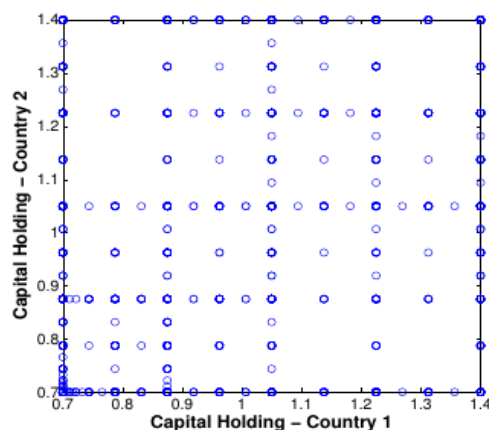


# Hands on Session (I) – Several Sparse Grid Toolboxes

Simon Scheidegger  
simon.scheidegger@gmail.com

July 16<sup>th</sup>, 2019

Open Source Macroeconomics Laboratory – BFI/UChicago

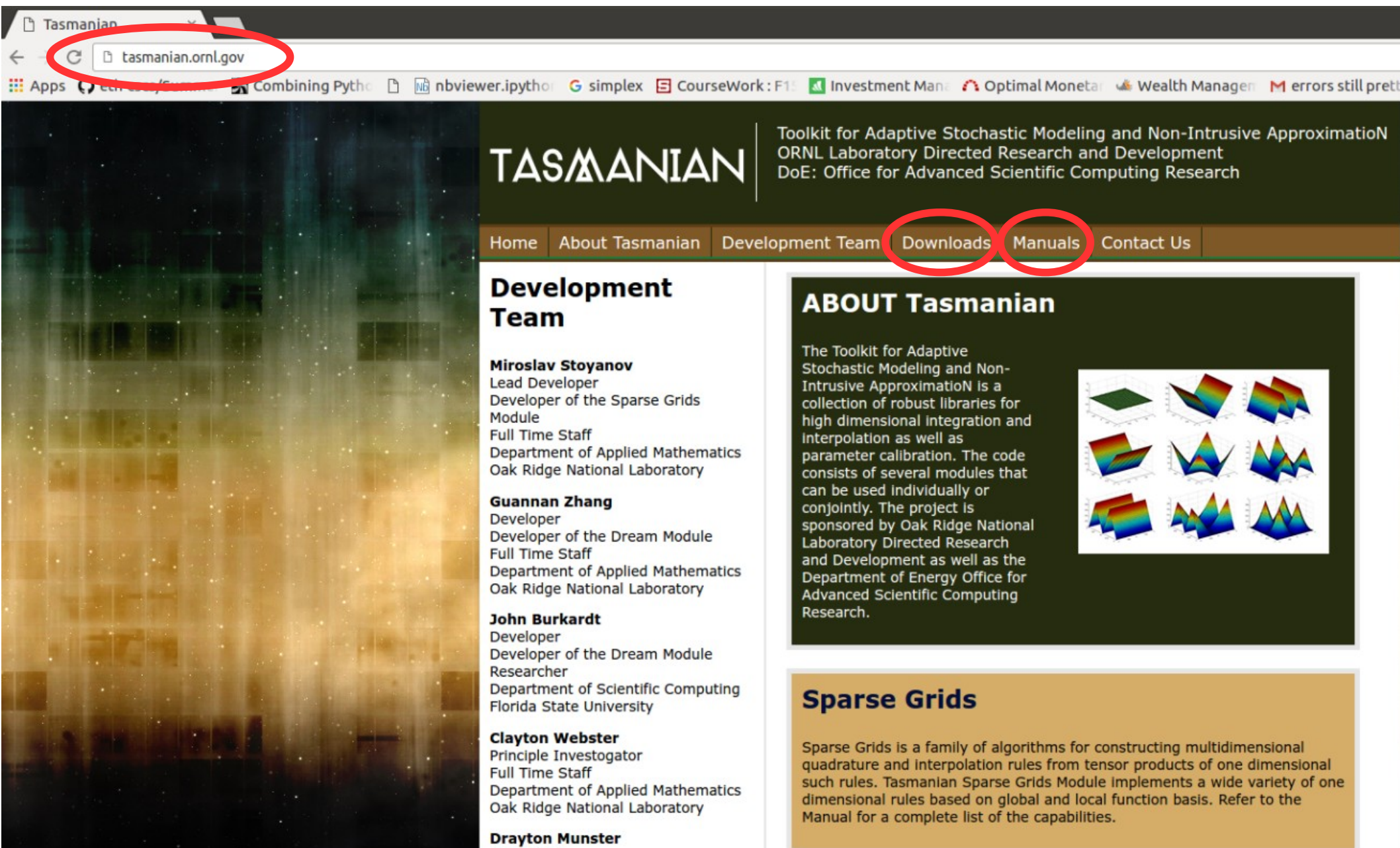


# Note – Install all libraries

- Note: I prepared you the packages with all the dependencies.
- in order to install, follow these steps:
  1. log onto MIDWAY, and go to your git repository.  
> `cd OSE2019/day1/SparseGridCode`
  2. Install Tasmanian (SG library), SPINTERP (SG library), IPOPT, and PYIPOPT (optimizer)

```
> install_SG.sh
```

# TASMANIAN – open source code



The screenshot shows a web browser window with the address bar displaying `tasmanian.ornl.gov`. The website header features the "TASMANIAN" logo and a description: "Toolkit for Adaptive Stochastic Modeling and Non-Intrusive ApproximationN ORNL Laboratory Directed Research and Development DoE: Office for Advanced Scientific Computing Research". The navigation menu includes links for Home, About Tasmanian, Development Team, Downloads, Manuals, and Contact Us. The "Downloads" and "Manuals" links are circled in red. The main content area is divided into three sections: "Development Team" listing team members, "ABOUT Tasmanian" with a description and a grid of 3D surface plots, and "Sparse Grids" with a brief description of the algorithm family.

Tasmanian

tasmanian.ornl.gov

Apps nbviewer.ipynb Combining Python simplex CourseWork: F1 Investment Man Optimal Monet Wealth Man errors still prett

**TASMANIAN**

Toolkit for Adaptive Stochastic Modeling and Non-Intrusive ApproximationN  
ORNL Laboratory Directed Research and Development  
