Spatial Analysis and Modeling Tool (SAMT), a spatial modeling toolkit written in Python

Ralf Wieland

January 25, 2016

SAMT

Is an integrated **spatial simulation toolkit:**

- Matlab: matrix operations, powerful visualization
- GIS: points, lines, polygons, raster with geographic coordinates, database connection
- Between Matlab and a GIS to support spatial simulations

SAMT

Is an integrated **spatial simulation toolkit:**

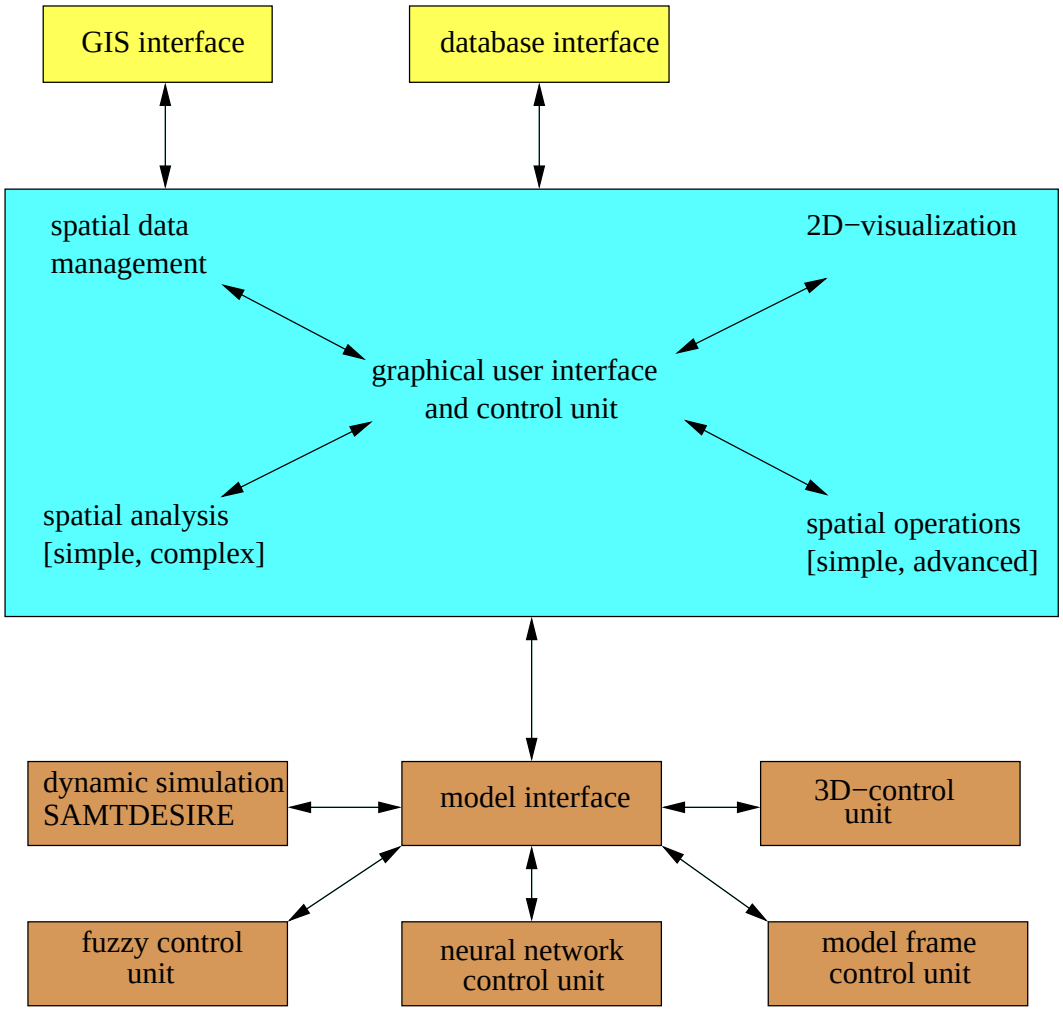
- Matlab: matrix operations, powerful visualization
- GIS: points, lines, polygons, raster with geographic coordinates, database connection
- Between Matlab and a GIS to support spatial simulations
- Open Source Project: https://github.com/Ralf3/samt2
- Toolkit consists of: SAMT, SAMTFUZZY, SADATO

