InGrid

Innovative Grid-Entwicklungen für ingenieurs-wissenschaftliche Anwendungen

5. Bericht AP 2.5: Documentation FEFLOW® Grid Extension

Project Partner:

WASY Gesellschaft für wasserwirtschaftliche Planung und Systemforschung mbH
Waltersdorfer Straße 105
12526 Berlin

Editors:

Dipl.-Ing. Olaf Arndt (WASY GmbH)
Dipl.-Hyd. Udo Junghans (WASY GmbH)
Dipl.-Ing. Frank Thilo (University of Siegen)

Contents

Figures

Tables

1 Introduction

In the field of numerical groundwater modeling (typically based on Finite-Element methods), increasingly large areas and/or complicated hydro-geological structures have to be simulated and are often represented by complex 3-dimensional FEM meshes. Concurrently the performance increase of traditional single CPU systems is beginning to decelerate. Therefore, new techniques have to be used to solve complex groundwater modeling tasks on parallel computing resources.

Two principal ways to speed up groundwater simulation can be used: applying fine-grained parallelism in the core of the simulation software itself or to execute several simulations simultaneously, thus utilizing parallelism on a more coarse-grained level. The former can be used to potentially speed up any simulation, but requires changes to the solver code. The latter can be applied, whenever several simulations would have to be run consecutively in a traditional, sequential setup and the input of one simulation does not depend on the result of another. This paper will concentrate on the second parallelization approach.

During the groundwater modeling workflow, series of model applications are common at different lifecycles of the model. These include the calibration of model parameters, sensitivity analysis of parameters which have to be estimated, and technical model based optimization. A typical example for the last application field is to minimize operational costs of pumping stations while still satisfying constraints like a minimum allowed depth to water table or to meet certain groundwater quality thresholds. Each of the above tasks can lead to a large number of model simulation runs, which require no interconnects apart from modifying the parameters and returning the simulation results. Hence, these tasks are perfectly suited to be executed in parallel.

Grid technology [5] is a promising approach to implement a distributed resource infrastructure which is not limited to single clusters, sites or administrative boundaries. Based on the Service Oriented Architecture (SOA) [6] software engineering concept, several services like groundwater simulation, distributed optimization and sensitivity analysis can be implemented and deployed to resources in the Grid. To solve complex groundwater modeling tasks, these services are orchestrated to perform the necessary workflow and solve the required simulations on distributed Grid simulation resources. Different services can be provided by particular experts, e.g. an optimization expert will offer several optimization algorithms as a service, while a resource provider offers low-level computation services. Even software licensing may be offered as service to avoid the inflexibility of having each potential user to purchase the required licenses a-priori. Linking the service offers of different providers promises the potential to solve even complex and otherwise extremely time consuming groundwater engineering tasks in an adequate time.

The FEFLOW® Grid Extension brings groundwater modelling tasks onto Grid infrastructure. In order to realize sensitivity analysis or large optimization tasks groundwater modelling will be distributed on multiple resources to run large numbers of simulation processes in parallel.

This document describes the FEFLOW® Grid Extension structure, its configuration, and usage. Beside a general administration and usage guide, the configuration of a Grid job is explained using a concrete example.

2 System and Components

The FEFLOW® Grid Extension covers a set of Grid services, which must be installed and configured on the respective Grid resources, and some client components, which can be used to test or operate the system. Figure 1 gives an overview of the involved Grid nodes, as well as the interaction of the different services.

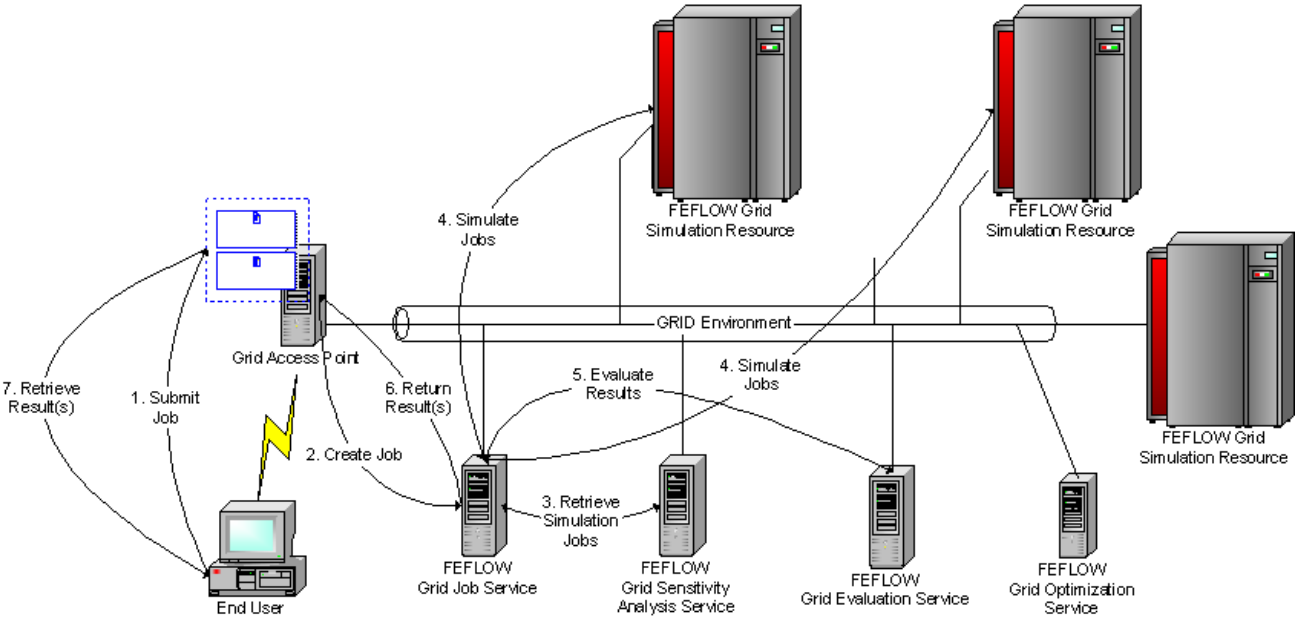

Figure 1: Overview of the FEFLOW Grid Services.

The interaction describes the workflow of a sensitivity analysis. Therefore, the end user submits the Grid job to the Grid Access Point. The Grid Access Point is in general the host running the FEFLOW Grid Job Service. The FEFLOW Grid Job Service uses in the case of sensitivity analysis the FEFLOW Grid Sensitivity Analysis Service to generate multiple FEFLOW simulation jobs based on the user's job definition. To solve other groundwater tasks like optimization it may contact a different service at this stage. Afterwards, the simulation jobs are distributed on the available FEFLOW Grid Resources which are calculating the simulation jobs. Finally, the simulation results are collected and evaluated by the FEFLOW Grid Evaluation Service.

2.1 FEFLOW Grid Client

The FEFLOW® Grid Client runs as an extension of the FEFLOW® GUI. It is required by the user for creating and managing the Grid job, but it is not required during job runtime for job processing. So, the client application can be terminated or restarted during job runtime without having any influence to the Grid job processing. Detailed information on how to use the FEFLOW® Grid client module are given in the user guide section 3.4 and 4.

2.2 FEFLOW Grid Job Service

The FEFLOW® Grid Job Service is the managing service of a complete Grid job. It acts as the point of contact to the FEFLOW® Grid client. It is used to initiate, control, and monitor Grid jobs on different Grid resources and retrieve the final results. When a new job is started, its task is to allocate and configure the required Grid resources and to start the single simulations calculated by the optimization service. It links the several services like sensitivity analysis service, optimization service, simulation service, and evaluation service to achieve a complete workflow. It provides also an interface to enable the client application to retrieve the current or overall optimization results.

2.3 Sensitivity Analysis Service

The sensitivity analysis service is a generic service, which provides different mathematical methods to generate a set of simulation jobs. The service itself does not implement any method, rather each method must be implemented as Python script based on a template. Therefore, it is easy to extent.

The provided methods can be lists of predefined values, min/max-ranges or any distribution like Gaussian distribution. But also random methods or complex statistical distributions are possible. s

2.4 Optimization Service

The optimization service is a separate Grid resource, which provides in general a set of different generic optimization algorithms, which are used to optimize a given groundwater problem.

The purpose of the Optimization Service is to control the optimization task. Therefore, it is responsible to calculate an initial set of solutions (candidate solution set) and to improve the achieved best solution by performing multiple iterations and to calculate an additional candidate solution set per iteration based on previous simulations results.

The Optimization Service is based on the Optix® suite. For further information refer the Optix® documentation.

Each currently active Grid optimization task is managed by one Optimization Job Manager. Initially, it sets up the Optimization Service. During the course of the optimization it repeatedly acquires new candidate solutions from the Optimization Service and distributes these to the FEFLOW® Grid Services for simultaneous evaluation, meaning a costly FEFLOW® simulation run for each. It collects the results of these evaluations and forwards them to the Optimization Service which in turn uses this information to create enhanced solutions. When no better results can be obtained, the overall result is sent to the FEFLOW® Grid Job Service and the job is terminated.

The Optimization Service consists of a set of different services or programs as follows:

1. The Optix® suite is a standalone application which is able to manage different optimization algorithms.

2. The Optix® suite Grid wrapper provides a SOA-based Grid service interface to the Optix® suite. It is a generic interface, which can be used for different optimization tasks.
3. The FEFLOW® Optimization Service uses the generic Optimization Service to generate the FEFLOW® model parameterization, but also the Evaluation Service to calculate generic results (objective function and constraints) as required by Optix® from the FEFLOW® simulation results.

2.5 Evaluation Service

The evaluation service is a separate Grid resource, which calculates generic results from groundwater relates results as retrieved by the simulation resource.

The evaluation service acts as link between the generic optimization definition (objective function and constraints) and the FEFLOW® specific simulation results. It calculates for example a single objective value (e.g. overall extraction rate) from multiple node or element based results (e.g. single extraction rates per well).

During job setup the user configures the objective function and additional constraints by setting up a parameter (e.g. extraction rate) and a method (e.g. summarize) for each. While the parameter definition is forwarded to the simulation resource, the method is handled by the evaluation service. To enable a high flexibility to extend the provided methods, each method can be described by a Python class implementation of a predefined template. The evaluation service provides SOA-compliant access to those methods including browsing the available methods and their parameters, setting up parameters and calculating the results by using the Python class implementation.

2.6 FEFLOW Grid Simulation Service

The FEFLOW® Grid Simulation Service resides on different computation Grid resources. In general, multiple computation resources are available for optimization, each with one or more clusters hidden behind a Grid enabled head node, and the service will be available on the corresponding head nodes.

The FEFLOW® Grid Simulation Service is used to evaluate single simulation jobs as calculated by the Sensitivity Analysis Service or the Optimization Service. To do so, the corresponding job scripts are executed by the FEFLOW® groundwater simulation on site-local resources, e.g. on a compute cluster. The service is responsible for invoking FEFLOW® via the local scheduling system to simulate the given finite element model suspect to parameters defined by the solution to be evaluated. After the simulation has finished it must collect the results and return them to the Grid Job Manager.

To do so, it uses the local GRAM or WS-GRAM installation of Globus Toolkit 4 to submit the jobs and notifications to forward job state changes to the Grid job manager. To choose GT2-style GRAM or GT4-style WS-GRAM

refer to chapter 3.1.4. After successful execution, the results are returned by GridFTP to the managing host.

