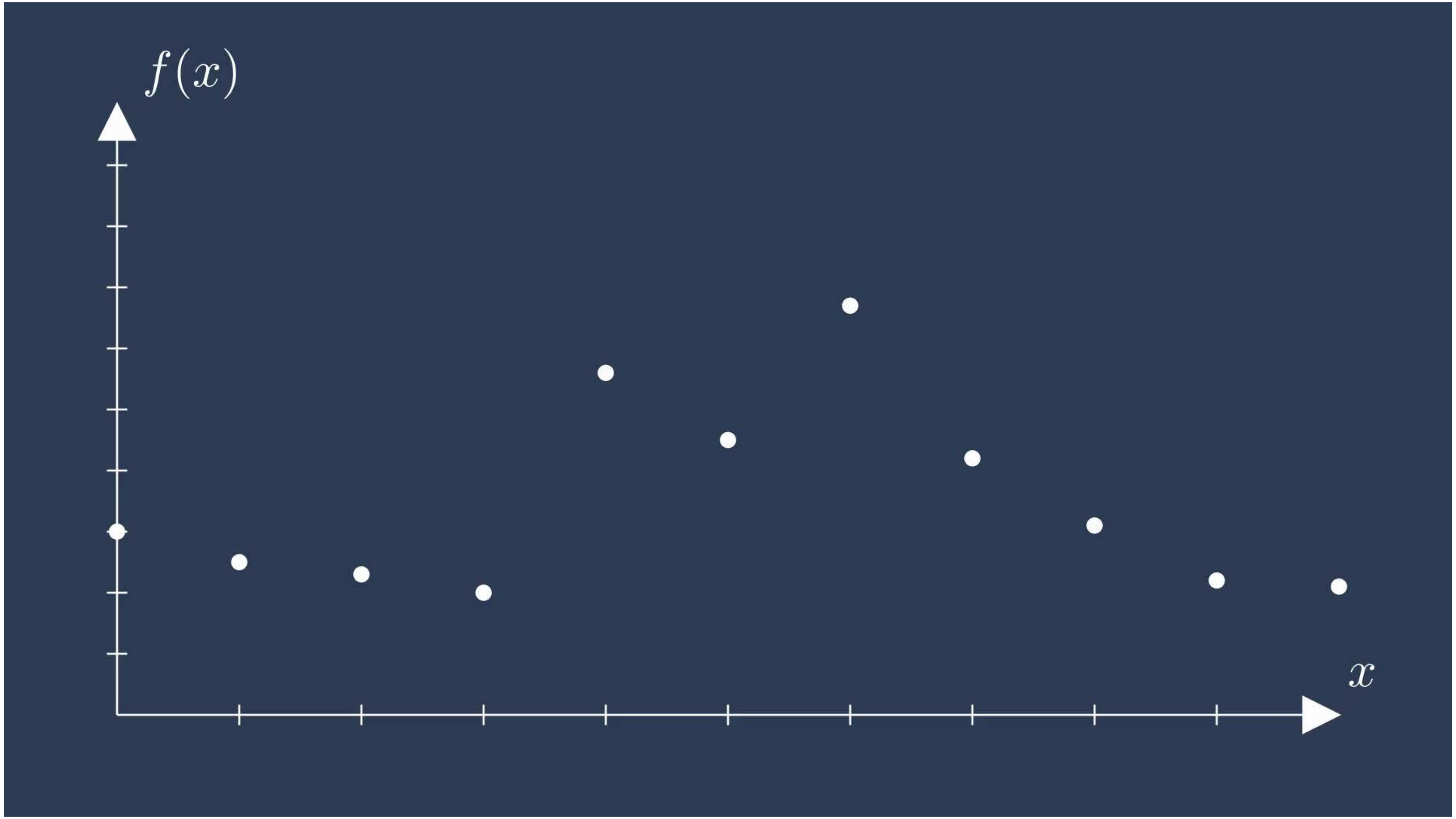
Using tabular data and interpolants in OpenMDAO