SAMT

Is an integrated **spatial simulation toolkit:**

- Matlab: matrix operations, powerful visualization
- GIS: points, lines, polygons, raster with geographic coordinates, database connection
- Between Matlab and a GIS to support spatial simulations
- Open Source Project: https://github.com/Ralf3/samt2
- Toolkit consists of: SAMT, SAMTFUZZY, SADATO
  - Can be used with a graphical user interface,
  - as a part of a simulation on the computer cluster,
  - as a tool to build a bridge between the modeler and the stakeholder: "Open Science"

Structure of SAMT



Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

- Python is a well structured high level language,
  easy to learn and use, but slow

Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

- Python is a well structured high level language, easy to learn and use, but slow
- Way out: compile the "typed Python" code with **Cython** (about 30..80% of C/C++)

Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

- Python is a well structured high level language, easy to learn and use, but slow
- Way out: compile the "typed Python" code with **Cython** (about 30..80% of C/C++)
- Python comes with a set of **scientific modules**
- NumPy can be used to vectorization of programs, (Matlab)

```
x[x>6.35] = x[x>6.35]*28.33E-03
```

Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

- Python is a well structured high level language, easy to learn and use, but slow
- Way out: compile the "typed Python" code with **Cython** (about 30..80% of C/C++)
- Python comes with a set of **scientific modules**
- NumPy can be used to vectorization of programs, (Matlab)
  `x[x>6.35] = x[x>6.35]*28.33E-03`
- Matplotlib produces high quality scientific plots
- SciPy provides common scientific methods (solve differential eqns)
- **SciKit** includes nearly all methods of soft computing

Implementation

- C/C++ is more an assembler than a high level language
- Basic libraries are changing permanently (not compatible)

- Python is a well structured high level language, easy to learn and use, but slow
- Way out: compile the "typed Python" code with **Cython** (about 30..80% of C/C++)
- Python comes with a set of **scientific modules**
- NumPy can be used to vectorization of programs, (Matlab)
  `x[x>6.35] = x[x>6.35]*28.33E-03`
- Matplotlib produces high quality scientific plots
- SciPy provides common scientific methods (solve differential eqns)
- **SciKit** includes nearly all methods of soft computing
- Other modules helps for daily work: datetime, **hdf5**, pandas,...

Parts of SAMT2

- IO: ASCII, HDF, random
- Access: header, value, matc, matp

Parts of SAMT2

- IO: ASCII, HDF, random
- Access: header, value, matc, matp
- Visualization: show, showi, show3d, showbw, show_hist
- Statistics: hist, mean_std, min, max, unique, corr, sample
- Simple: norm, classify, replace, add, mul
- Complex: invert, lut, border, cut, varpart

Parts of SAMT2

- IO: ASCII, HDF, random
- Access: header, value, matc, matp
- Visualization: show, showi, show3d, showbw, show_hist
- Statistics: hist, mean_std, min, max, unique, corr, sample
- Simple: norm, classify, replace, add, mul
- Complex: invert, lut, border, cut, varpart
- Kernel: kernel_squ, kernel_cir, remove_trend
- Combination: mul, add, min, max, or

Parts of SAMT2

- IO: ASCII, HDF, random
- Access: header, value, matc, matp
- Visualization: show, showi, show3d, showbw, show_hist
- Statistics: hist, mean_std, min, max, unique, corr, sample
- Simple: norm, classify, replace, add, mul
- Complex: invert, lut, border, cut, varpart
- Kernel: kernel_squ, kernel_cir, remove_trend
- Combination: mul, add, min, max, or
- Point: transform, interpolation, voronoi, distance
- Flow: d4, d8, grad_d4, grad_d8, floodFill