3 Administration

In the following sections the installation and configuration of the WASY Grid Extension is described. If any compilation task should be performed, it is assumed that the Globus Toolkit is installed [3] and the corresponding environment is set. This will be done by executing the following commands (refer Globus Toolkit documentation):

export GLOBUS_LOCATION=/usr/local/globus

source $GLOBUS_LOCATION/etc/globus-devel-env.sh

3.1 Simulation Resource

To enable Grid-based FEFLOW® simulations, at least one simulation resources is required to run the FEFLOW® models. Therefore, some packages have to be installed, which enable the Grid-based FEFLOW® simulation. Furthermore the FEFLOW® Grid Extension provides different configurations for each simulation resource.

3.1.1 Requirements Head Node

The head node controls in general a computation cluster with several computation hosts. To do so, the Grid middleware has must be able to use the local batch scheduling system to submit jobs to a batch queue.

At least the following packages must be available on the head node:

- Sun Java JRE (recommended version >= 1.6)
- Globus Toolkit 4 (recommended version >= 4.0.5)
- Globus WSGRAM (requires Globus Toolkit version >= 4.0.5) or Globus GRAM installation with configured batch system (fork is also supported)
- WASY FEFLOW® Service (simulation service only) or WASY Grid Services (

If the services are available as source code the following packages are required to compile it:

- Sun Java SDK (recommended version >= 1.6)
- Apache Ant (recommended version 1.6.5)

3.1.2 Requirements Computation Nodes

In general, the computation nodes are controlled by a batch scheduling system and no Grid components are available on the computation nodes. Therefore, the FEFLOW® Grid Extension submits each single simulation job as stand-alone Python batch job. To do so, the following packages are required on each computation node:

- Python (version >= 2.5.1)
- X-Server (recommended Xvfb)

- OSF/Motif Runtime Libraries
- Python-enabled FEFLOW® installation [3] (version >= 5.3 P4)

3.1.3 Simulation Resource Installation

The WASY FEFLOW® Service is a subset of the complete WASY Grid Services package. It is recommended to install either the WASY FEFLOW® Service or the WASY Grid Services, not both. To install the complete package refer section 3.2.

To compile the WASY FEFLOW® Service change to the source code root directory. Type the following command to use the Ant build file *build.sim.xml* and to create the package:

ant –f build.sim.xml

```
Buildfile: build.sim.xml

init:
    [mkdir] Created dir: /homelocal/globus/tmp/testSource/feflow/build.sim
    [mkdir] Created dir: /homelocal/globus/tmp/testSource/feflow/build.sim/classes

...

copyBin:
     [copy] Copying 1 file to /homelocal/globus/tmp/testSource/feflow/tmp/gar
     [copy] Warning: Could not find file /homelocal/globus/tmp/testSource/feflow/deploy-client.wsdd to copy.
     [copy] Copying 1 file to /homelocal/globus/tmp/testSource/feflow/tmp/gar
      [jar] Building jar: /homelocal/globus/tmp/testSource/feflow/wasy_feflow_service.gar
   [delete] Deleting directory /homelocal/globus/tmp/testSource/feflow/tmp/gar

BUILD SUCCESSFUL
Total time: 43 seconds
```

The Ant build tool has now created the WASY FEFLOW® service package as *wasy_feflow_service.gar*.

The GAR archive can now be deployed on the head node by using the Globus deploy utility as globus user:

globus-deploy-gar wasy_feflow_service.gar

NOTICE: The WASY FEFLOW® Service is required by the FEFLOW® Grid Extension, however it is possible but not necessary to deploy it on the same host providing the GRAM/WSGRAM to the cluster resource (compare section 3.4.3).

```
Deploying gar file...

  Deploying gar with profile: <default>
  Created dir: /opt/globus-4/share/globus_wsrf_common/tmp/gar
…
  Copying 2 files to /opt/globus-4/lib
  Skipping fileset for directory /opt/globus-4/share/licenses. It is empty.
  deploying server config...
  Copying 1 file to /opt/globus-4/etc/wasy_feflow_service
  deploying JNDI config...
  Copying 1 file to /opt/globus-4/etc/wasy_feflow_service
  Deleting directory /opt/globus-4/share/globus_wsrf_common/tmp/gar

Deploy successful
```

To remove the WASY FEFLOW® Service use the Globus undeploy command:

globus-undeploy-gar wasy_feflow_service

3.1.4 Simulation Resource Configuration

To configure a FEFLOW® Grid simulation resource there are two possibilities. One possibility is to use the server-config.wsdd to configure the service defaults; the second one is to use a resource configuration file to configure a Grid job at submission time. To use the second approach refer section 3.4.3.

To configure the WASY FEFLOW® Service edit the corresponding Web Service Deployment Descriptor which can be found at the following location:

$GLOBUS_LOCATION/etc/wasy_feflow_service/service-config.wsdd

The descriptor contains a section *WASY/FEFLOW/FeflowService* to be edited. Some of the parameters of the section are general deployment parameters, which should not be modified. The configurable parameters are described in Table 1.

Table 1: Configuration parameters of the WASY FEFLOW® Service.

Name	Type	Description	Default
gramHost	GRAM WSGRAM (optional)	The parameter gramHost is required only, if the host providing GRAM or WS-GRAM differs from the host running the WASY FEFLOW® Service. To use another host uncomment the parameter and specify the full qualified host name.	localhost

Name	Type	Description	Default
jobManager	GRAM (required)	The parameter specifies the GRAM job manager used to submit jobs. The value depends on the system configuration of the GRAM host. Possible values are e.g. *jobmanager-fork* or *jobmanager-pbs*.	-
jobManagerPort	GRAM WSGRAM (optional)	The value describes the job manager port of the GRAM/WSGRAM host used for job submission.	GRAM: 2119 WSGRAM: 8443
useWSGram	(optional)	The parameter specifies whether the service should use the GT2 style GRAM or the GT4 style WS-GRAM. To use WS-GRAM at least the Globus Toolkit version 4.0.5 is required. Values are: • 1=True (WS-GRAM) • 0=False (GRAM)	0 (GRAM)
wsFactory	WSGRAM (optional)	Specifies the type of the managed job factory as supported by Globus (compare *ManagedJobFactoryConstants.FACTORY_TYPE*). Possible values are: • CONDOR • FORK • LSF • MULTI • PBS	Depends on the Globus Toolkit installation
wsQueue	WSGRAM (optional)	The parameter describes the name of the batch queue to be used by the job manager.	Defaults to the configured default queue of the batch scheduling system
pythonInterpreter	Comp. node (required)	Path to the Python interpreter executable as available on the computation nodes.	-
libPath	Comp. node (recommended)	The parameter specifies the library path as it will be used on the computation nodes. The path will be used as library search	Depends on the computation node con-

Name	Type	Description	Default
		path for the job execution (LD_LIBRARY_PATH) as well as Python module search path.	figuration
licenseManager	Comp. node (optional)	The parameter can be used as default license manager for computation jobs. If this parameter is specified the license host must be reachable by this name or IP from each computation host and the licenses are accessible by each admitted Grid user. A license proxy authorization as described in section **Fehler! Verweisquelle konnte nicht gefunden werden.** will not take place for the default license manager.	-
licenseOptions	Comp. node (optional)	Using this parameter the default type of the allocated FEFLOW license can be specified. Sample value is: *timeout=20,option=FMH3,model=floating*	-
enableCompressed	GRAM WSGRAM (optional)	If this parameter is set to *true* al simulation results are returned as *gzip*-compressed data.	false
commonTmp	Comp. node (optional)	If this parameter is set, temporary files (job description, results,…) are stored in the specified directory. This directory must be world read-/writeable and it must be accessible by the GRAM/WSGRAM-host as well as by each computation node. If the parameter is not specified, temporary data are stored in *~/.globus/feflow/tmp* of the corresponding Grid user.	User home

In addition to the Web Service Deployment Descriptor it may be necessary to modify the Xvfb-startup script. The script can be found as follows:

$GLOBUS_LOCATION/etc/wasy_feflow_service/config/scripts/startXvfb.py

Is the script is the full path to the Xvfb executable specified in the method *startVirtualFrameBuffer()*.

After modifying the deployment descriptor and the XVFB startup script the container should be restarted.

3.2 WASY Grid Services Resource

To enable FEFLOW® Grid usage, a main host is required to provide several services of the WASY Grid Extension. Currently, all required services are bundled in the WASY Grid Services package.

3.2.1 Requirements

The node running the required Grid services needs several packages to calculate the simulation parameters, to assemble the single simulation jobs and to evaluate the simulation results.

At least the following packages must be available on the resource:

- Sun Java JRE (recommended version >= 1.6)
- Globus Toolkit 4 (recommended version >= 4.0.5)
- Jython (recommended version >= 2.2)

If the services are available as source code the following packages are required to compile it:

- Sun Java SDK (recommended version >= 1.6)
- Apache Ant (recommended version 1.6.5)

3.2.2 Grid Service Resource Installation

The WASY Grid Services package contains all required services to enable Grid-based FEFLOW® jobs.

To compile the WASY Grid Services change to the source code root directory. Type the following command to use the Ant build file build. xml and to create the package:

ant -f build.xml

```
Buildfile: build.xml

init:
   [delete] Deleting: /homelocal/globus/projects/ingrid/feflow/deploy-server.wsdd
   [delete] Deleting: /homelocal/globus/projects/ingrid/feflow/deploy-jndi-config.xml
…
/homelocal/globus/projects/ingrid/trunk/feflow/deploy-client.wsdd to copy.
     [copy] Copying 1 file to /homelocal/globus/projects/Ingrid/feflow/tmp/gar
      [jar] Building jar: /homelocal/globus/projects/ingrid/feflow/wasy_Grid_services.gar
   [delete] Deleting directory /homelocal/globus/projects/ingrid/feflow/tmp/gar

all:

BUILD SUCCESSFUL
Total time: 1 minute 43 seconds
```

The Ant build tool has now created the WASY Grid services package as wasy_Grid_services.gar.

The GAR archive can now be deployed on the head node by using the Globus deploy utility as globus user:

globus-deploy-gar wasy_Grid_services.gar

```
Deploying gar file...
  Deploying gar with profile: <default>
  Created dir: /opt/globus-4/share/globus_wsrf_common/tmp/gar
  Skipping fileset for directory /opt/globus-4/share/globus_wsrf_common/tmp. It is empty.
…
  Creating Unix launcher script sample-Create
  Copying 1 file to /opt/globus-4/bin
  Deleting: /opt/globus-4/share/globus_wsrf_common/tmp/gar/etc/post-deploy.xml
  Deleting directory /opt/globus-4/share/globus_wsrf_common/tmp/gar
Deploy successful
```

To remove the WASY FEFLOW® Service use the Globus undeploy command:

globus-undeploy-gar wasy_Grid_services

NOTICE: The WASY Grid Services package includes also the WASY FEFLOW® Service. It is not recommended to deploy both packages on the same Globus installation.

3.2.3 Grid Service Resource Configuration

To configure the Grid Service Resource several parameters of the corresponding Web Service Deployment Descriptor must be modified. The descriptor file can be found as follows:

$GLOBUS_LOCATION/etc/wasy_Grid_services/service-config.wsdd

The file contains multiple sections, one section for each Grid service.

3.2.3.1 WASY FEFLOW Service Configuration

The configuration of the WASY FEFLOW® Service section in the deployment descriptor has to be done as described in section 3.1.4.