We now have fast fixed interpolants

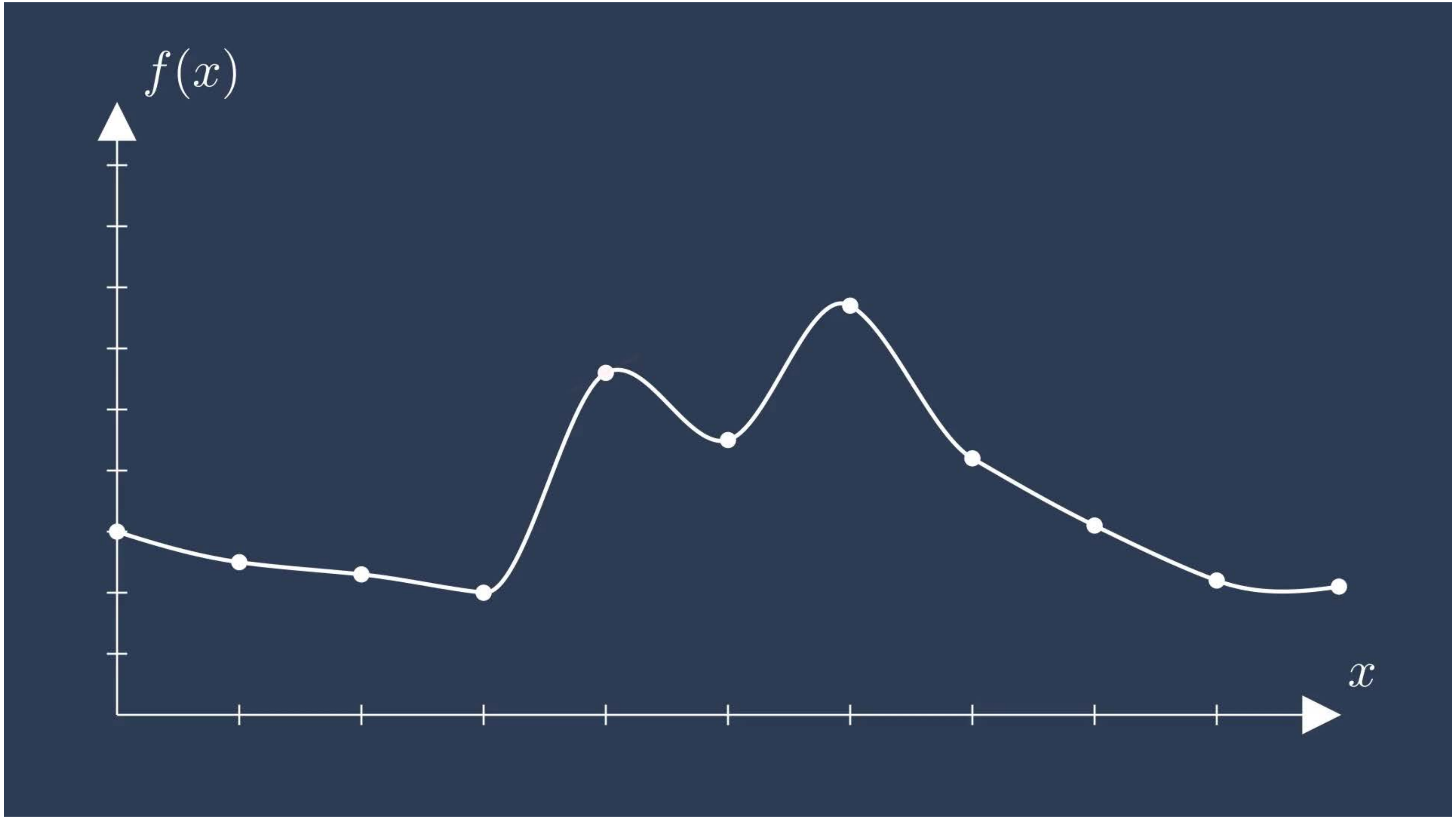
Be wary of using piecewise linear interpolants

Interpolant accuracy and efficiency matter

An interpolant (or metamodel or surrogate) fills in data between discrete points



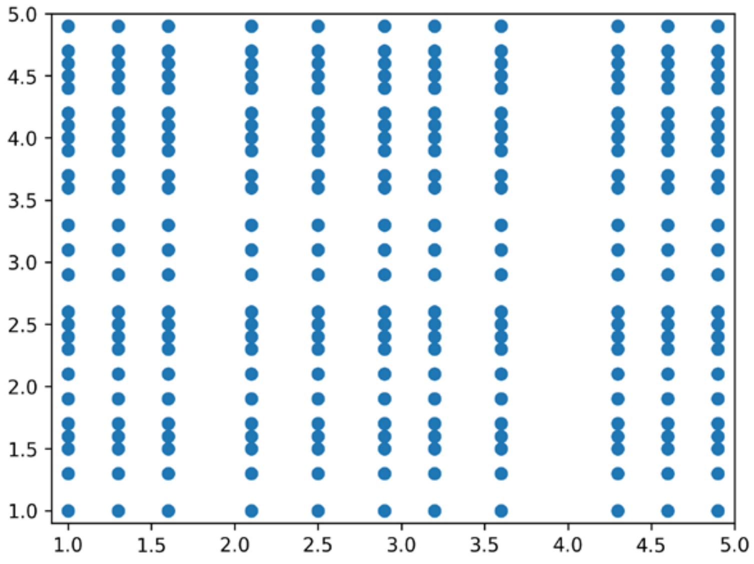
An interpolant (or metamodel or surrogate) fills in data between discrete points