Additionally: **full access to all scientific Python modules**

Graphical User Interface

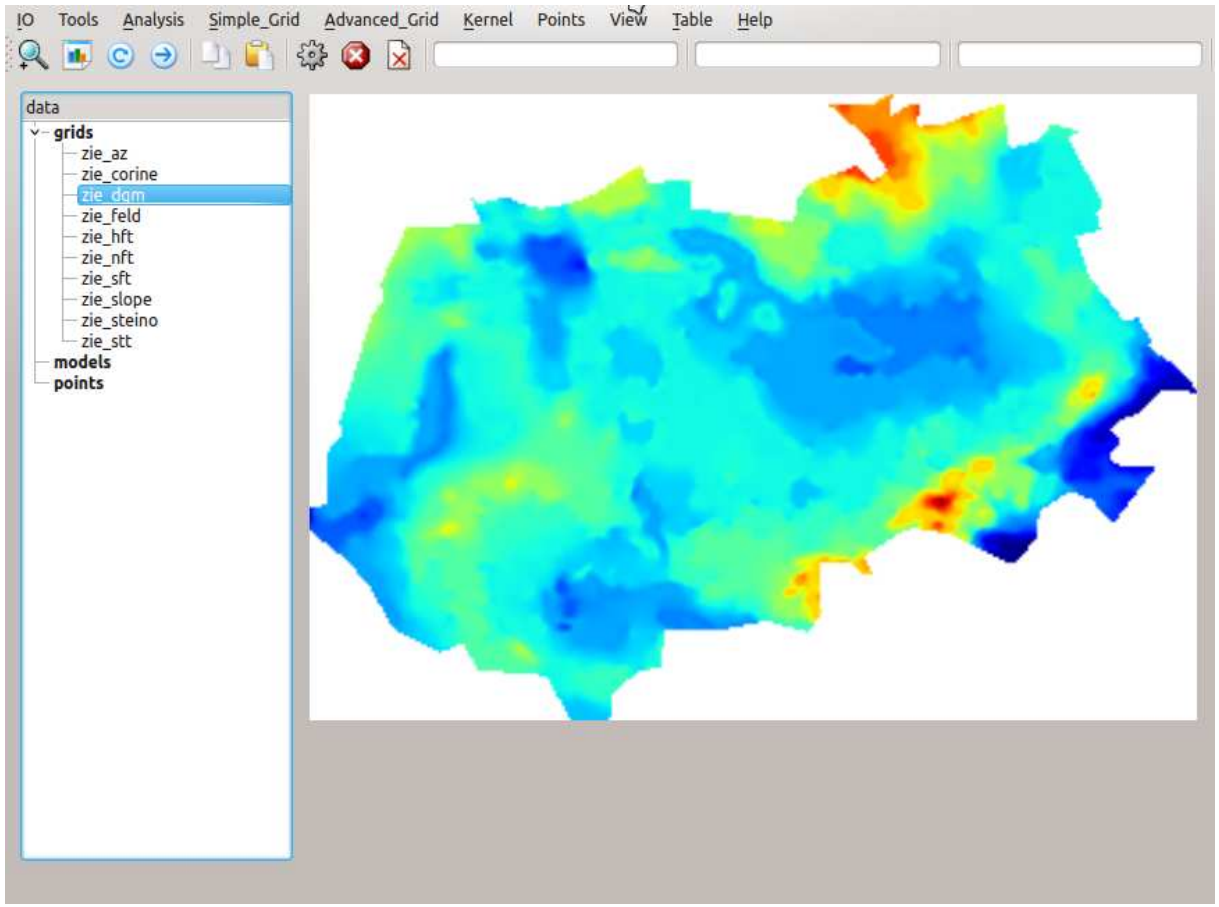