3.2.3.2 WASY Optimization Manager Service

The WASY Optimization Manager Service is required to map generic optimization information (decision variables, constraints, objectives…) to job scripts required by the WASY FEFLOW® Service. In general, no modifications of the corresponding section are required.

3.2.3.3 WASY Sensitivity Analysis Service

The WASY Sensitivity Analysis Service is required to generate parameterizations of single simulation by using a predefined method. In general, no modification of the corresponding section (WASY/FEFLOW/GeneratorService) parameters is required. In special cases the parameter of Table 2 can be modified, e.g. to enable a special user to modify corresponding Python scripts.

Table 2: Configuration parameters of the WASY Sensitivity Analysis Service.

Name	Type	Description	Default
templatePath	required	The parameter specifies the path to the Python scripts, which implement the methods used for job generation. It is possible to specify either an absolute path to the directory or a relative path related to $GLOBUS_LOCATION.	-

3.2.3.4 WASY Evaluation Service

The WASY Evaluation Service is required to aggregate or evaluate single simulation by using a predefined method. In general, no modification of the corresponding section (*WASY/FEFLOW/EvaluatorService*) parameters

is required. In special cases the parameter of Table 3 can be modified, e.g. to enable a special user to modify corresponding Python scripts.

Table 3: Configuration parameters of the WASY Evaluation Service.

Name	Type	Description	Default
templatePath	required	The parameter specifies the path to the Python scripts, which implement the methods used for evaluation. It is possible to specify either an absolute path to the directory or a relative path related to *$GLOBUS_LOCATION*.	-

3.2.3.5 WASY Grid Job Service

The WASY Grid Job Service is the central management service for each FEFLOW Grid job. It is responsible for creating the several simulation jobs by using the optimization or generator service, submitting the jobs to the simulation resources and collecting and evaluating the results by using the evaluation service. The Grid job service runs also a watchdog task per Grid job to detect simulation resource failure.

The parameters of the corresponding section (*WASY/FEFLOW/GridjobService*) which may be modified are described in Table 4.

Table 4: Configuration parameters of the WASY Grid Job Service.

Name	Type	Description	Default
restartCount	optional	This parameter specifies the number of restarts per simulation job in the case of failure. If the number of failures of a job exceeds the restartCount, the Grid job state will be set to *failed*.	10
watchdogCheck	optional	The watchdogCheck parameter specifies the period in seconds after which the watchdog checks, if jobs are still running or if they have terminated without *DONE* notification. If a job is no longer running (or existing) but no notification was received, the job gets a second chance (another watchdogCheck period) to respond. After this time, the job will be restarted automatically. A value of 0 disables the watchdog functionality.	disabled
enableBlockResources	optional	If the parameter is set to *true*, the service allocates as much simulation jobs per resource as nodes are specified. If it is set to *false*, a simulation job is allocated each time it is ready to run. In general, the block allocation is faster, because the Grid overhead (e.g. delegation) is minimized.	false

Name	Type	Description	Default
enableCompressed	optional	If the parameter is set to *true*, each job script will be created, stored and transferred as gzipped file. This reduces required storage and minimizes the transfer rate, but it slows down job creation and increases load of the Grid Job Service host.	False
embeddedFEM	optional	If this parameter is set to *true*, the FEM file is embedded in each simulation job script. Because this increases the data transfer it is not recommended to set this parameter to true.	False
jobPath	optional	If this parameter is set, all job related files are stored in the specified directory. To do so, the directory must be world read-/writeable. If the parameter is not specified, temporary data are stored in *~/.globus/feflow/jobFiles* of the corresponding Grid user.	User home
scriptPath	required	The parameter specifies the path to the Python scripts, which may be required by the service, e.g. to UU-encode he FEM file. It is possible to specify either an absolute path to the directory or a relative path related to *$GLOBUS_LOCATION*. In general, no modification of the parameter is required.	-
parameterPath	required	The parameter specifies the path to the Python scripts, which describe available FEFLOW parameters to be modified. It is possible to specify either an absolute path to the directory or a relative path related to *$GLOBUS_LOCATION*. In general, no modification of the parameter is required.	-
resultPath	required	The parameter specifies the path to the Python scripts, which describe available FEFLOW parameters to be read. It is possible to specify either an absolute path to the directory or a relative path related to *$GLOBUS_LOCATION*. In general, no modification of the parameter is required.	-

3.3 Optix® Optimization Service

To enable optimization of groundwater-related problems, the FEFLOW® specific WASY services rely on a generic optimization service. This service provides a couple of optimization algorithms which are suitable for solving real-valued, constrained, non-linear optimization problems. Furthermore, the algorithms are capable of generating several candidate solutions simultaneously thus allowing for parallel evaluation of the candidates in the Grid.

The GT4 optimization in turn relies on the Optix® optimization suite which provides the actual algorithms [1][4]. The service itself is merely a wrapper.

3.3.1 Requirements

The node which should provide the optimization service needs to be equipped with the following software packages:

- Sun Java JRE (recommended version >= 1.6)
- Globus Toolkit 4 (recommended version >= 4.0.5)
- APPSPACK (if APPS algorithm is desired, version 4.0)

If the services are available as source code the following packages are required to compile it:

- Sun Java SDK (recommended version >= 1.6)
- Apache Ant (recommended version 1.6.5)
- scons (recommended version >= 0.97)
- g++ (recommended version >= 4.0)

3.3.2 Optix Installation

To compile the Optix® optimization suite, change to the Optix® root directory. If the APPS algorithm should be included, the APPSPACK library must have been installed and the file SConstruct must be edited to adjust the value of the APPSHOME variable. Then type the following command to create the Optix® executable:

scons

```
scons: Reading SConscript files ...
Checking for C library pvm3... (cached) no
*** Warning: PVM_ROOT or PVM_ARCH not set
Checking for C library pvm3... (cached) no
*** Warning: no pvm, disabling all features using it (simbasedproblem)
Checking for C++ library ifm... (cached) no
*** Warning: FEFLOW not found, disabling it
Checking for C++ library appspack... (cached) yes
scons: done reading SConscript files.
scons: Building targets ...
g++ -o optmain.o -c -DHAVE_APPS -
DBASEPATH="/srv/home/globus/projects/ingrid/optix" -Isrc -
I/home/thilo/appspack/include/appspack optmain.cc
…
ranlib src/liboptix.a
g++ -o algo optmain.o src/liboptix.a -L/home/thilo/appspack/lib -lappspack
Install file: "algo" as "bin/algo"
scons: done building targets.
```

The scons tool has now built the Optix® executable bin/algo which can now simply be copied to its final destination, e.g. /opt/optix/bin/algo.

Remember this location because it is later needed to correctly configure the optimization service.

All of the above warnings can be safely ignored. In case a full Optix® installation with cluster computing support is desired, please refer to the included README file.

3.3.3 Optimization Service Installation

To compile the Optimization Service from source, change to the optservice root directory. Type the following command to create the gar package:

./globus-build-service.sh opt

```
Buildfile: build.xml

init:
   [delete] Deleting directory /srv/home/globus/projects/ingrid/optservice/build/lib
    [mkdir] Created dir: /srv/home/globus/projects/ingrid/optservice/build/lib

…
     [copy] Copying 1 file to /srv/home/globus/projects/ingrid/optservice/tmp/gar
      [jar] Building jar:
/srv/home/globus/projects/ingrid/optservice/de_siegen_service_opt.gar
   [delete] Deleting directory /srv/home/globus/projects/ingrid/optservice/tmp/gar

all:
BUILD SUCCESSFUL
Total time: 13 seconds
```

The Ant build tool has now created the optimization service package as de_siegen_service_opt.gar.

The GAR archive can now be deployed on the head node by using the Globus deploy utility as globus user:

globus-deploy-gar de_siegen_service_opt.gar

```
Deploying gar file...
  Deploying gar with profile: <default>
  Created dir: /opt/globus-4/share/globus_wsrf_common/tmp/gar
  Skipping fileset for directory /opt/globus-4/share/globus_wsrf_common/tmp. It is empty.
…
  Creating Unix launcher script sample-Create
  Copying 1 file to /opt/globus-4/bin
  Deleting: /opt/globus-4/share/globus_wsrf_common/tmp/gar/etc/post-deploy.xml
  Deleting directory /opt/globus-4/share/globus_wsrf_common/tmp/gar
Deploy successful
```

To remove the Optimization Service use the Globus undeploy command:

globus-undeploy-gar de_siegen_service_opt

3.3.4 Optimization Service Configuration

To configure the Optimization Service, one parameter of the corresponding Web Service Deployment Descriptor must be modified. The descriptor file can be found as follows:

$GLOBUS_LOCATION/etc/de_siegen_service_opt/service-config.wsdd

Table 5: Configuration parameters of the Optimization Service.

Name	Type	Description	Default
optixPath	required	The parameter specifies the path to the Optix® executable which is the backend providing the actual optimization algorithms and logic. It should be specified as an absolute path. The executable must be readable and executable by the globus user.	-

3.3.5 Optimization Service Test

To be able to verify that the service has been installed correctly and is working, a simple test client is provided which facilitates the optimization service to optimize a mathematical test function.

3.3.5.1 Building the Test Client

To build the client, change to the optservice root directory and issue the following command:

javac -classpath build/stubs/classes:$CLASSPATH de/siegen/client/opt/OptTestClient.java

There should be no output, except for that indicating success.

3.3.5.2 Performing the Test

To perform the test, invoke the test client like this (substitute myhost.mydomain with the actual hostname) as a Grid user:

java -classpath build/stubs/classes:$CLASSPATH
de.siegen.client.opt.OptTestClient
https://myhost.domain:8443/wsrf/services/siegen/OptFactory

```
CPUs: 10
new best: 10139.944025194798  (0.744779 -2.62469 -2.65872)
new best: 707.5999999999998  (1.2 -1.2 1.2)
new best: 45.05897059629217  (0.0328078 0.377651 -0.401028)
…
new best: 3.1702741644129695  (-0.0379909 -0.0314807 0.0969336)
new best: 3.1652487897191177  (-0.0366825 -0.0313889 0.0968821)
new best: 3.16467661868879  (-0.0391851 -0.0319931 0.0962782)
```

Whenever a new best solution is found, the client will print the newly found best objective function value and the corresponding values of all the decision variables. When the optimization has converged, the client will simply terminate. If there are no error messages and there is a visible progress of the solution quality over time, the test was successful. The progress of the optimization can also be followed in greater detail by inspecting the Optix® logfile, which can be found on the node hosting the service as /tmp/optixlog.<n> (.0, .1, etc.).

Besides the URI specifying the optimization service to use, the client understands the following parameters to customize its behaviour:

Table 6: Command line parameter of the optimization service test client.

Name	Type	Description	Default
-dim <n>	optional	The parameter specifies the dimension, i.e. the number of variables of the problem to solve. The problem is the extended Rosenbrock function with an additional constraint. The smallest useful value is 2. The problem becomes harder to solve with increasing number of variables.	3
-cpus <n>	optional	The number of virtual CPUs can be specified with this parameter. Because the evaluation of the solution candidates for this test is done directly within the test client and not on distributed resources in the Grid, no real CPUs are used. Instead the parameter sets the maximum number of solution candidates that will be generated at one time.	10
-algo <name>	optional	This parameter can be used to specify the desired optimization algorithm. By default, Optix provides DPS, PSS and APPS (only if installed with APPSPACK).	DPS

Service URI	required	This is the URI of the Optimization Service factory, e.g.: *https://myhost.domain:8443/wsrf/services/siegen/OptFactory*	-

3.4 WASY Grid Client

To enable a user to use the FEFLOW Grid functionality within the FEFLOW GUI, the user requires the WASY Grid IFM module. Until the module will be included in the FEFLOW installation it must be compiled using the source distribution.