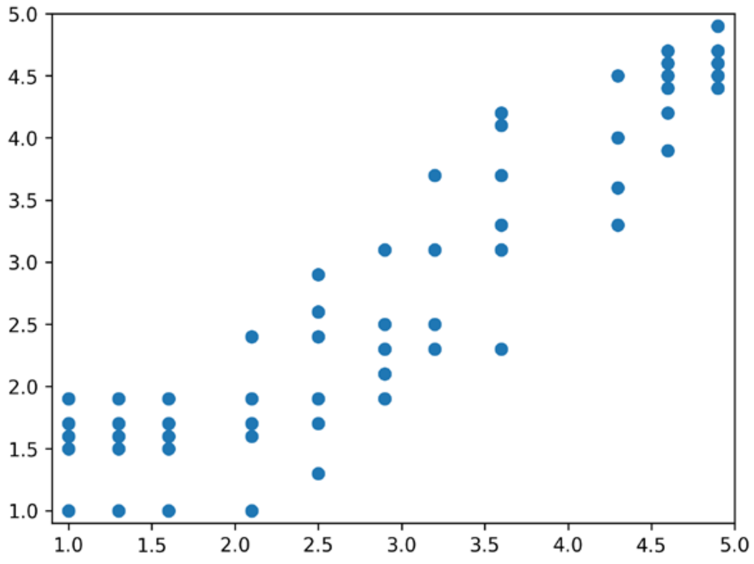
Using curve fits across tabular data happens all the time in systems engineering

- Fitting engine performance data
- Pressure and thrust coefficients across a control scheme for wind turbines
- Aerodynamic performance data
- Atmospheric modeling across different altitudes
- Heat exchanger performance at different temperatures and fluid flow rates

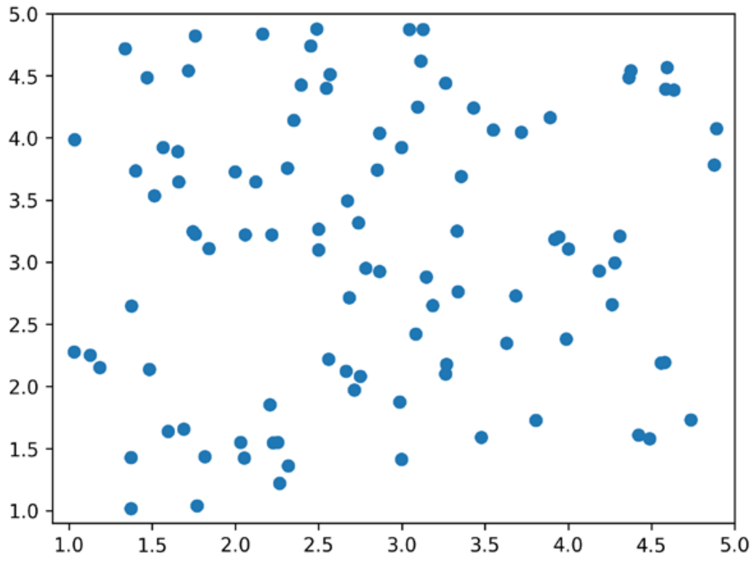
Sometimes data is structured; sometimes it's not