DoE: Office for Advanced Scientific Computing Research

Home About Tasmanian Development Team **Downloads** **Manuals** Contact Us

### Development Team

**Miroslav Stoyanov**  
Lead Developer  
Developer of the Sparse Grids Module  
Full Time Staff  
Department of Applied Mathematics  
Oak Ridge National Laboratory

**Guannan Zhang**  
Developer  
Developer of the Dream Module  
Full Time Staff  
Department of Applied Mathematics  
Oak Ridge National Laboratory

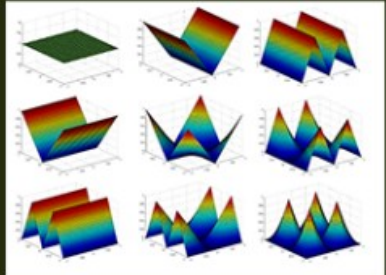
**John Burkardt**  
Developer  
Developer of the Dream Module  
Researcher  
Department of Scientific Computing  
Florida State University

**Clayton Webster**  
Principle Investigator  
Full Time Staff  
Department of Applied Mathematics  
Oak Ridge National Laboratory

**Drayton Munster**

### ABOUT Tasmanian

The Toolkit for Adaptive Stochastic Modeling and Non-Intrusive ApproximationN is a collection of robust libraries for high dimensional integration and interpolation as well as parameter calibration. The code consists of several modules that can be used individually or conjointly. The project is sponsored by Oak Ridge National Laboratory Directed Research and Development as well as the Department of Energy Office for Advanced Scientific Computing Research.



### Sparse Grids

Sparse Grids is a family of algorithms for constructing multidimensional quadrature and interpolation rules from tensor products of one dimensional such rules. Tasmanian Sparse Grids Module implements a wide variety of one dimensional rules based on global and local function basis. Refer to the Manual for a complete list of the capabilities.

