This readme file is under construction! Please report problems regarding the installation and operation of BRNS to florian.centler@ufz.de!

Installation

Prerequesites

To get BRNS up and running you need:

1. Fortran compiler
2. Maple, including SHARE package
3. this BRNS package

In the following, we assume that you have a Windows system with English as the system language and you use

- the Intel Fortran Compiler embedded in Microsoft Visual Studio 8 as available in the GReaT MoDE group (**`w_fc_p_10.1.025.exe`**, with licence file **`COM_W_CMP_FOR_L8FS-R5WD2RB7.lic`**)
- Maple 6 as available in the GReaT MoDE group or Maple 10 or 13 (available at the UFZ as a network licence)

Software Installation

1. Install the Intel Fortran Compiler. You will be asked for the licence file during the installation.
2. Install Maple.
    a. For Maple 6, install the single user version and copy the licence file **`licence.dat`** into the **`Licence`** directory within the Maple directory.
    b. For Maple 10 or 13, follow the installation instructions provided by the UFZ. The licence server is **`linzenz.intern.ufz.de`**.
3. Install SHARE library
    a. If using Maple 6, you might have to install **`share.tar.Z`** from the **`BRNSPackage`** directory: place the extracted **`share`** directory within the Maple 6 directory.
    b. For Maple 10 and 13, install **`sharelib_maple_above_6.zip`** from the **`BRNSPackage`** directory (downloaded from http://www.maplesoft.com/applications/app_center_view.aspx?AID=1907 , resp. http://myweb.tiscali.co.uk/maplenut/) Follow the instructions in **`install.txt`**.
    c. Test whether the library is working: Start Maple and type **`with(share);`** you should get **`See ?share and ?share,contents for information about the share library`**
    d. If an error message appears, try to create the **`share`** folder as a regular user and not as superuser / administrator.
4. Extract the **`BRNSPackage`** to **`C:\Program Files`**. This is recommended even if you are using a german Windows system.

Configuration

If you have used other than the recommended directory locations, you will have to update the paths accordingly in these files: **Generic-Ex.mws**, **proc0903-Mp.mws**. Search for "C:\" and replace all occurencies according to your installation.

Configuration for Drag&Drop Model Creation

If you intend to use the Drag&Drop model creation feature (recommended), make sure that all paths are correctly set in the beginning of **makeBrns.bat**. You will in any case have to update the paths pointing to Visual Studio and Maple, if you are on a german Windows system (**Programme** vs. **Program Files**). In **makeBrns.bat** you also choose which Maple version is to be used. A version with a local licence is preferable. Also make sure that the paths are correctly set in the Maple Input File versions of the **mws** files: **Generic-Ex.mpl** and **proc0903-Mp.mpl**. If your **mws** files are already ok, simply open them in Maple 10 or 13 and select "Export as … Maple Input" to create these files. Finally, create a link of **makeBrns.bat** at a convenient place, for example the Desktop.

Running a Simulation

Running a simulation consists of several steps, which can be done manually, or automatically by a Drag&Drop operation.

Manual Model Creation

1. Copy **Generic-Ex.mws** to a new file, edit it in Maple according to your simulation set up and execute it. This will create a **spread.m** file.
2. Open **proc0903-Mp.mws** in Maple and execute it. This will read the **spread.m** file and create the Fortran source code.
3. Copy the generated Fortran files from **GeneratedFortranFiles** to **FortranFiles**. (This step can be avoided if setting the paths accordingly in **proc0903-Mp.mws**)
4. Open **all.sln** in th **FortranFiles** directory. Make sure, that **Release** is selected (and not **Debug**) just below the menu bar, right-click on **all** and select **Rebuild**.
5. Move **all.exe** from the **FortranFiles\Release** directoy to a convenient directory and run it.

For compiling the DLL version of BRNS, use the ***DLL.mws** files ,and move **parameters.f** from the **FortranFiles** directory to the directory **FortranFiles\BrnsDll** and right-click on **BrnsDll** for compilation. (If parameters are not passed from the tranport code, the movement of **parameters.f** needs only be done once.)