Structured



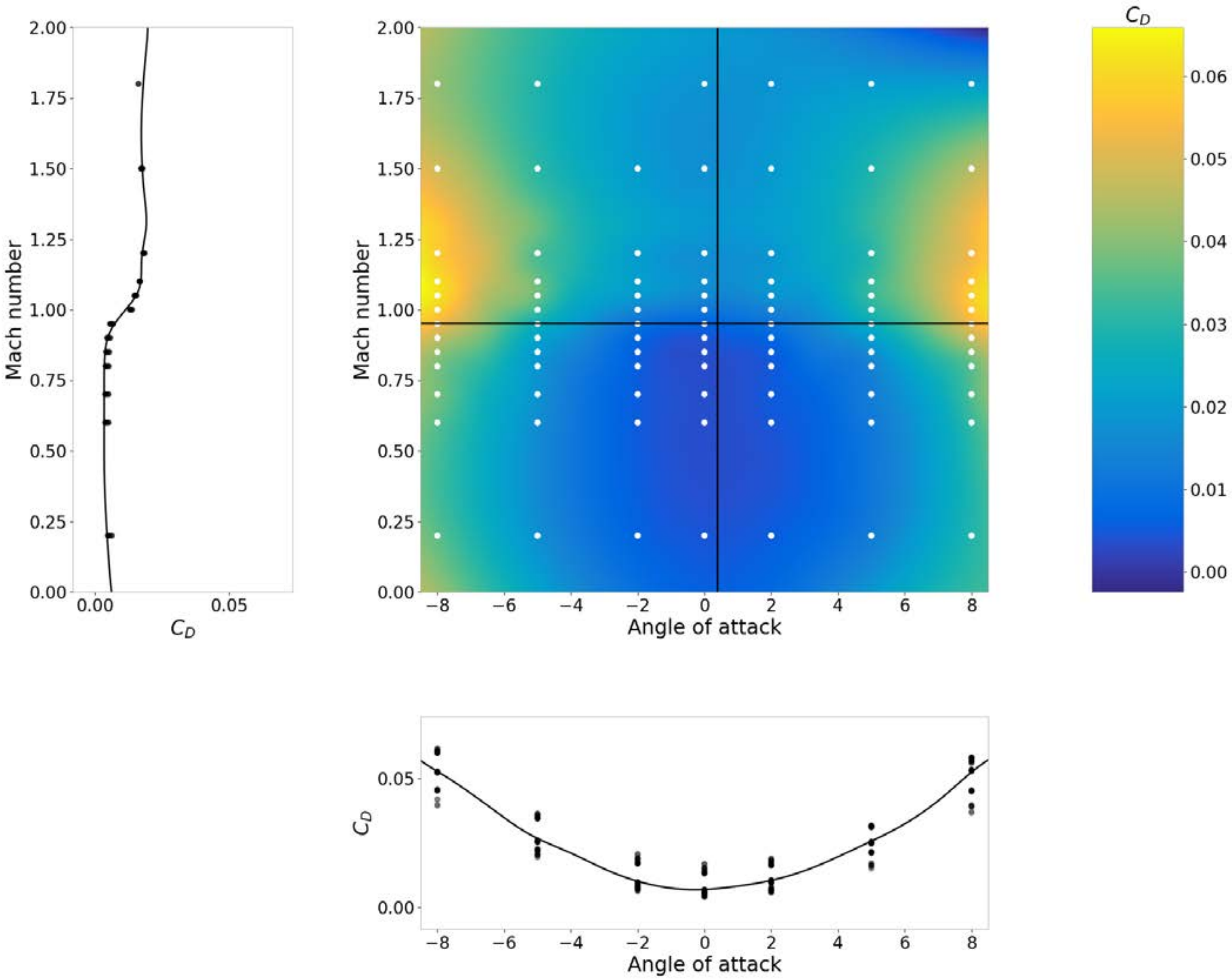
Semi-structured



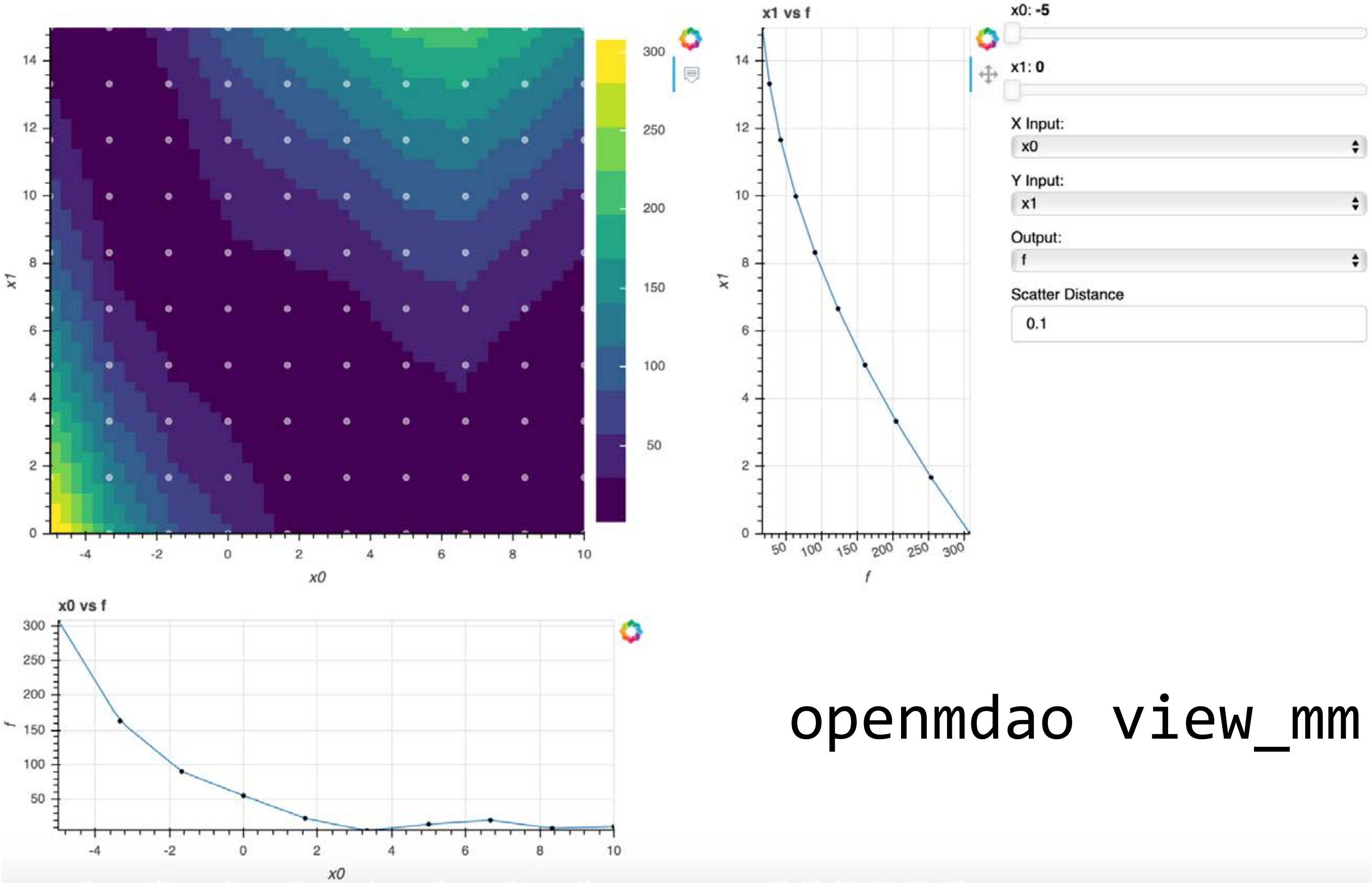
Unstructured

OpenMDAO offers metamodels for all three types

During my PhD I developed a script to interactively view multidimensional fits



Now you too have that power, thanks to OpenMDAO and the dev team!




```
openmdao view_mm run_script.py
```

Using tabular data and interpolants in OpenMDAO

We now have fast fixed interpolants

Be wary of using piecewise linear interpolants

Interpolant accuracy and efficiency matter

Sometimes you need to call an interpolant millions of times during an optimization

In the conceptual design phase, you might use precalculated tabular aerodynamic data

Surprisingly, interpolating this data might become the bottleneck if you're doing a trajectory optimization

The interpolation schemes were not optimal for our trajectory design problems

1. Some tables were interpolated using higher order fits unnecessarily
2. The interpolation methods were implemented
   as a series of nested 1-D interpolations
3. They also provided derivatives wrt the training points
4. This flexibility decreased performance
5. Vectorization wasn't possible

OpenMDAO now includes *fast,* purpose-built metamodels for fixed data

OpenMDAO now includes *fast,* purpose-built metamodels for fixed data

OpenMDAO

Search this book...