# Software tutorial

The Toolkit for Adaptive Stochastic Modeling and Non-Intrusive Approximation  
<http://tasmanian.ornl.gov/>

## TASMANIAN Sparse Grids.

Very recent open source library written in CPP:

- Contains “ordinary and adaptive” sparse grids.
- Many more basis functions (global polynomials, wavelets,...).
- Interfaces to **Python** and **Matlab**.
  - **You better use it out of C++ or Python**
- Moderately parallelized (with OpenMP).

# Compile & run TASMANIAN\*

!!! READ THE \*\*\* MANUAL (RTFM) !!!

→ **Log on to MIDWAY**

> ssh USERNAME@midway2.rcc.uchicago.edu

→ **Load matlab**

> module load matlab

→ **Start MATLAB without graphical interface**

> matlab -nojvm

**1. go to simple example:**

> cd OSE2019/day1/SparseGridCode/analytical\_examples/TASMANIAN\_Matlab

**2. let's have a look at the example:**

> tsg\_example\_OSM\_19.m

**3. launch matlab & run example:**

> tsg\_example\_OSM\_19()

**4. NOTE: Tasmanian [-1,1]^d instead of [0,1]^d**

If you are interested in CPP code examples, TASMANIAN provides examples here:

> cd TasmanianSparseGrids/Example/example.cpp

> make

\*more examples provided from TASMANIAN in InterfaceMATLAB/tsgExample.m → beyond scope of lecture.

# TASMANIAN in Python

!!! READ THE \*\*\* MANUAL (RTFM) !!!

→ **Log on to MIDWAY**

> ssh USERNAME@midway2.rcc.uchicago.edu

**1. go to simple example:**

> cd OSM2019/day1/SparseGridCode/analytical\_examples/TASMANIAN\_Python

**2. let's have a look at the example:**

> OSM\_example.py

**3. run example:**

> python OSM\_example.py

**4. NOTE: Tasmanian  $[-1,1]^d$  instead of  $[0,1]^d$**



# Alternative Toolboxes (I)

<http://www.ians.uni-stuttgart.de/spinterp/>

**Sparse Grid Interpolation Toolbox**

The Sparse Grid Interpolation Toolbox is a Matlab toolbox for recovering (approximating) expensive, possibly high-dimensional multivariate functions.

It was developed by Andreas Klimke at the [Institute of Applied Analysis and Numerical Simulation](#) at the [High Performance Scientific Computing](#) lab ("Lehrstuhl für Numerische Mathematik für Höchstleistungsrechner"), [Universität Stuttgart](#) during his Ph.D. studies.

Andreas continues to maintain and improve the toolbox in his spare time since April 2006. He is very grateful to the group and, in particular, Prof. Dr. Wohlmuth for the possibility to continue to host the Sparse Grid Interpolation Toolbox on the institute's Web site.

For more information on sparse grid interpolation and the features of the toolbox, please go to the [About page](#).

Please note the [License](#) information.

When referencing the toolbox in a publication, [please cite these references](#).

**Latest news**

Date	Headline
May 25, 2008	<a href="#">Version v5.1.1 released</a>
February 24, 2008	<a href="#">Version v5.1.0 released</a>
December 23, 2007	<a href="#">Version v5.0.0 released</a>
October 24, 2007	<a href="#">Version v4.0.0 released</a>
March 3, 2007	<a href="#">Version v3.5.1 released</a>
July 25, 2006	<a href="#">Version v3.5.0 released</a>
June 12, 2006	<a href="#">Version v3.2.0 released</a>
January 30, 2006	<a href="#">Version v3.0.1beta released</a>
January 13, 2006	<a href="#">Sparse Grid Interpolation Toolbox Web page online</a>

Matlab is a registered trademark of The Mathworks, Inc.