3.4.1 Requirements

The node running the WASY Grid client module requires several packages to enable the module usage.

At least the following packages must be available on the resource:

- Sun Java JRE (recommended version >= 1.6)
- FEFLOW installation (version >= 5.3)

If the services are available as source code the following packages are required to compile it:

- FEFLOW 5.3 SDK
- Sun Java SDK (recommended version >= 1.6)
- Globus Toolkit 4 (recommended version >= 4.0.5)
- Jython (recommended version >= 2.2)
- Trolltech QT (recommended version >= 4.3.2)

It is also required that the ***CLASSPATH*** contains the ***jython.jar*** archive and the ***QT bin*** directory is in the search path. Furthermore the WASY FEFLOW SDK has been compiled.

3.4.2 WASY Grid Client Installation

The WASY Client Module package contains all required files and archives to run the IFM module. Beside the IFM module the Globus Toolkit JAR archives are also copied into the package, so no separate Globus Toolkit installation is required on the client host.

To compile the WASY Client Module change to the source code root directory. Type the following command to use the Ant build file *build.client.xml* and to create the package:

ant -f build.client.xml

```
Buildfile: build.client.xml

init:
    [mkdir] Created dir:
/homelocal/globus/projects/beinGrid/be06/trunk/feflow/build.client
    [mkdir] Created dir:
/homelocal/globus/projects/beinGrid/be06/trunk/feflow/build.client/client_module
    [mkdir] Created dir:

...

    [bzip2] Building:
/homelocal/globus/projects/beinGrid/be06/trunk/feflow/wasy_client_module.tar.bz2

build_client:

BUILD SUCCESSFUL
```

To install the FEFLOW Grid Client Module change to the destination directory (e.g. /opt/wasy) and extract the generated archive wasy_client_module.tar.bz2 on the desired client node by executing the following command:

tar jxvf wasy_client_module.tar.bz2

3.4.3 WASY Grid Client Configuration

To configure the Grid client module, two steps are required. The first step is to set a proper library and class path. The class path must include all required java classes of the Globus Toolkit as well as the compiled FEFLOW Grid Client classes. The library path must include the client wrapper library. It is recommended to use the provided client module start-up script (…/wasy_client_module/bin/feflow) or to modify the FEFLOW start-up script (*/usr/bin/feflow*).

As the second step, the available Grid resources must be configured. The resources are described by XML files. To enable the IFM module to find the appropriate files, the environment variable Grid_RESOURCE_CONFIG must point to the directory, which contains the XML files. Sample resource files are installed by the WASY Grid Service installation in the directory
$GLOBUS_LOCATION/etc/wasy_Grid_services/config/demoScripts.

Each XML file describes a complete Grid resource environment with all required service hosts. The corresponding parameters are described in Table 7.

Table 7: Parameter description of the XML resource config file.

Tag	Parameter	Type	Description
resourcelist		Required	The *resourcelist* tag covers the complete resource description. Currently only the first resource list tag of the file is used. To specify multiple resources use different files.
	Description	Required	The description parameter is displayed in the WASY Client module to select the desired Grid resource. It has no additional purpose.
mainhost		Required	The *mainhost* tag describes the simulation resource run-

Tag	Parameter	Type	Description
			ning the WASY Gridjob Service. It is the central Grid management instance.
	address	Required	The address parameter specifies the host name and service path, e.g. *https://hostname.domain:8443/wsrf/services*.
opthost		Optional	The opthost parameter describes the optimization resource. It is required for optimization task.
	address	Optional	The address parameter specifies the host name and service path, e.g. *https://hostname.domain:8443/wsrf/services*.
evalhost		Required	The evalhost parameter describes the resource to manage evaluation methods. It is required for all Grid-based tasks.
	address	Required	The address parameter specifies the host name and service path, e.g. *https://hostname.domain:8443/wsrf/services*.
simhost		Required	The simhost parameter describes the resource running the WASY FEFLOW Service. Currently only one simhost entry per Grid resource description is possible, but it is possible to specify multiple resource entry to be managed by this service instance.
	address	Required	The address parameter specifies the host name and service path, e.g. *https://hostname.domain:8443/wsrf/services*.
resource		Required	At least one resource must be configured. The resource block specifies a resource, which is used for simulation. The resource is accessed from the simulation host by GRAM or WS-GRAM only. It is possible to specify an arbitrary number of resource entries, but the GRAM host entry of each resource block must be unique.
	wsgram	Required	The flag specifies, whether the GRAM (=0) or WS-GRAM (=1) should be used.
	gramhost	Required	The gramhost parameter specifies the host name of the host running GRAM or WS-GRAM.
	jobmanager	Optional	The parameter specifies the job manager which will be used for job submission (refer Globus Toolkit documentation), e.g. jobmanager-pbs. The parameter is required for GRAM only.

Tag	Parameter	Type	Description
	factory	Optional	The parameter specifies the factory type, e.g. PBS (refer Globus Toolkit documentation). The parameter is required for WS-GRAM only.
	jobmanagerport	Optional	The parameter specifies the port of the job manager or the job manager factory (refer Globus Toolkit documentation). If the port is not specified, the default port 2119 will be used.
	Nodes	Required	Specifies the number of nodes available for this resource. It may be the real number of cores of the corresponding cluster, but it may be also higher to use queuing or lower to use just a subset of the resource nodes.
	Xvfb	Optional	The parameter describes the path to Xvfb startup script on the GRAM or WS-GRAM resource. If no script is specified, the configured script of the WASY FEFLOW Service will be used.
	libPath	Optional	The parameter describes the library path of the GRAM or WS-GRAM resource. If no path is specified, the configured script of the WASY FEFLOW Service will be used.
	python	Optional	The parameter describes the path of the python executable on the GRAM or WS-GRAM resource. If no path is specified, the configured executable of the WASY FEFLOW Service will be used.
	licenseManager	Optional	The parameter describes the one or multiple license managers, which are accessible from the GRAM or WS-GRAM resource. If no license manager is specified, the configured license manager of the WASY FEFLOW Service will be used. The parameter syntax is as follows: *Hostname.domain[[:port][+hostname.domain[:port]…]]*
	licenseOptions	Optional	The parameter specifies the default license which will be used (if available). If this parameter is not specified, the licenseOptions of the WASY FEFLOW Service will be used. The parameter syntax is as follows (refer FEFLOW documentation): timeout=20,options=FMH3,model=floating
	maxWallTime	Optional	The parameter describes the maximum wall time of a simulation job on the GRAM or WS-GRAM resource. If no maximum wall time is specified, the configured wall time of the WASY FEFLOW Service will be used. ***NOTICE***: The valid value of this parameter depends on the batch scheduling system on the remote resource. If the specified maximum wall time is larger than the maximum wall time of the scheduling system, job execution may be rejected.

Tag	Parameter	Type	Description
	maxCPUTime	Optional	The parameter describes the maximum CPU time of a simulation job on the GRAM or WS-GRAM resource. If no maximum CPU time is specified, the configured CPU time of the WASY FEFLOW Service will be used. ***NOTICE***: The valid value of this parameter depends on the batch scheduling system on the remote resource. If the specified maximum CPU time is larger than the maximum CPU time of the scheduling system, job execution may be rejected.
	minMemory	Optional	The parameter describes the minimum memory of a simulation job required on the GRAM or WS-GRAM resource. If no minimum memory is specified, the configured memory of the WASY FEFLOW Service will be used. ***NOTICE***: If the specified minimum memory is larger than the memory provided by any node, job execution may be delayed forever.
	queue	Optional	The parameter describes the name of the queue which should be used for a simulation job on the GRAM or WS-GRAM resource. If no queue is specified, the configured queue of the WASY FEFLOW Service will be used. ***NOTICE***: Whether the user is allowed to use the queue or not depends on the scheduling system. In general no queue should be specified. This enables the batch scheduler to select an appropriate queue.

4 User Guide

4.1 IFM-Module Usage

FEFLOW allows expanding its capabilities by using an open programming interface called Interface Manager (Diersch 2007). So users are capable to create extension on their own or to use existing extension. These extension called modules are physically Shared Libraries or Dynamic shared objects (DSO) under Linux or Dynamic Link Libraries (DLL) under Microsoft Windows. Detailed information on FEFLOW Interface Manager (IFM) can be obtained by the FEFLOW developer documentation.

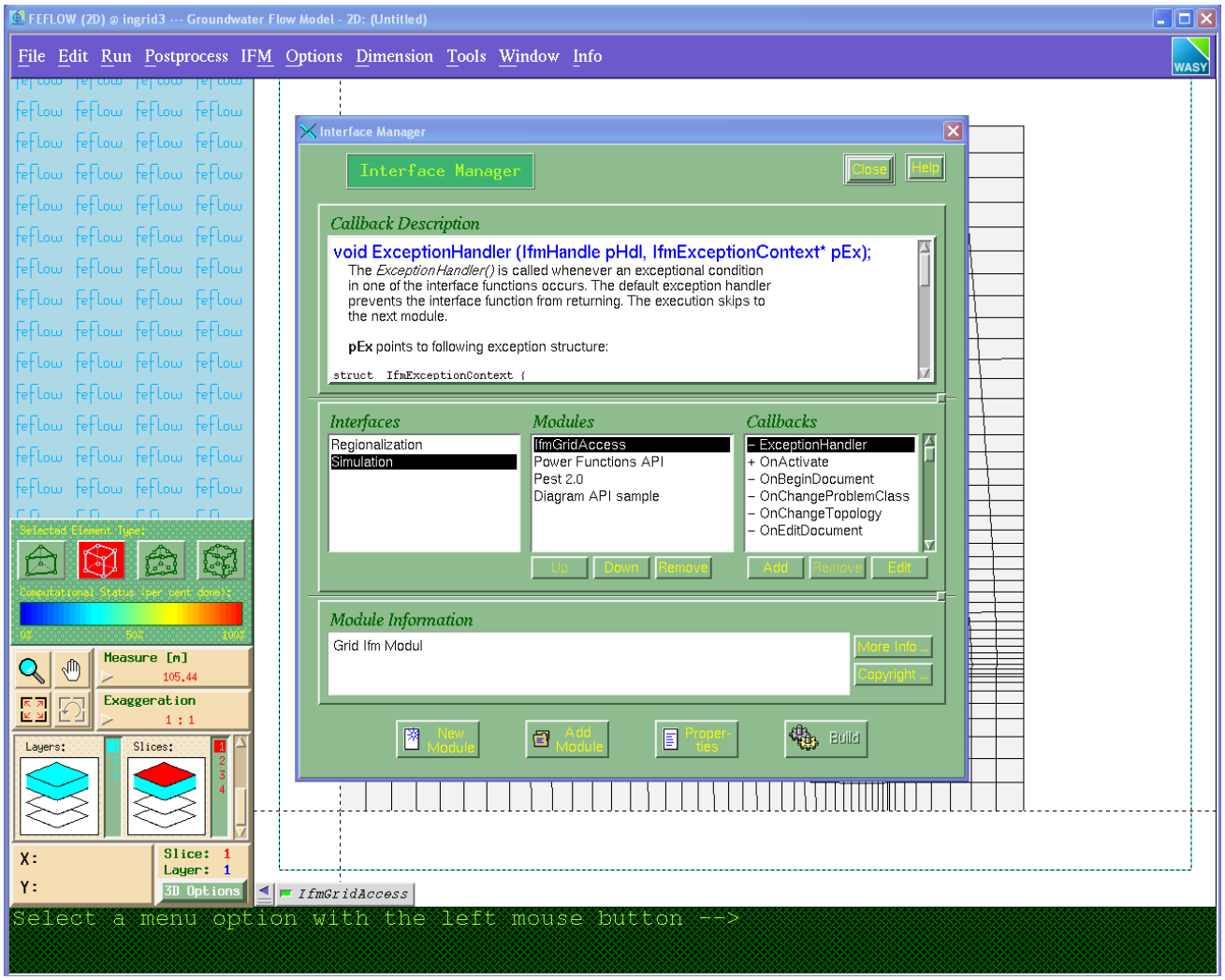

Figure 2: Configuration dialog of the FEFLOW Interface Manager.