Figure 1: SAMT2 with some data

Graphical User Interface

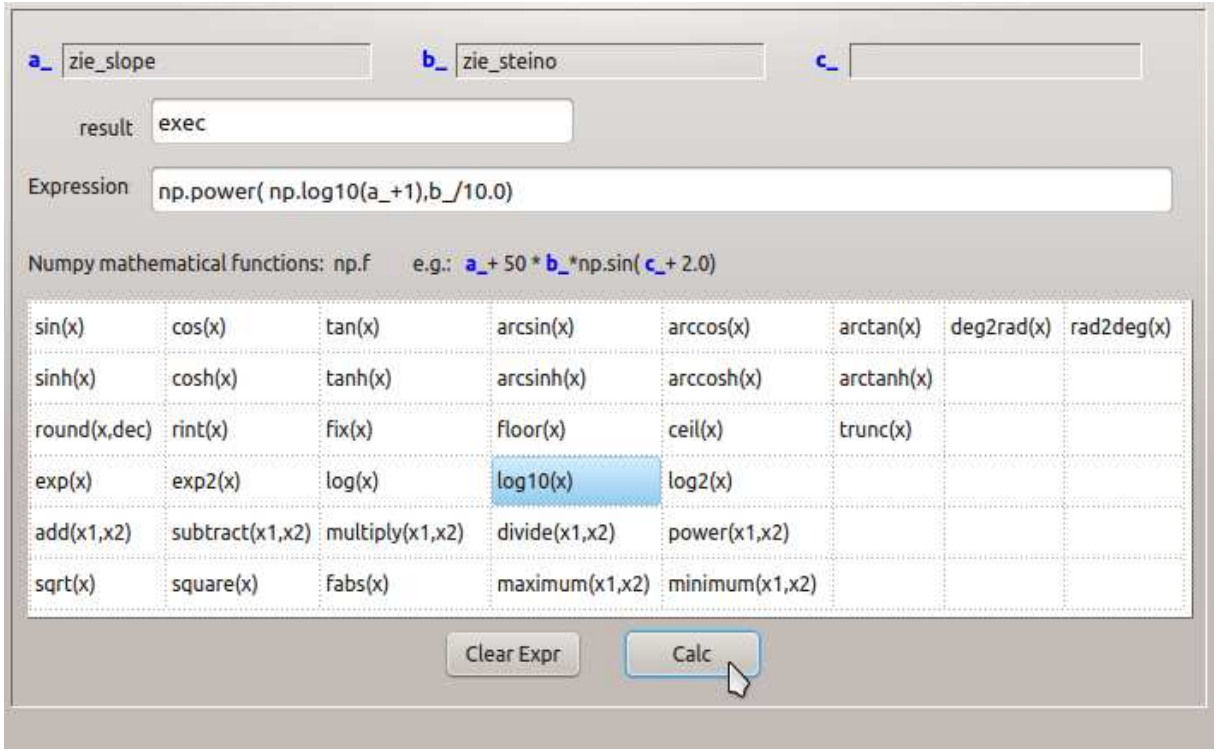
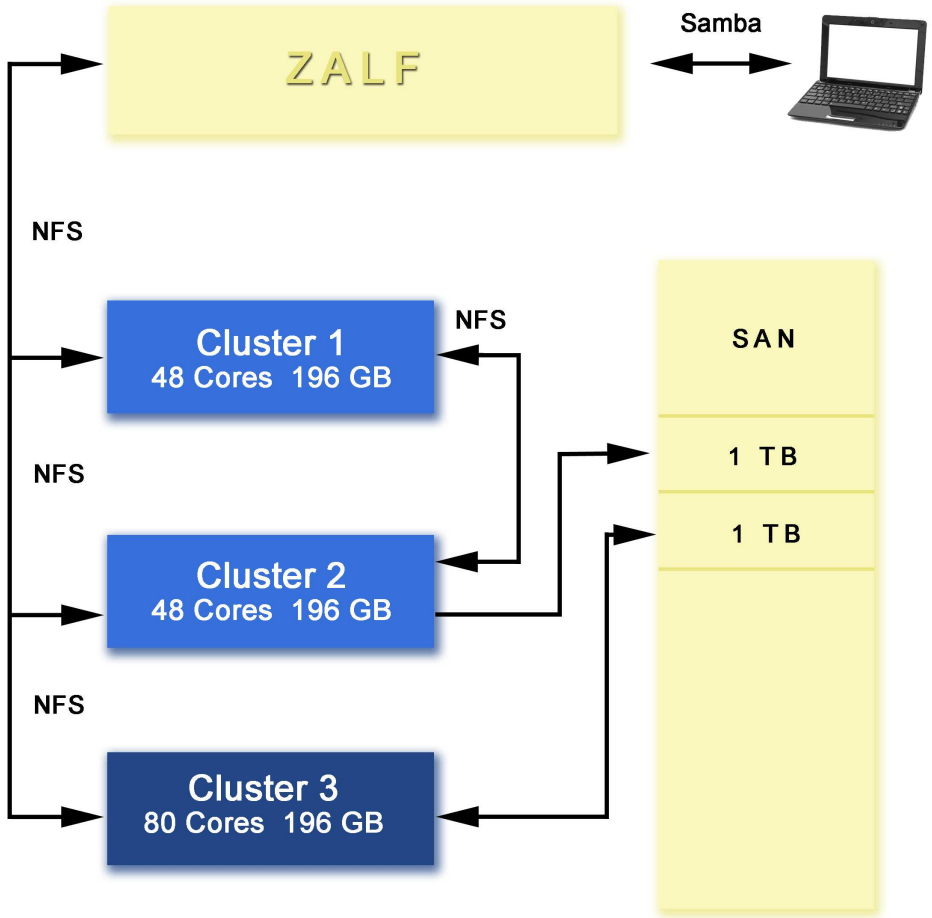