# Alternative Toolboxes (II)

<http://www.ians.uni-stuttgart.de/spinterp/>

- spinterp.
- Matlab-based implementation of sparse grids
- Not updated since 2008
- Piecewise linear basis function and few others (global)
- Dimensional adaptivity as options
- **no general adaptivity**
- **Not parallel**



# Run Example Code on MIDWAY

→ **Log on to MIDWAY**

> ssh USERNAME@midway2.rcc.uchicago.edu

→ **Load matlab**

> module load matlab

→ **Start MATLAB without graphical interface**

> matlab -nojvm

→ **Go to example and run it.**

> cd OSE2019/day1/SparseGridCode/spinterp\_v5.1.1

> addpath('spinterp\_v5.1.1')

> spinit

> cd examples

> spdemo

```
%  
% A 2D-example for multi-linear sparse grid interpolation using the  
% Clenshaw-Curtis grid and vectorized processing of the function.  
%  
% See also SPINTERP, SPVALS.
```

```
% Author : Andreas Klimke, Universität Stuttgart  
% Version: 1.1  
% Date : September 29, 2003
```

```
% -----  
% Sparse Grid Interpolation Toolbox  
% Copyright (c) 2006 W. Andreas Klimke, Universitaet Stuttgart  
% Copyright (c) 2007-2008 W. A. Klimke. All Rights Reserved.  
% See LICENSE.txt for license.  
% email: klimkeas@ians.uni-stuttgart.de  
% web : http://www.ians.uni-stuttgart.de/spinterp  
% -----
```

```
% Some function f  
f = inline('1./((x*2-0.3).^4 +(y*3-0.7).^2+1)');
```

```
% Define problem dimension  
d = 2;
```

```
% Create full grid for plotting  
gs = 33;  
[X,Y] = meshgrid(linspace(0,2,gs),linspace(-1,1,gs));
```

```
% Set options: Switch vectorized processing on.  
options = spset('Vectorized', 'on', 'SparseIndices', 'off');
```

```
% Compute sparse grid weights over domain [0,2]x[-1,1]  
z = spvals(f, d, [0 2; -1 1], options);
```

```
% Compute inpterpolated values at full grid  
ip = spinterp(z, X, Y);
```

```
% Plot original function, interpolation, and error  
subplot(1,3,1);  
mesh(X,Y,f(X,Y));  
title('original');
```

```
subplot(1,3,2);  
mesh(X,Y,ip);  
title('interpolated');
```

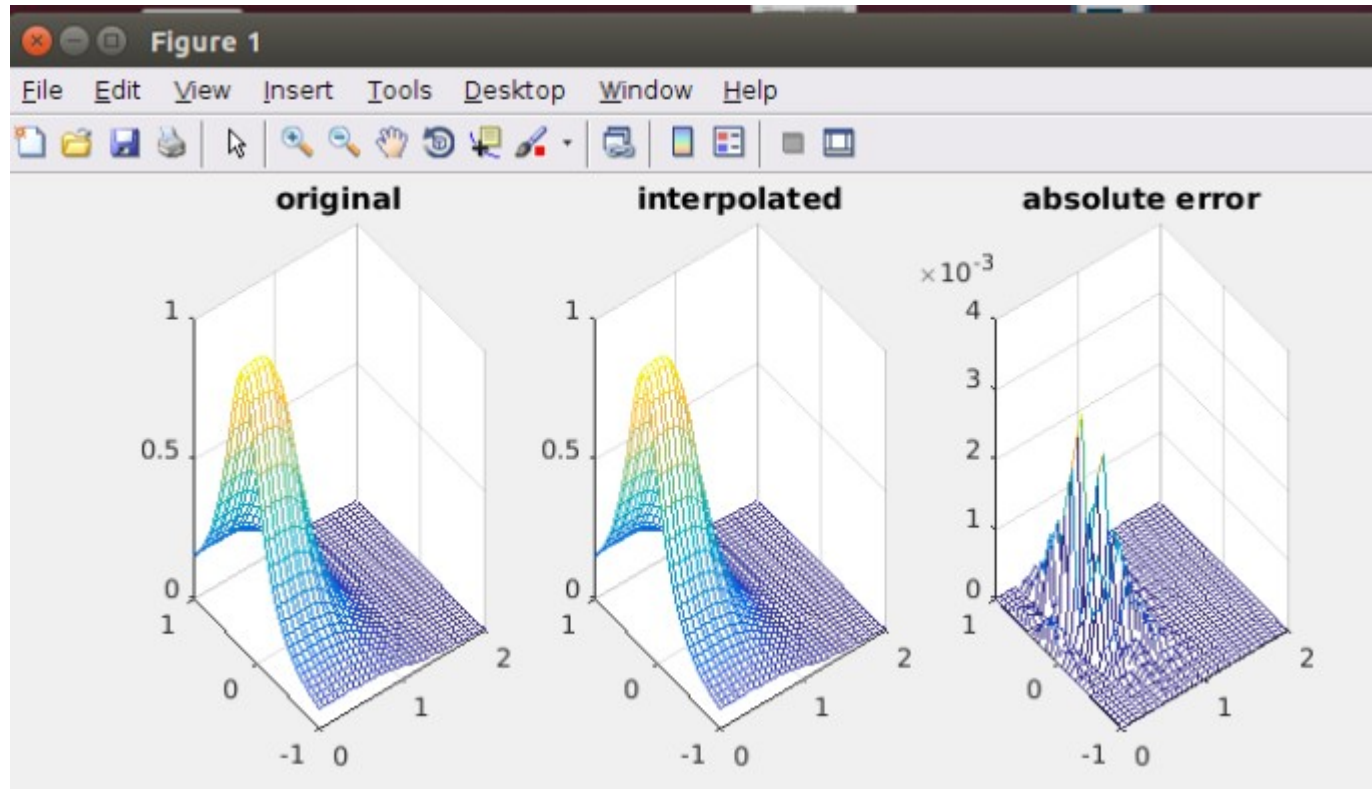
```
subplot(1,3,3);  
mesh(X,Y,abs(f(X,Y)-ip));  
title('absolute error');
```

```
disp(' ');  
disp('Sparse grid representation of the function:');
```