FEFLOW Interface Manager was used to provide a graphical user interface for FEFLOW-Grid access. Embedded in the environment to which the user is used to FEFLOW-Grid module offers a graphical user interface which supports the user in getting certificates, in defining the Grid job, in choosing the Grid resources, in establishing the Grid job and in getting information about the actual status of the jobs running. In Figure 2 the

configuration of the FEFLOW-Grid module using the IFM configuration dialog is shown.

4.1.1 Module access and certificate management

The IFM-Module has to be available as loadable DLL resp. DSO module as described in the FEFLOW User Manual (Diersch 2007). It is recommended to load the FEFLOW-Grid module with the FEM-File to be used for the Grid-job. Therefore, the module must be registered to the FE model as shown in Figure 3.

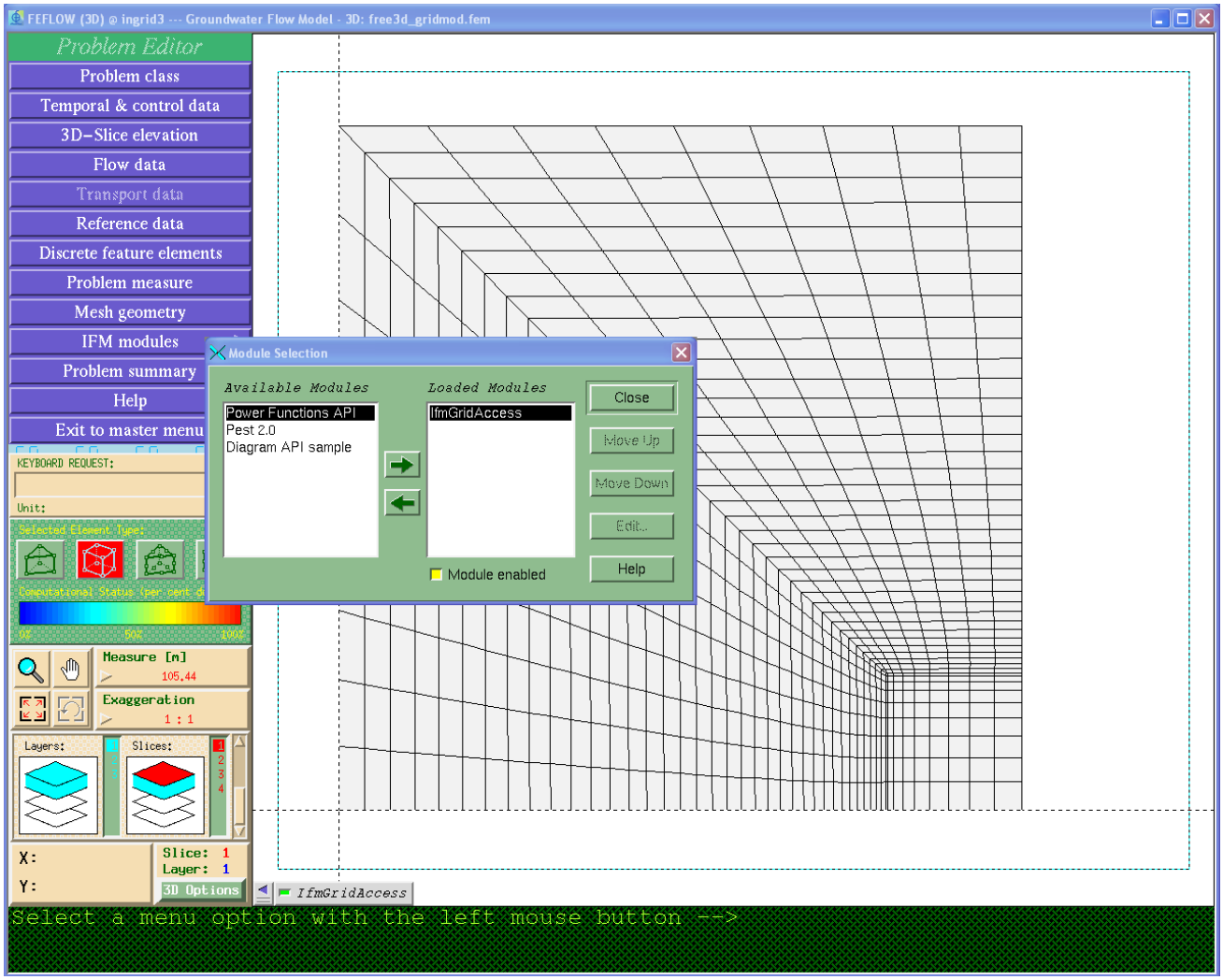

Figure 3: IFM module selection dialog.

Main information for the job definition is read from the FEM file directly. To access the FEFLOW Grid job GUI enter the activated module. A dialog for the certificate management appears.

4.2 Configuring Grid Jobs

4.2.1 Choosing Grid resources

FEFLOW-Grid capabilities are available on different Grid resources. So the user has to define which resources have to be used. To enable easy resource selection by the user, the available resources have to be preconfigured by the administrator as described in section 3.4.3. Afterwards the user can select an available *Grid Access Point* (including the whole configuration) by selecting the corresponding description as shown in Figure 4.

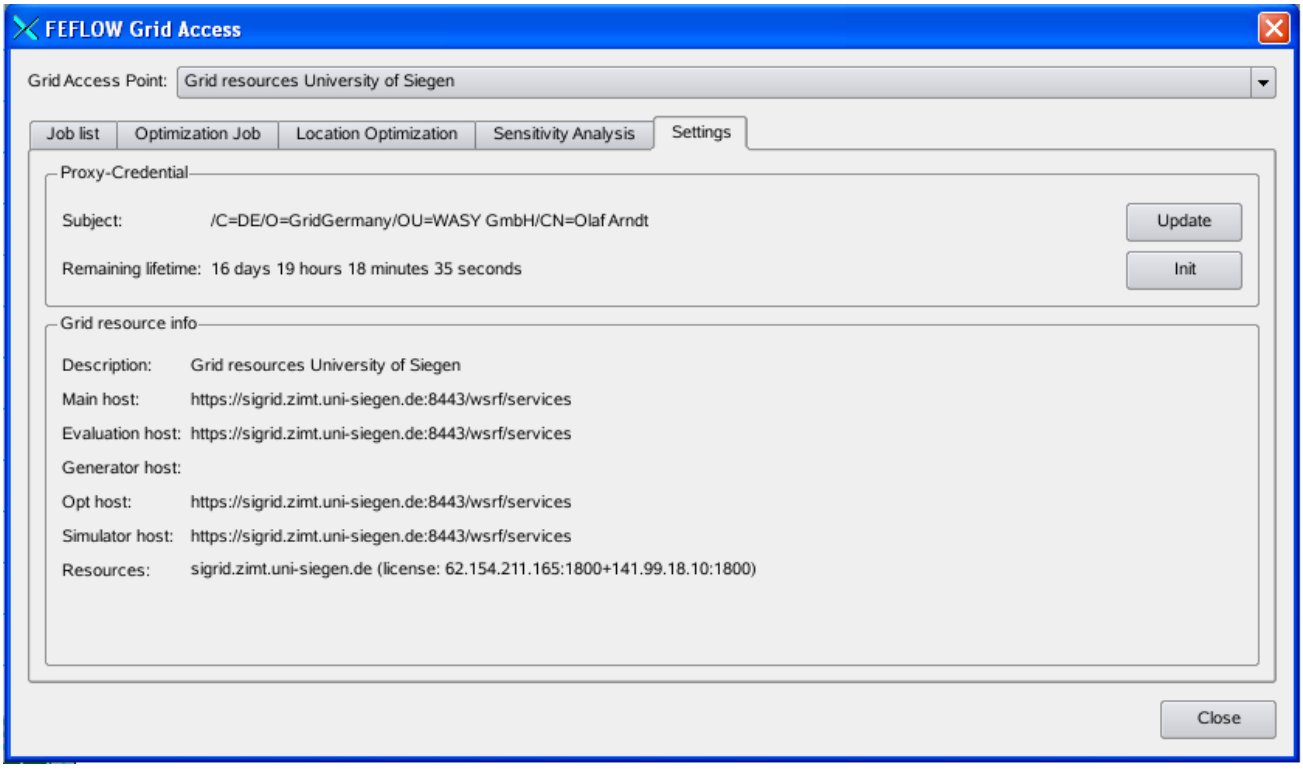

Figure 4: Grid settings dialog.

In the *Settings* tabular of the FEFLOW Grid client GUI all relevant information are listed, including the users credential and the potential used Grid service addresses.

4.2.2 Optimization Job Configuration

The Grid client GUI provides a tabular for configuring a general optimization job. The corresponding dialog is shown in Figure 5.

The dialog provides access to all relevant optimization settings as follows:

- Optimization parameter

  The optimization parameter configuration enables the configuration of the parameter which should be modified (= optimized) during optimization. In the optimization context this setting defines the decision variables of the optimization algorithm. In the corresponding parameter selection box all parameters are provided which are available on the current configuration. In addition to the

parameter itself, several options of the parameter must be defined which depend on the parameter. All available options are listed by pushing the "+" button including their corresponding description.

- Objective Function

  The objective function describes the measurement for the quality of a candidate solution. It is divided in two parts:

  1. the FEFLOW parameter, which should be used to calculate the objective function, and
  2. the method how to calculate the objective function.

  Both selection boxes are automatically filled with the available parameters and methods (similar to the optimization parameter selection) and provide their configurable options.

- Constraints

  The constraint section provides access to the optional parameter and method which must be fulfilled by a valid solution. Unlike the objective function, where only a single value can be returned, multiple constraints are possible, e.g. a certain value at several observation points. Currently, a solution is valid if each constraint value is less or equal zero.

- Optimization Settings

  Currently, in the optimization settings section are only the available algorithms and a maximum number of parallel degree is selectable. All further configuration parameters of the optimization (like number of decision variables or constraints) are taken from the configuration settings above.

- Finite Element Model

  Finally, the Finite Element model files (FEM) has to be selected, which is used for the optimization process. This FEM file may be the currently loaded model, but is not necessary to be.