Figure 2: Expression builder of SAMT2

Graphical User Interface

Figure 3: Access to OSM data

Number crunching



Number crunching

- Application:
  - Yield estimation under climate change large regions in Germany (Brandenburg, Thuringia and Saxony with an grid size of 1ha)

Number crunching

- Application:
  - Yield estimation under climate change large regions in Germany (Brandenburg, Thuringia and Saxony with an grid size of 1ha)
  - Analysis of soil slices from X-Ray CT image stack

Number crunching

- Application:
  - Yield estimation under climate change large regions in Germany (Brandenburg, Thuringia and Saxony with an grid size of 1ha)
  - Analysis of soil slices from X-Ray CT image stack
  - Modeling of Mosquitoes movement and habitat quality for Mosquitoes of Germany

Number crunching

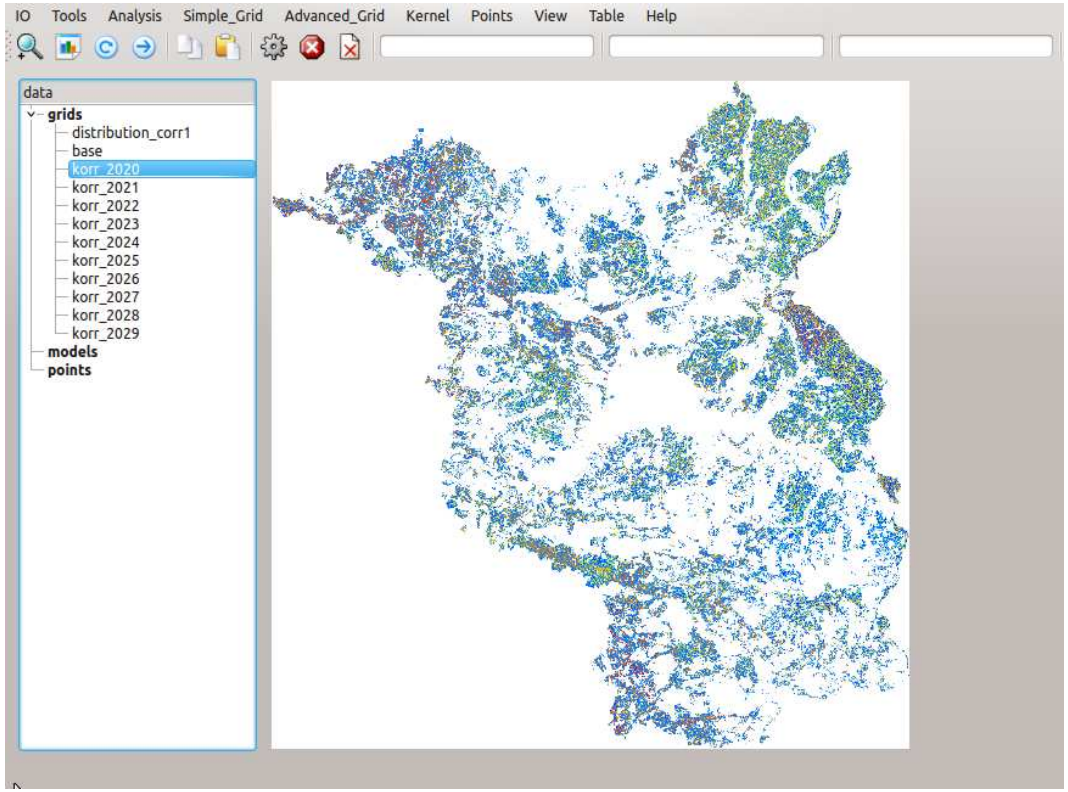
- Application:
  - Yield estimation under climate change large regions in Germany (Brandenburg, Thuringia and Saxony with an grid size of 1ha)
  - Analysis of soil slices from X-Ray CT image stack
  - Modeling of Mosquitoes movement and habitat quality for Mosquitoes of Germany
  - using a SVM to generate maps for controlling the fertilizer broadcaster