Test function

Interpolate

# What you should see...



# Other Toolboxes (III)

<http://sgpp.sparsegrids.org/>

- C++ with some plug-ins
- Multiple local basis functions

sgpp.sparsegrids.org

Apps eth-cscs/Summer Combining Python nbviewer.ipynb simplex CourseWork : F1 Investment Man Optimal Monet Wealth Man errors still pretty High Performance

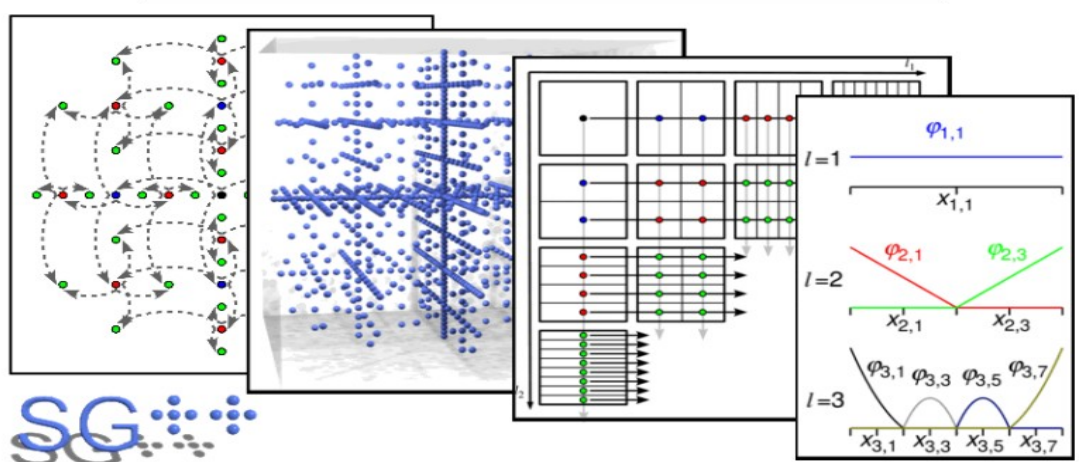
## SG++

Main Page Related Pages Namespaces Classes Files

### SG++ Documentation

Welcome to the SG++ documentation.  
The current version of SG++ can be found at [Downloads and Version History](#).

If you use any part of the software or any resource of this webpage and/or documentation, you implicitly accept the copyright (see the [Copyright](#)). This includes that you have to cite one of the papers dealing with sparse grids when publishing work with the help of SG++ (see below).



Images taken from [1]

[1] D. Pflüger, Spatially Adaptive Sparse Grids for Higher-Dimensional Problems. Verlag Dr. Hut, München, 2010. ISBN 9-783-868-53555-6.

### Overview