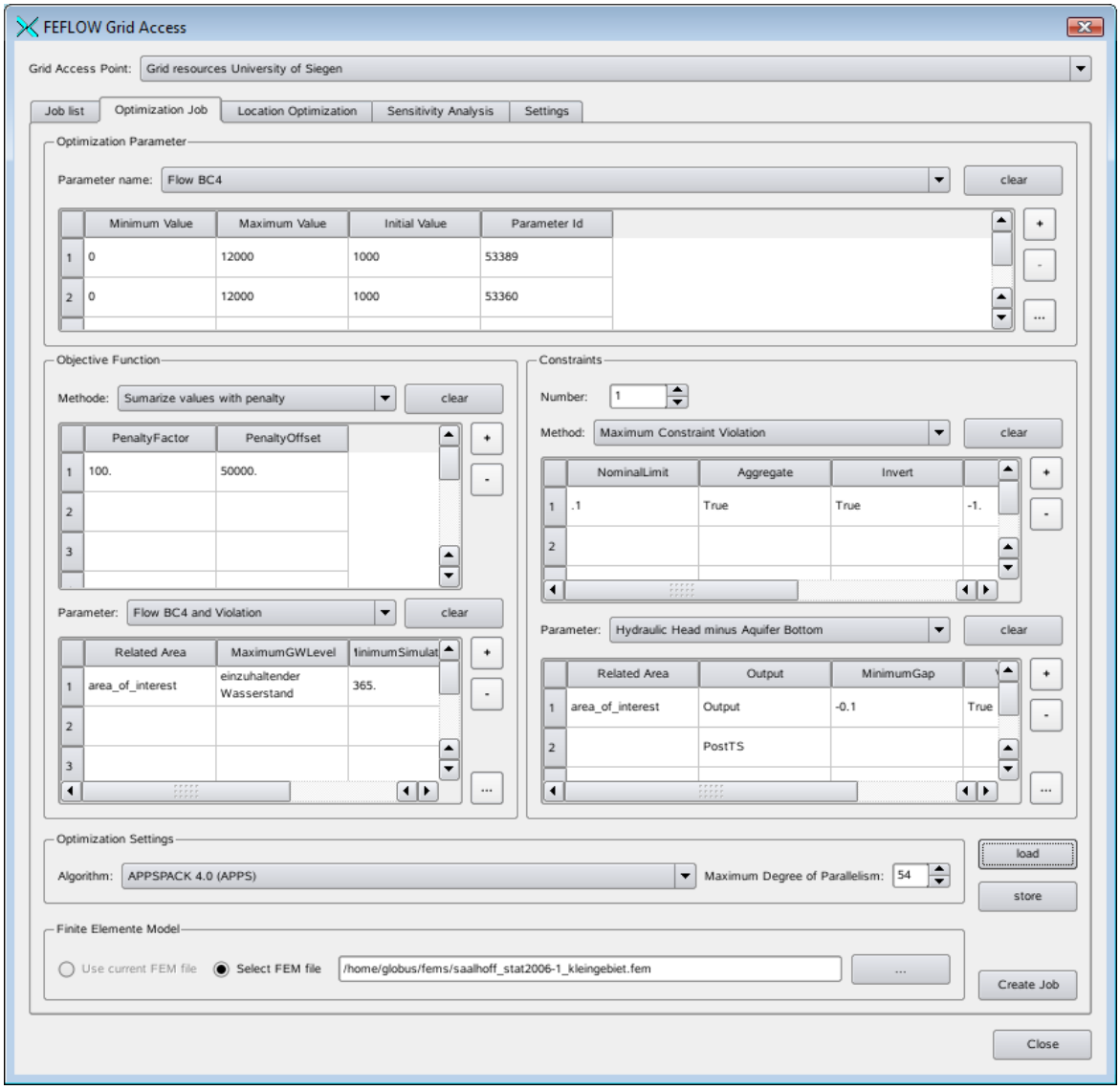

Figure 5: Grid optimization job definition dialog.

A detailed description of a sample configuration is given in section **Fehler! Verweisquelle konnte nicht gefunden werden.**. After completion of the job settings the job can be submitted to the Grid resources by clicking the "*Create Job*" button.

4.2.3 Monitoring jobs

To monitor jobs a job list tabular is provided by the client GUI. In the job list dialog (Figure 6) all jobs of the current user on the currently selected Grid resource are listed. Besides listing the jobs, the dialog provides the functionality of stopping, continuing, or deleting jobs. Furthermore, it provides access to details of each jobs.

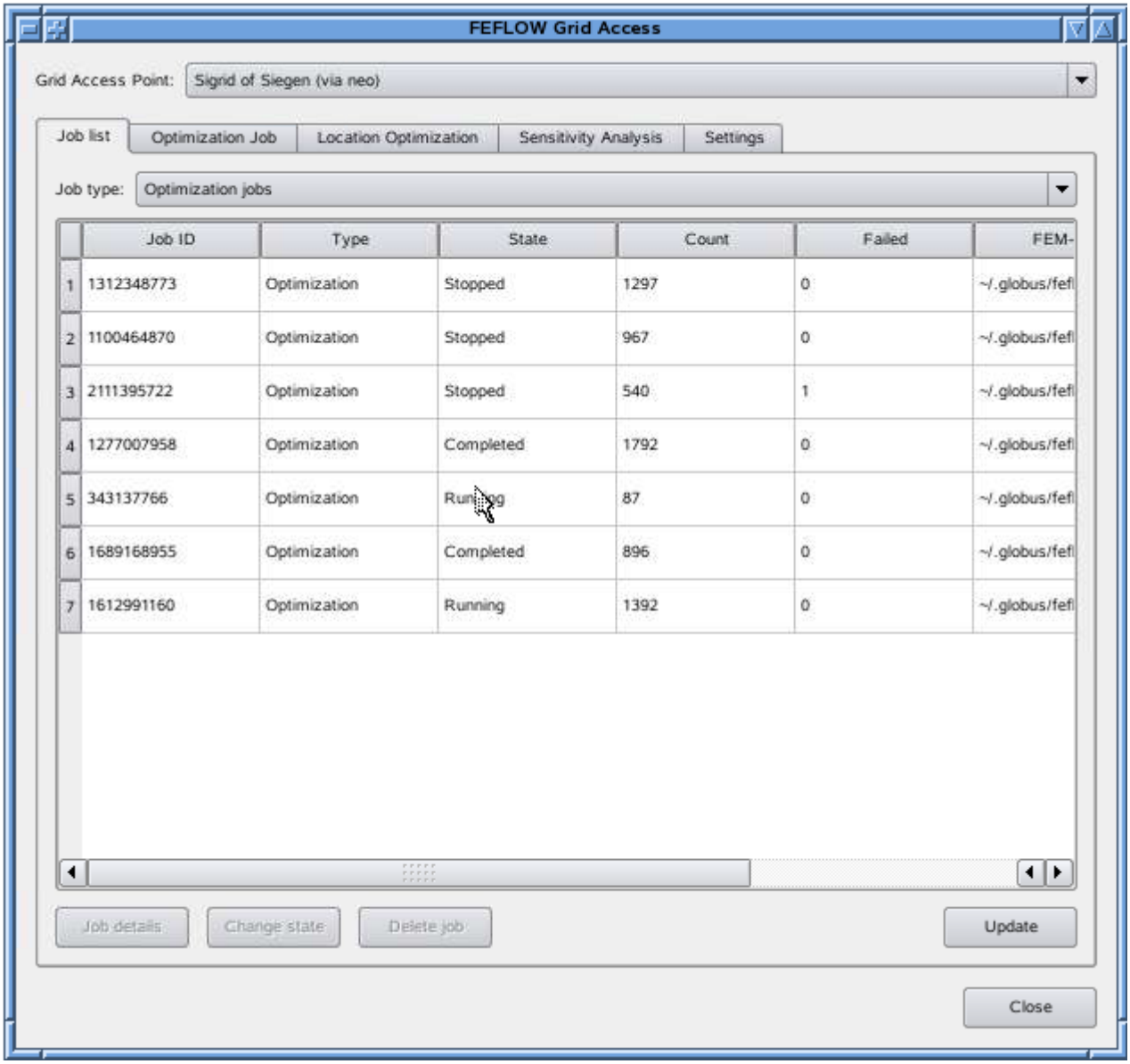

Figure 6: FEFLOW Grid client job list dialog.

Currently, the details of a job cover the decision variables, objective function and constraints. The corresponding dialog is shown in Figure 7. Using this dialog, the currently simulated jobs can be listed and a quick visualization of the optimization process is also provided. For further processing it is also possible to export the current, valid, invalid, or best results to an external file.

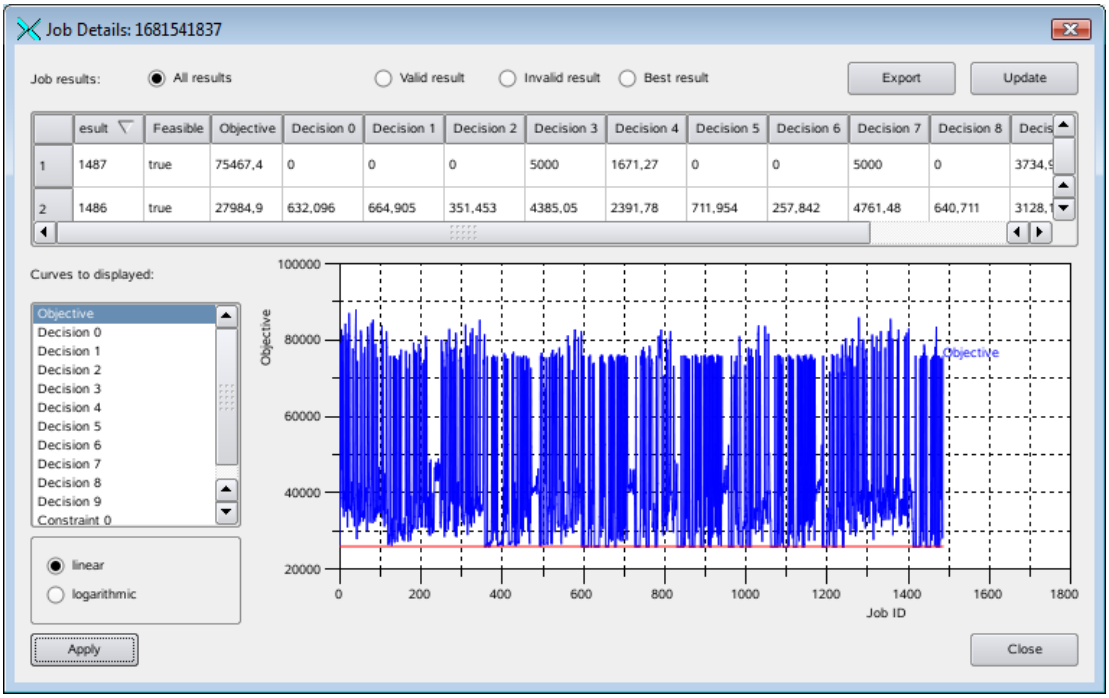

Figure 7: FEFLOW Grid client job detail dialog.

5 Examples

5.1 Groundwater model LINEG

The model used in the project is based on an existing groundwater model representing the quaternary aquifer between river Rhine and river Niers in the Lower Left Rhine Area from Krefeld to Xanten. It spreads over an area of more than 1000 km² and is represented by more than 100000 finite element nodes. DHI-WASY software FEFLOW® was used to build up the model and still is used for model maintenance and solution support in several projects.