Drag&Drop Model Creation

Instead of editing the **mws** file in Maple, edit the Maple Input File version (**mpl** file) with any text editor. You can also create this file by selecting **Export as … Maple Input** from Maple 10 and 13. Drag and drop the **mpl** file on makeBrns.bat (or a shortcut to it). This will automatically go through all steps and finally move the executable **all.exe** to the directory of the **mpl** file. In the Command shell, you can also type: "**makeBrns.bat myScenario.mpl**" and "**makeBrns.bat myScenrario.mpl dll**" to create the DLL version of BRNS.

Compilation on Unix

make -f makefile.gnu [clean|brns|brns.so]

Choose **brns** for the stand alone version and **brns.so** for the shared object version for coupling to external transport codes. When compiling the dll / so version on a new architecture / cluster, the blas and lapack libraries likely have to be recompiled on the system. When doing so, modify the master makefile to include the "-fpic" switch which is required for generating a shared object file.

Known Issues

1. When using the makeBrns.bat script with Maple 10, sometimes the second Maple invocation has problems reading 'spread.m' ("invalid back reference"). No idea what that means and how it can be solved. Stick with Maple 6 for now if possible, when using the script.
2. When using the makeBrns.bat script, sometimes the compilation fails because a Fortran code file starts with stupid characters ("bytes used …"). If this happens, just run the script again. If the problem persists, send me an email (florian.centler@ufz.de), the script can be updated to avoid this problems. (It just happened for one source code file on my machine, and this is already handled by the script)
3. Compilation/Linking: If symbol DGESV_ is not found, the call to this function could be outcommented. Double check however, whether this function is really never called in your simulation. If the write statements in invokebrns.f lead to errors, they can be outcommented.

Coupling the Chemical Solver of BRNS to External Transport Codes

Coupling Function

Use the ***DLL.mws** / ***DLL.mpl** files for describing your system when coupling to an external transport code. These files only contain the information relevant to the chemical system.

The DLL / SO file provides the function "invokebrns()". This function basically expects to get a concentration vector for one grid element, and computes the new concentration vector at t+timeStep according to the maple-defined chemical reaction network. The size of the time step must be provided in timeStep. Invokebrns() expects 13 arguments as listed below. All arguments are passed by reference.

Argument	Type	Description
1	array(real*8),NoC	concAfterTransport
2	array(real*8),NoC	concBeforeTransport(used as init. guess)
3	array(real*8),NoC	outputConcentrations(new conc are here)
4	integer	numberOfSpecies (NoC)
5	real*8	timeStep
6	array(integer),NoC	fixedConcentrationBoundary
7	integer	returnValue
8	real*8	pos_x
9	real*8	pos_y
10	real*8	pos_z
11	real*8	porosity
12	real*8	waterSaturation
13	array(real*8),NoP	parameterVector

NoC: Number of chemical species, NoP: Number of parameters; parameters can be defined in the Maple input (just as switches). But while switches only can take the values zero and one, a paramter can be assigned any value each time invokebrns() is called.
FixedConcentrationBoundary
0: No fixed (boundary) concentration for this species
1: keep concentration fixed
Return value:
0: everything ok
1: negative concentration occured
2: exceeding newton iteration
3: exceeding newton iteration, negative concentration occured

Rate Output

The last time the reactive solver is called, it can write the file "ratesAtFinish.dat" with all rates information for the whole domain. An existing file will be overwritten. The format is: "x y z rate1 rate2 rate3 ...". These are exactly the rates as defined in the maple input file. The writing of the file is triggered by setting "returnValue" to a negative value. File deletion occurs, when a value of "-1" is passed (should be used when chemical solver is called for the first time in the last iteration).

Files

Generic-Ex.mws - Maple Worksheet defining the model
Generic-ExDLL.mws - Maple Worksheet defining the chemical model, ready for coupling to an external transport code
proc0903-M.mws - Maple Worksheet that creates the Fortran Sourcecode
*.mpl - The same in Maple Input format for drag&drop
GeneratedFortranFiles - These are the files created by proc0903-M.mws
FortranFiles - Sourcecode of the simulator, load all.dsw in Visual Fortran, all.sln in MS Visual Studio
acglib-Maple6 - Maple Library needed by proc0903-M.mws, performing the code generation
for Maple 6
acglib-Maple10 - the same for Maple versions > 6
share - share library (including Macrofort)
makeBrns.bat - script for batch creating BRNS