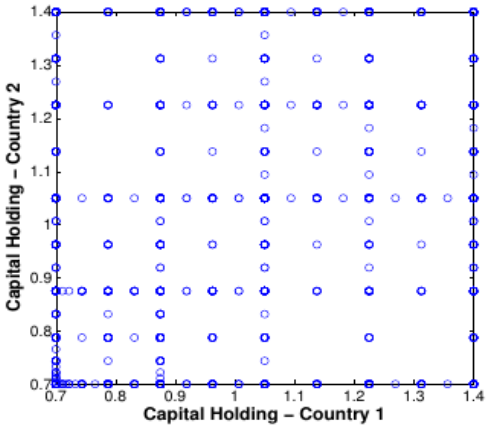
UNIL | Université de Lausanne

Hands on Sesson (I) – Several Sparse Grid Toolboxes

Simon Scheidegger
simon.scheidegger@gmail.com
July 16th, 2019
Open Source Macroeconomics Laboratory – BFI/UChicago



Note – Install all libraries

- Note: I prepared you the packages with all the dependencies.

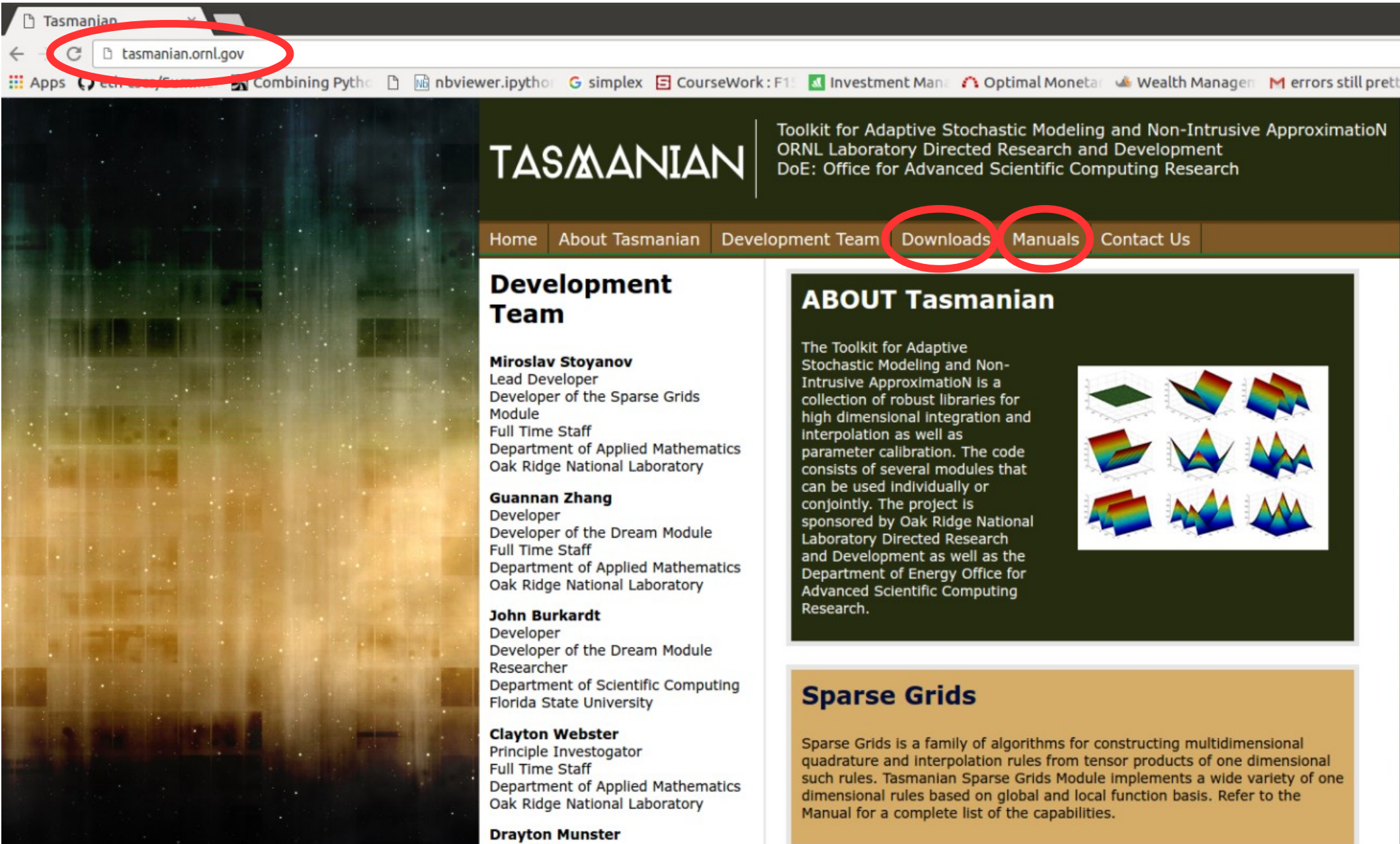
- in order to install, follow these steps:

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



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:

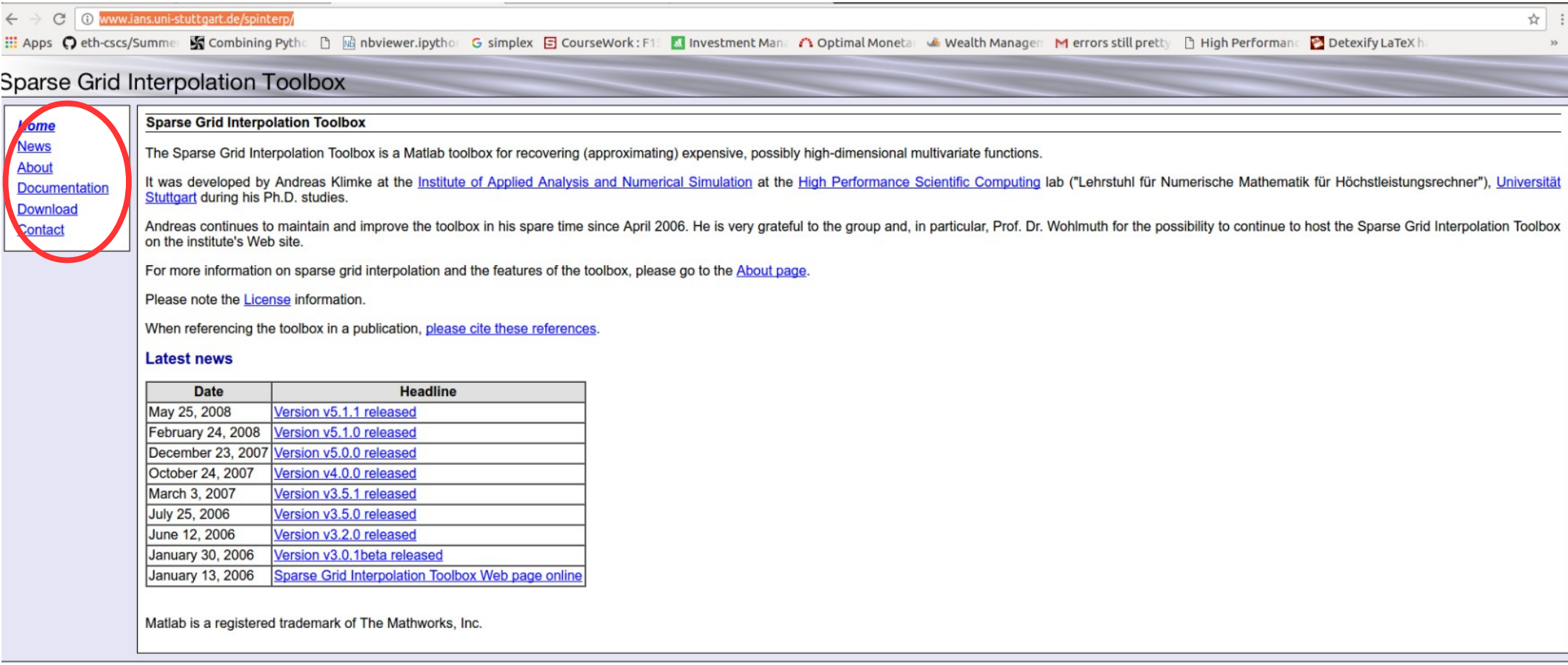
> python OSM_example.py

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

***more examples provided from TASMANIAN in InterfacePython/example.py → beyond scope of lecture.**

Alternative Toolboxes (I)