Number crunching

- Application:
  - Yield estimation under climate change large regions in Germany (Brandenburg, Thuringia and Saxony with an grid size of 1ha)
  - Analysis of soil slices from X-Ray CT image stack
  - Modeling of Mosquitoes movement and habitat quality for Mosquitoes of Germany
  - using a SVM to generate maps for controlling the fertilizer broadcaster
- Python controls the cluster using MPI
  load data, calc, collect data

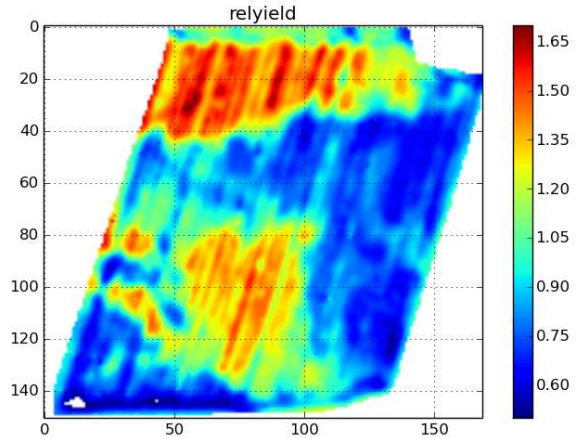
Number crunching

yield estimation based on market scenarios and climate change with 1ha resolution



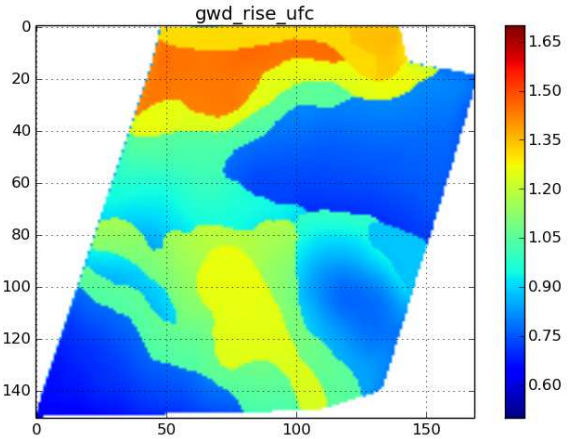
Number crunching

relyield from equipment



measured yield

SVM



**ground water distance,
capillary rise,
usable field capacity**

Open Science

Application: Can the user be included in modeling?

- Science is data collection plus modeling
- Applied science is an application of science to problems

Open Science

Application: Can the user be included in modeling?

- Science is data collection plus modeling
- Applied science is an application of science to problems
- We have the software and the interactive environment: "ipython notebook",
  - open source software,
  - documentation,
  - interactivity

Open Science

Application: Can the user be included in modeling?

- Science is data collection plus modeling
- Applied science is an application of science to problems
- We have the software and the interactive environment: "ipython notebook",
  - open source software,
  - documentation,
  - interactivity
- the stakeholders have knowledge (models) and knows the problems, why do we not include them in science?