☐ Include Source Docs

Welcome to OpenMDAO

GETTING STARTED

Getting Started

BASIC USER GUIDE

Basic User Guide

ADVANCED USER GUIDE

Advanced User Guide

REFERENCE GUIDE

Theory Manual

Features

Examples

OTHER USEFUL DOCS

Command Line Tools

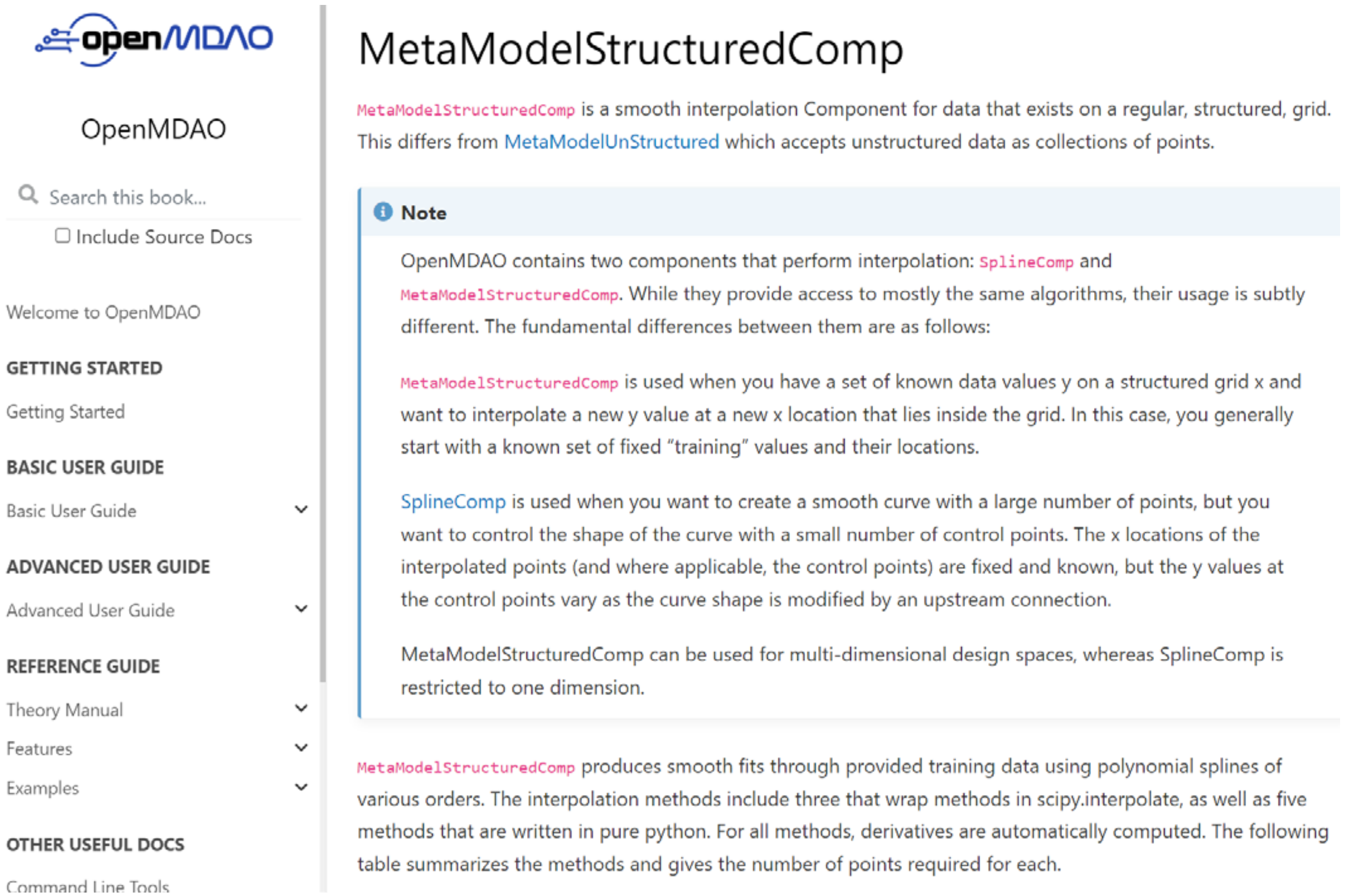
MetaModelStructuredComp

`MetaModelStructuredComp` is a smooth interpolation Component for data that exists on a regular, structured, grid. This differs from MetaModelUnStructured which accepts unstructured data as collections of points.

> **Note**
>
> OpenMDAO contains two components that perform interpolation: `SplineComp` and `MetaModelStructuredComp`. While they provide access to mostly the same algorithms, their usage is subtly different. The fundamental differences between them are as follows:
>
> `MetaModelStructuredComp` is used when you have a set of known data values y on a structured grid x and want to interpolate a new y value at a new x location that lies inside the grid. In this case, you generally start with a known set of fixed “training” values and their locations.
>
> SplineComp is used when you want to create a smooth curve with a large number of points, but you want to control the shape of the curve with a small number of control points. The x locations of the interpolated points (and where applicable, the control points) are fixed and known, but the y values at the control points vary as the curve shape is modified by an upstream connection.
>
> MetaModelStructuredComp can be used for multi-dimensional design spaces, whereas SplineComp is restricted to one dimension.

`MetaModelStructuredComp` produces smooth fits through provided training data using polynomial splines of various orders. The interpolation methods include three that wrap methods in scipy.interpolate, as well as five methods that are written in pure python. For all methods, derivatives are automatically computed. The following table summarizes the methods and gives the number of points required for each.

OpenMDAO now includes *fast*, purpose-built metamodels for fixed data



Method	Order	Description
slinear	1	Basic linear interpolation
lagrange2	2	Second order Lagrange polynomial
lagrange3	3	Third order Lagrange polynomial
akima	3	Interpolation using Akima splines
cubic	3	Cubic spline, with continuity of derivatives between segments
scipy_slinear	1	Scipy linear interpolation. Same as slinear, though slower
scipy_cubic	3	Scipy cubic interpolation. More accurate than cubic, but slower
scipy_quintic	5	Scipy quintic interpolation. Most accurate, but slowest
1D-slinear	1	Linear on a fixed 1D table
2D-slinear	1	Linear on a fixed 2D table
3D-slinear	1	Linear on a fixed 3D table
1D-akima	3	Akima on a fixed 1D table
3D-lagrange2	2	Second order Lagrange on a fixed 3D table
3D-lagrange3	3	Third order Lagrange on a fixed 3D table