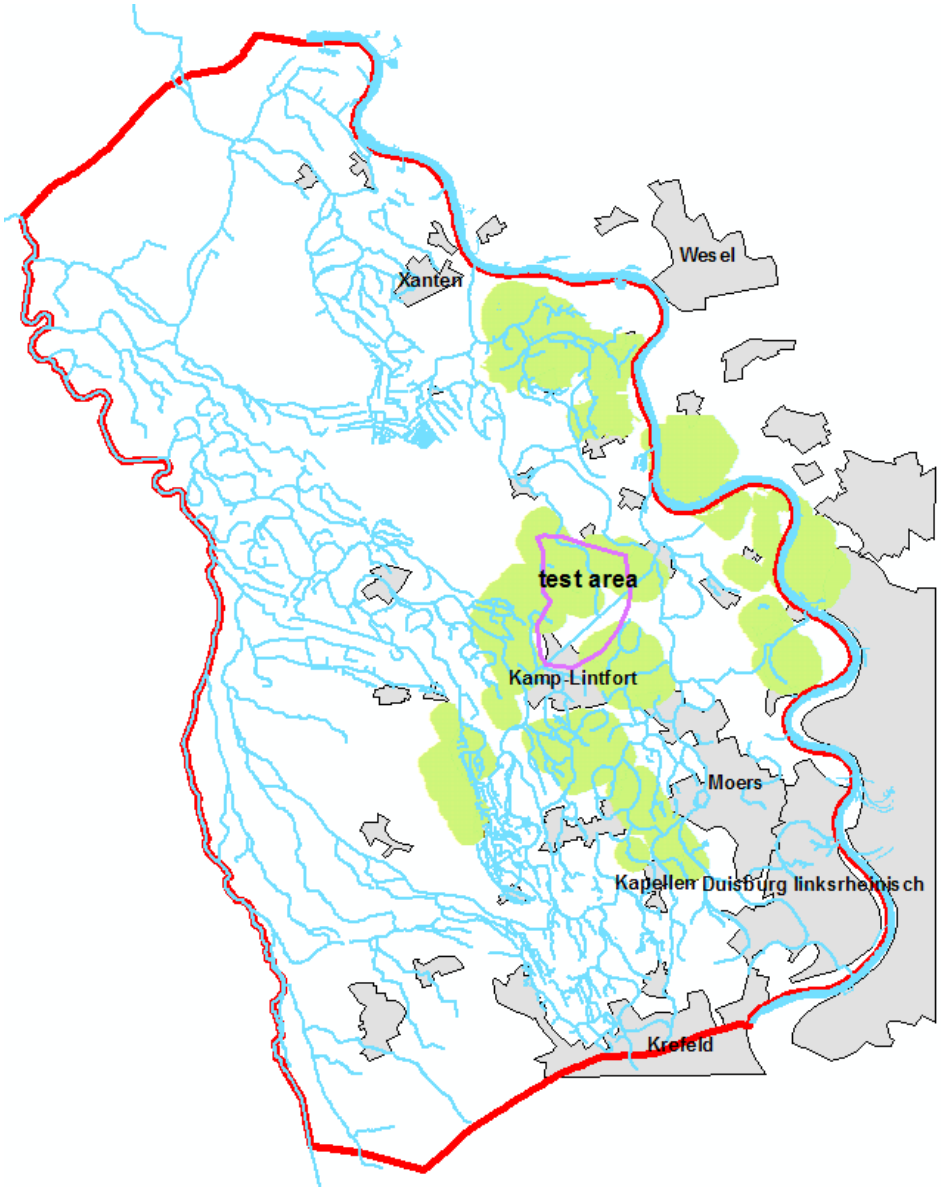

Figure 8: Map of the model border line (red line) showing the surface water net (blue lines), the area influenced by soil subsidence (green area), urban areas (grey

area) and the test area (purple border line) used in the project.

Wide parts of the aquifer described with model are influenced by soil subsidence which is caused by coal and salt mining activities. During decades of mining surface and subsurface dewatering structures were build up as need arises. More than 200 pumping stations and a wide mesh of surface water structures interacting with the aquifer are used for dewatering tasks. A heterogeneous land use enforces a detailed groundwater recharge consideration. Regarding all these aspects to describe the groundwater flow lead to a complex transient model, which was reduced to a quasi stationary version for the project use.

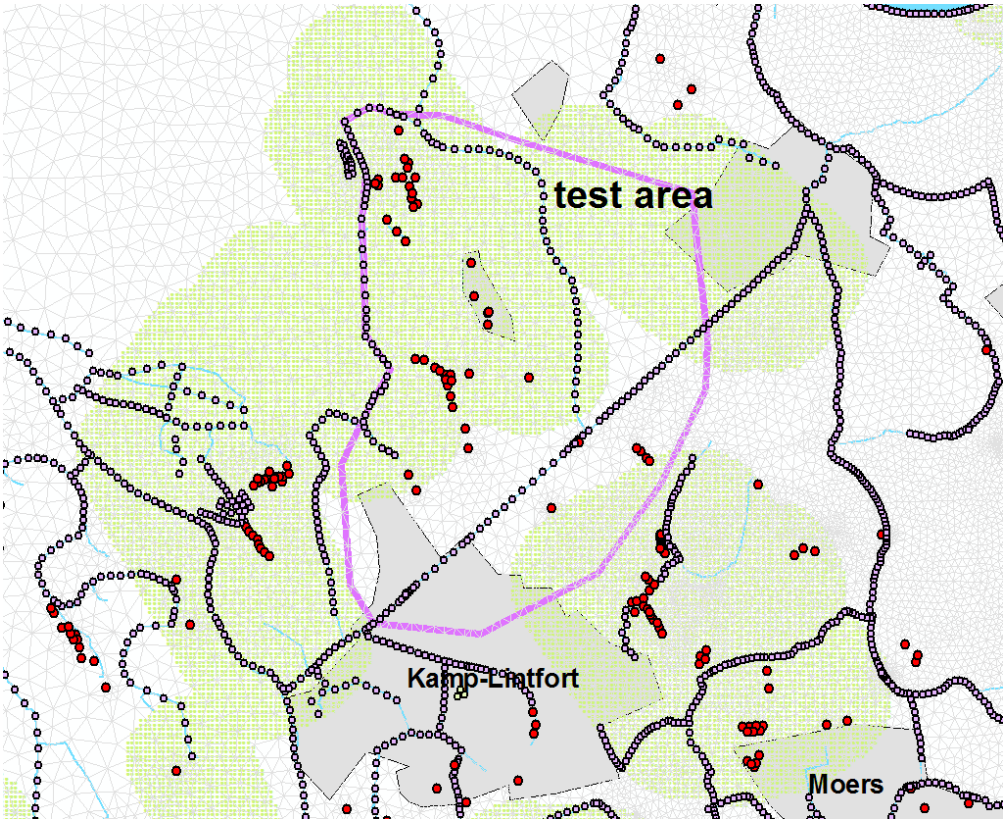

Figure 9: Detailed view on the test area including the finite element net and boundary nodes (red: well boundary condition; grey: river boundary condition). The wells located in the test area have to be substituted during the optimization process.

In near future coal mining activities in this area will be given up so stabile underground structure is going to allow a long term planning of dewatering system. A test area of 20 km² was chosen to proof the general usability of the selected optimizing algorithm in the field of location optimization. The test area is located in the central model region called Saalhoff. It is very heterogeneous in land use and hydrogeology which has to be respected during the optimization process.

5.1.1 Optimization Job Definition

Objective of the optimization task is the overall minimization of pumped water in the test area. Pumping stations delivery rates form the decision variables along with the real world coordinates of the locations. As constraint a minimum allowed depth to water table depending on land use was used. This constraint was mapped as a water table which was generated using a digital terrain model (DTM) and a land use map. It is stored in the FEFLOW ground water definition file as a reference distribution. The constraint must be satisfied in the whole test area but pumping station placement may be restricted. So areas with low conductivities and territories which cannot be used as pumping station location due to different reasons were defined as "no go zones" stored as a reference distribution as well.

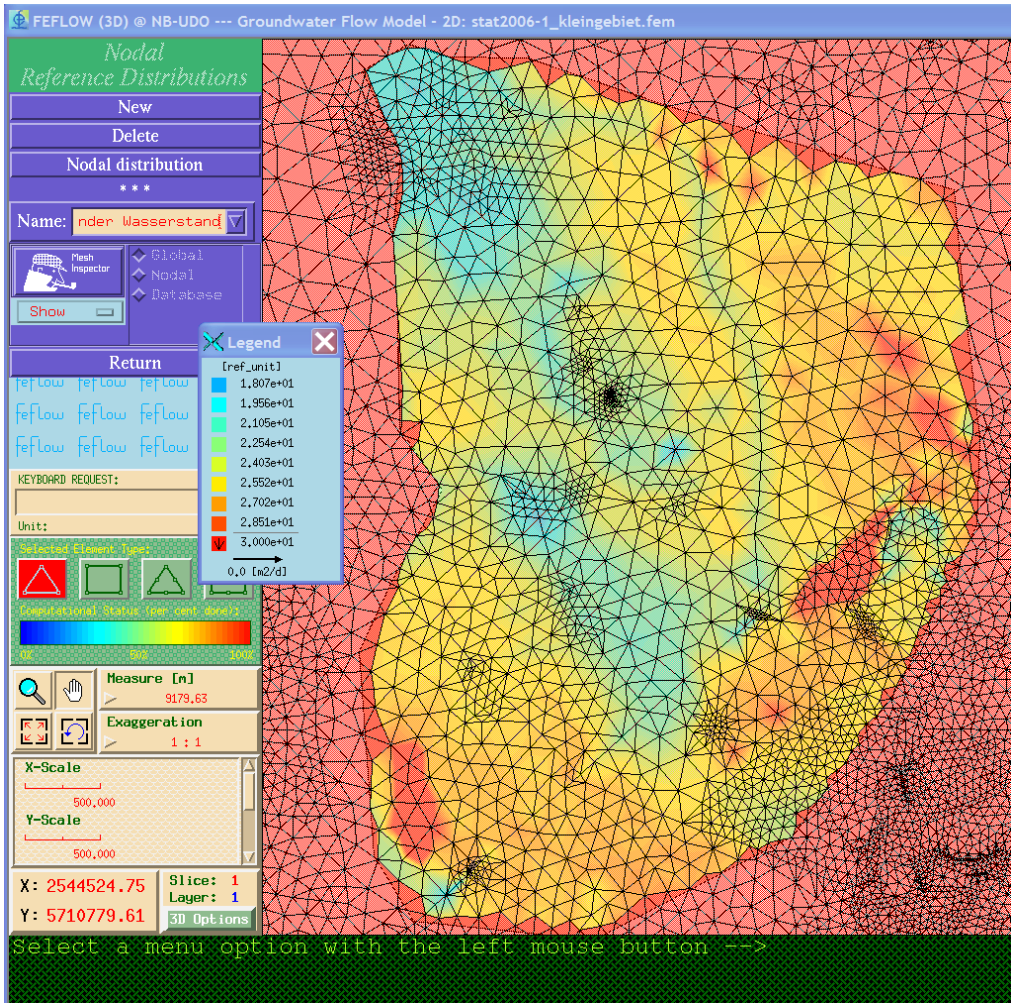

Figure 10: FEFLOW® GUI environment showing the constraint of minimum allowed depth to water table stored in a reference distribution.

Based on the job description above, a Grid optimization job was defined of using up to 30 wells. Each well was realized by three decision variable, real world x-coordinate, real world y-coordinate, and extraction rate. Unfortunately, the optimization was not able to handle this with the given resources and algorithms. This was caused by the large number of deci-

sion variables on one hand. On the other hand, the objective was defined as sum of the extraction rates and there was no feedback of the location with respect to the objective. Therefore, a modification of coordinates did not result in a modification of the objective value, which has caused a lot of different candidate solutions with the same quality with respect to the optimization algorithm. To avoid this, it was necessary to split the optimization task in two parts.