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



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.**

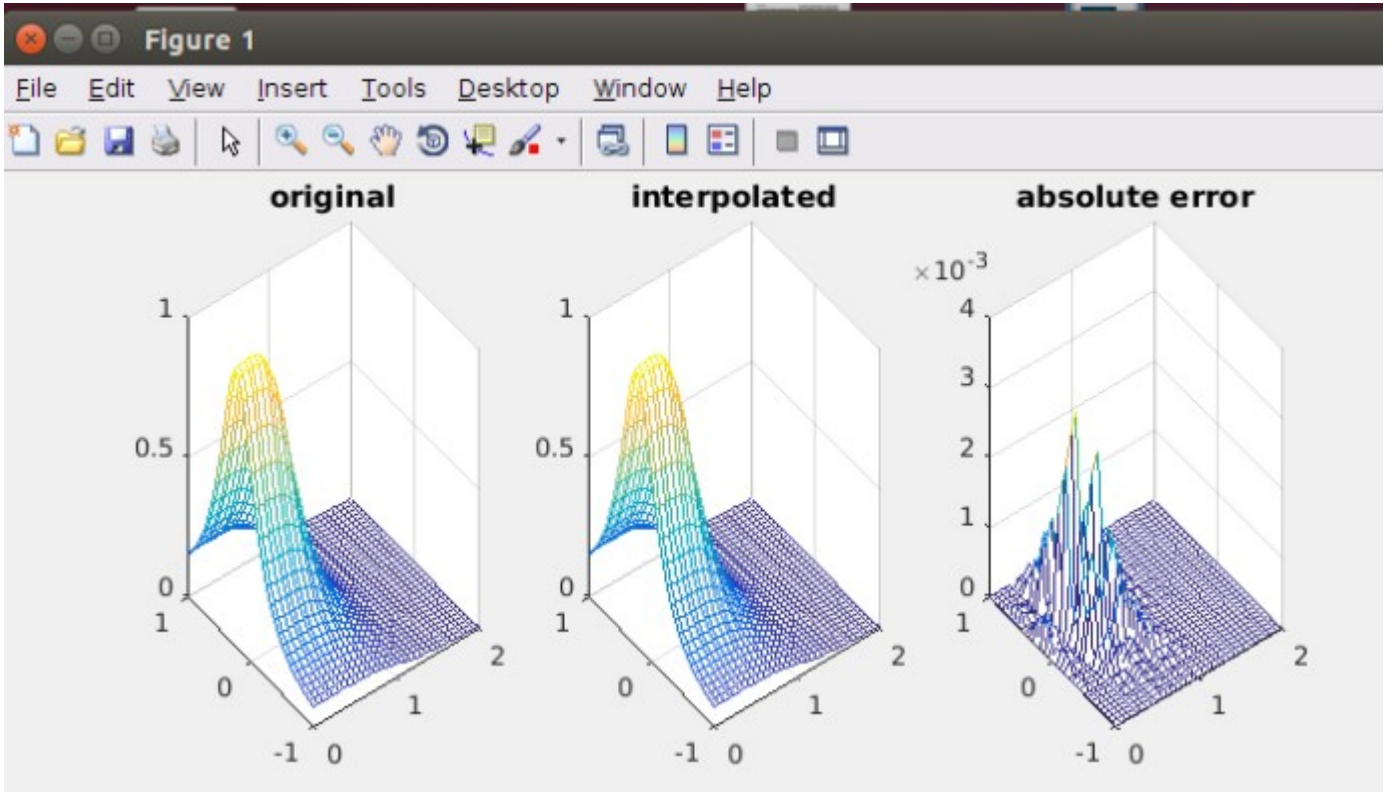
> cd OSE2019/day1/SparseGridCode/spinterp_v5.1.1
> addpath('spinterp_v5.1.1')
> spinit
> cd examples
> spdemo

```
%
% 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

SG++

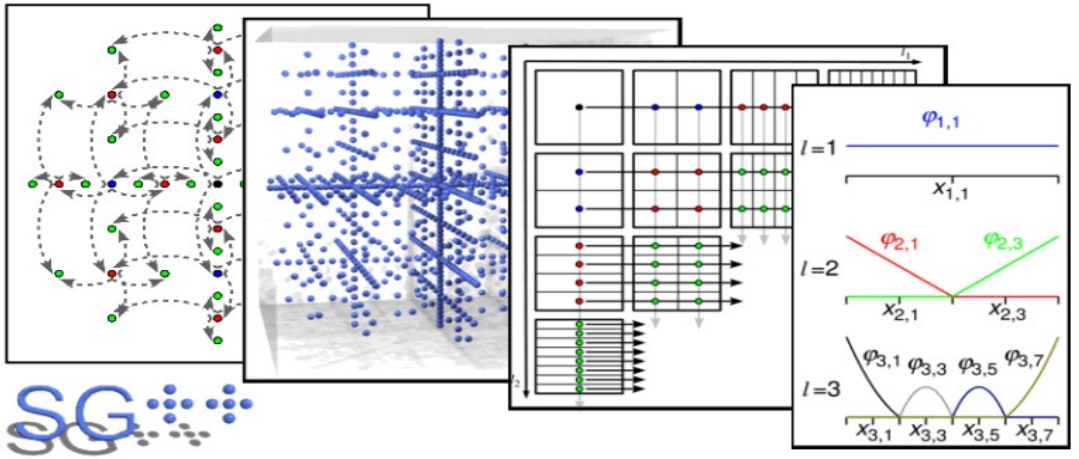
Main Page | Related Pages | Namespaces | Classes | Files

- SG++
  - SG++ Documentation
  - Todo List
  - 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