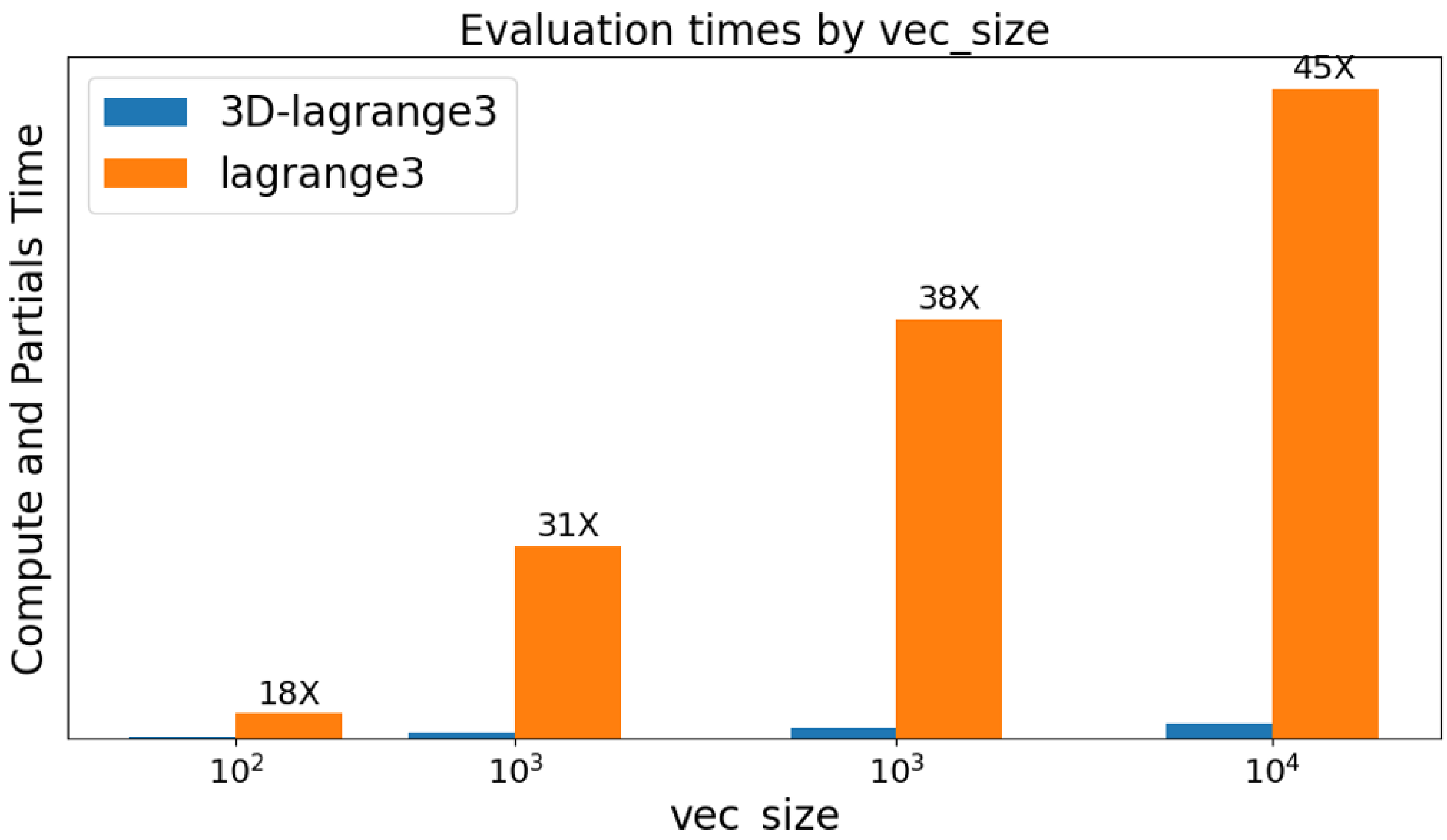
OpenMDAO now includes *fast*, purpose-built metamodels for fixed data

3D-lagrange3 is a sped-up version of *lagrange3* with:

- Fixed 3D size instead of N-D
- Fully vectorized calculation
- Cached coefficients for each cell

Method	Order	Description
slinear	1	Basic linear interpolation
lagrange2	2	Second order Lagrange polynomial
lagrange3	3	Third order Lagrange polynomial
akima	3	Interpolation using Akima splines
cubic	3	Cubic spline, with continuity of derivatives between segments
scipy_slinear	1	Scipy linear interpolation. Same as slinear, though slower
scipy_cubic	3	Scipy cubic interpolation. More accurate than cubic, but slower
scipy_quintic	5	Scipy quintic interpolation. Most accurate, but slowest
1D-slinear	1	Linear on a fixed 1D table
2D-slinear	1	Linear on a fixed 2D table
3D-slinear	1	Linear on a fixed 3D table
1D-akima	3	Akima on a fixed 1D table
3D-lagrange2	2	Second order Lagrange on a fixed 3D table
3D-lagrange3	3	Third order Lagrange on a fixed 3D table