5.1.1.1 Location Optimization

In the first task, an optimization job was defined using constant extraction rates. The location (represented by real world x/y-coordinates) was used as decision variables, while the objective function was the summarized violation of the groundwater level or – if not violated – the summarized (negative) compliance of the groundwater level.

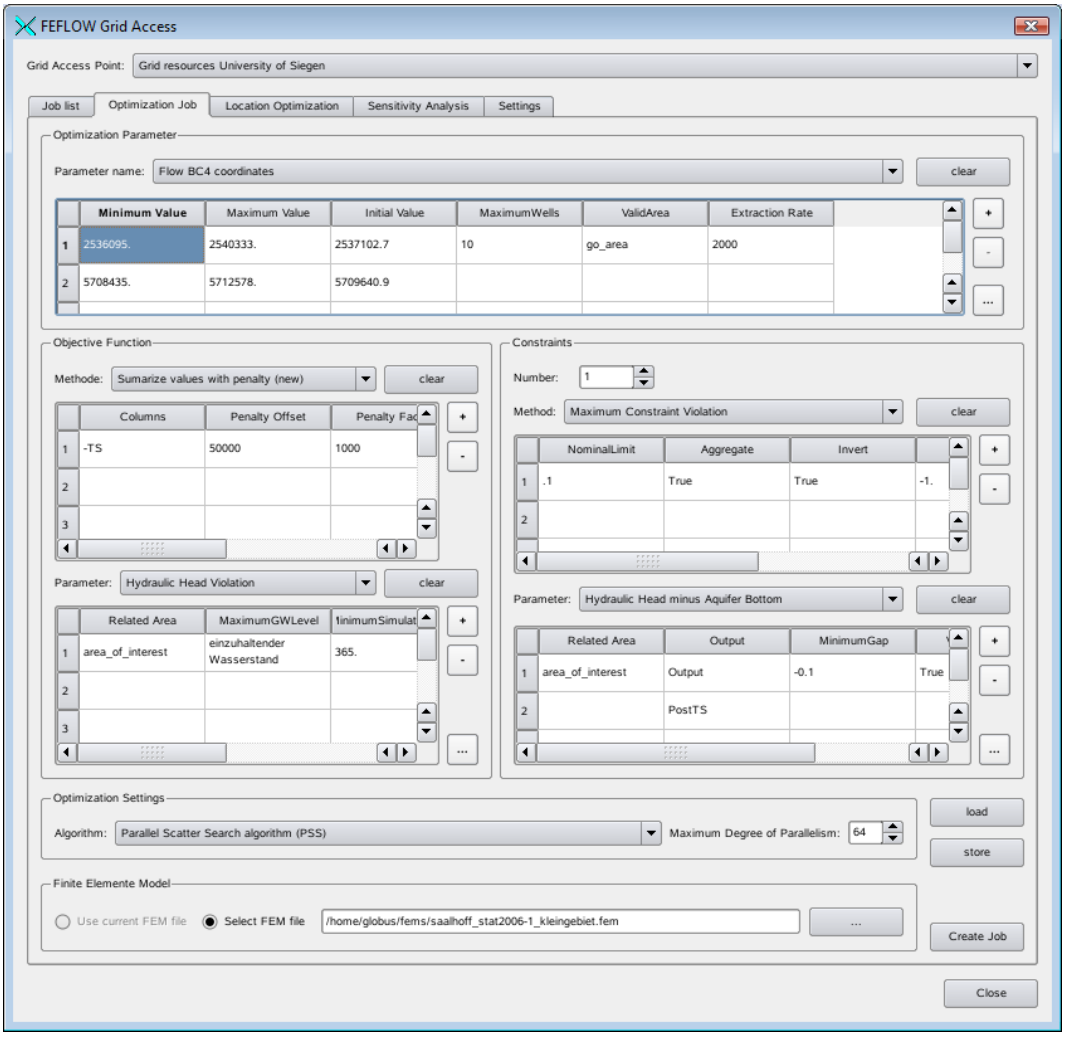

Figure 11: Saalhoff location optimization job setup.

In addition a single constraint was defined as maximum constraint violation of the hydraulic head minus the aquifer bottom. The constraint was necessary to avoid simulation in an undefined hydraulic behavior, thus

simulations were the groundwater level falls below the bottom of the aquifer. In this case, the simulation becomes unstable which leads to very small time steps and therefore to enormous runtimes per simulation.

The corresponding job setup is shown in Figure 11. The main advantage of that job definition is that there is a direct feedback of the solution quality (objective value) with respect to the decision variables (the well location).

5.1.1.2 Rate Optimization

Based on a temporary valid result of the location optimization, a second optimization job was defined using the "currently best" location result. Now, the decision variables were used to optimize the extraction rates at the obtained locations.

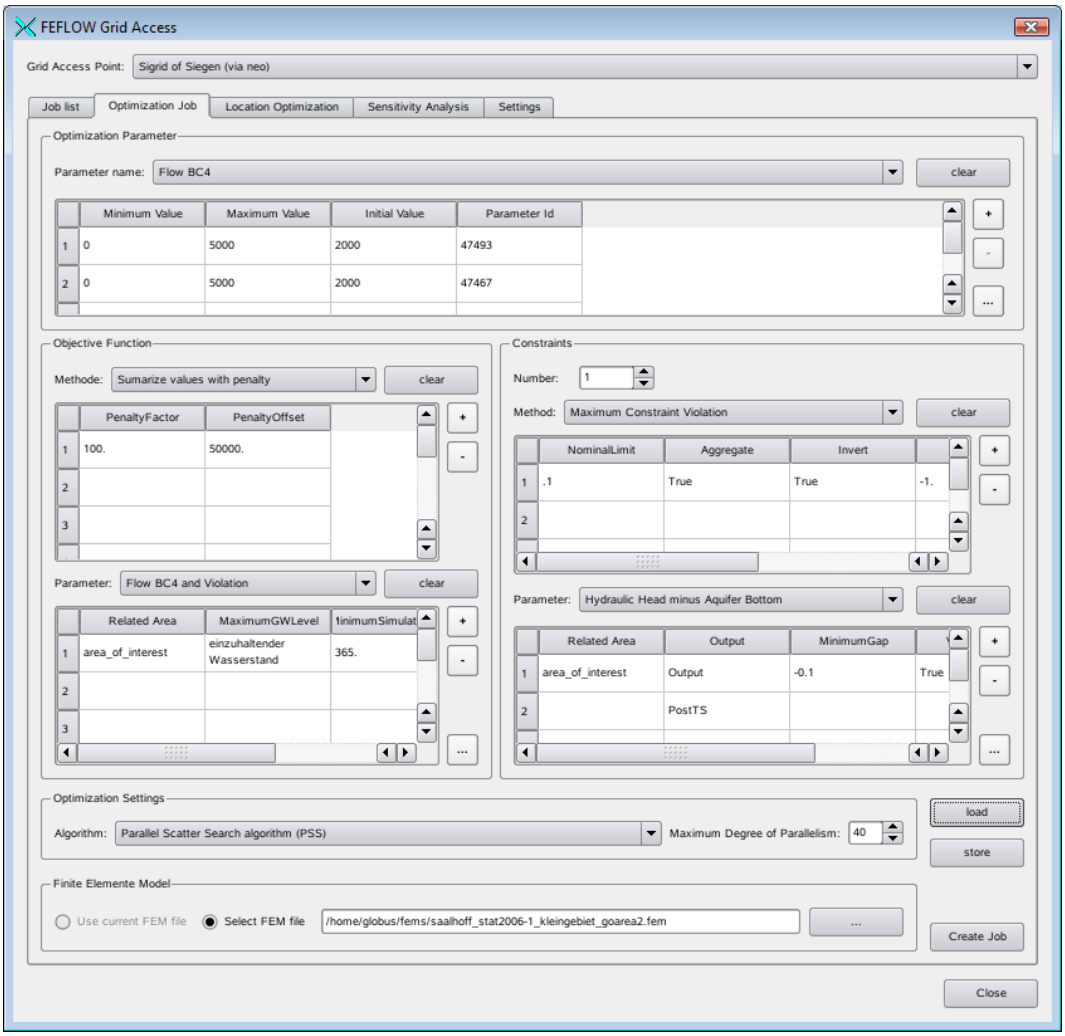

Figure 12: Grid job definition to optimize the extraction rates of the location optimization result.

The corresponding job setup is shown in Figure 12. For each well, a decision variable was defined representing the corresponding extraction rate at a given position. As objective function the sum of all extraction rates within the area of interest was used, including a penalty for invalid solu-

tions with respect to the optimization definition. As during the location optimization job, the constraint method was used to avoid unstable numerical behavior.

5.1.2 Optimization Results

Using those two optimization steps, it was possible to achieve results accomplish the optimization task. In the location optimization it was able to achieve valid solution using 10 wells á 2000 m^3/d.

During the second run the optimization was able to eliminate 5 of the wells, thus setting their extraction rate to zero by concurrently reducing the overall extraction rate by 5% with respect to the actual extraction strategy. In this case, 5% reduction is equivalent to ½ Mill. m^3 per year.

Because the optimization tasks were not completed it can be expected that it is possible to achieve better results and therefore a higher reduction of the extraction rates in the future. Nevertheless, realizing this optimization job it was possible to verify the system functionality and the approach of Grid-based optimization.

6 Glossary

Acronym/Term	Definition/Description
BC	Boundary condition
DLL	Dynamic link library
DSO	Dynamic shared object (shared library)
FE	Finite Element
FEFLOW®	2D/3D Finite Element Simulation System of the DHI-WASY GmbH
FEM	Finite Element Model
GRAM	Globus Resource Allocation and Management
GSI	Grid Security Infrastructure
GT4	Globus® Toolkit 4
GT2	Globus® Toolkit 2
GUI	Graphical user interface
IFM	FEFLOW Interface Manager
ITU-T	International Telecommunication Union Standardization Sector
JRE	JAVA runtime environment
LM	License Manager
NetLM	DHI-WASY GmbH Network License Manager
Optix®	Suite of optimization algorithms by the University of Siegen.
PKI	Public Key Infrastructure
PMI	Privilege Management Infrastructure
SDK	Software development kit
SOA	Service Oriented Architecture
WS-GRAM	Web Service Globus Resource Allocation and Management
WWW	World Wide Web Consortium (W3C)
XML	Extensible Markup Language
Xvfb	X virtual framebuffer
X.509	ITU-T standard for PKI and PMI

7 References

[1] Barth, T., Freisleben, B., Grauer, M., and Thilo, F.: "Distributed Solution of Simulation-Based Optimization Problems on Networks of Workstations", *Journal on Computer Science and Systems*, pp. 94-105, 2000.

[2] H.-J. Diersch: "FEFLOW 5.3 - Finite Element Subsurface Flow & Transportation System", User's Manual, WASY GmbH, Berlin, 2007

[3] globus.org: The Globus Alliance: http://www.globus.org (visited 2008-01-27)

[4] Hough, P., Kolda, T. G., and Torczon, V., "Asynchronous Parallel Pattern Search for Nonlinear Optimization", *SIAM Journal of Scientific Computing* **23(1)**, 134-156 (2001).

[5] Foster, I., and Kesselman, C. (Eds.): The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004.

[6] Singh, M., P.; Huhns, M., N.: Service-Oriented Computing Semantics, Processes, Agents, Wiley 2005.