These fast methods sped up our trajectory interpolants by >30x



3D table dimensions: 25 x 15 x 12; 4500 points

Using tabular data and interpolants in OpenMDAO

We now have fast fixed interpolants

Be wary of using piecewise linear interpolants

Interpolant accuracy and efficiency matter

Using piecewise linear interpolants can introduce problems when using gradients

Piecewise linear fits can be problematic in real engineering cases

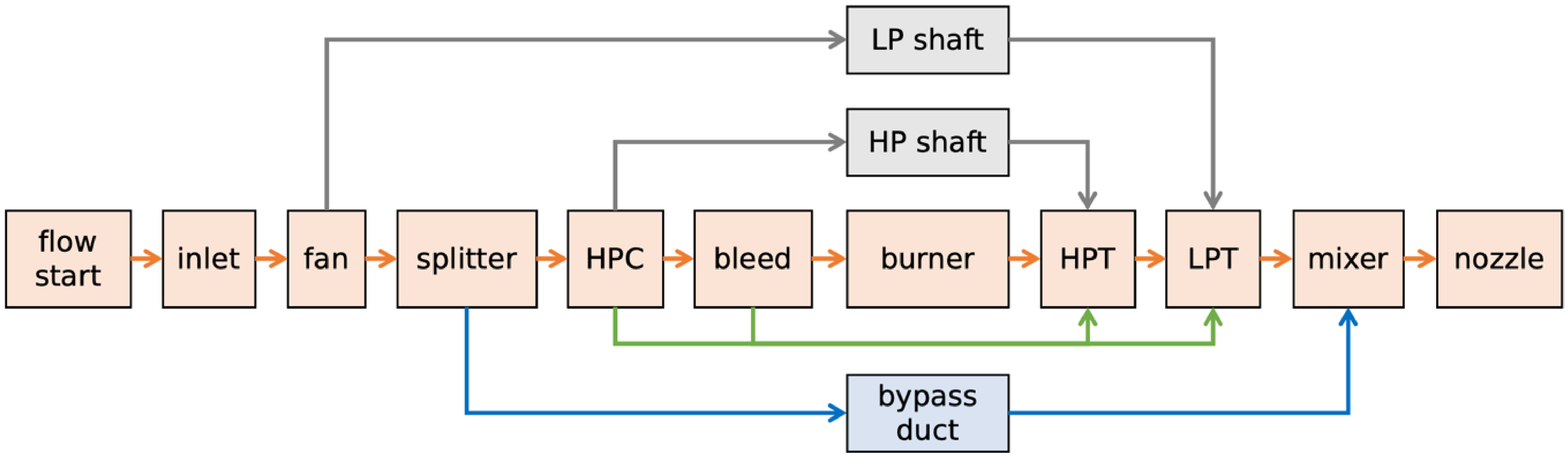


Figure courtesy Laurens Voet, MIT

Piecewise linear fits can be problematic in real engineering cases

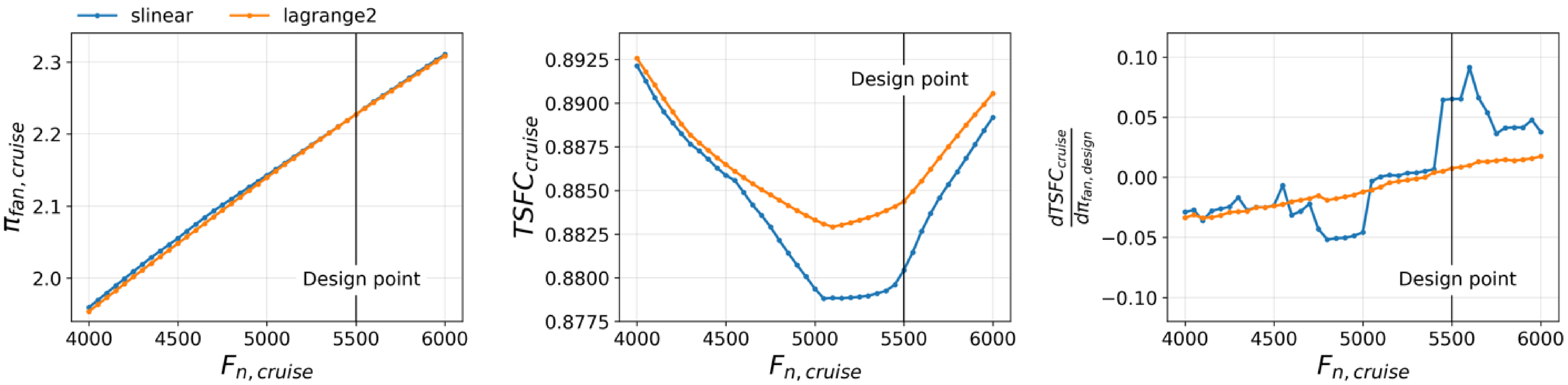


Figure courtesy Laurens Voet, MIT

Using tabular data and interpolants in OpenMDAO

We now have fast fixed interpolants

Be wary of using piecewise linear interpolants

Interpolant accuracy and efficiency matter

Please critically consider what interpolants you’re using, your data, and the balance of computational cost and accuracy

Thanks!

john.jasa@nasa.gov