Using the CVODE/SUNDIALS C+-library in GroIMP.

# Prerequisites:

GroImp (64-bit)

Visual studio 2015: Confirm that you have access to “VS 2015 x64 Native Command Prompt”

CMake 3.4.1 (32-bit) or 3.4.3 (32-bit)

Sundials 2.6.2

Administrator rights on your computer

# Step 1. Creating the shared CVODE library files

1. Create a folder “SUNDIALS” under My Documents and create three subfolders called “srcdir”, “builddir” and “instdir”.
2. Extract the Sundials.tar and copy all files to “…/SUNDIALS/srcdir”.
3. Set the local path variable for CMake
   1. Find where CMake was installed (normally under C:/Program Files(x86)/CMake)
   2. Right click “My computer” and select “Properties”
   3. Click on “Advanced system settings” (**requires administrator rights**)
   4. Under the advanced tab click on “Environment variables…”
   5. Under **User variables** find the PATH variable, click edit and add “;C:\Program Files(x86)\CMake\bin”. If no PATH variable is present, click “New…” and manually add the variable. **DO NOT CHANGE ANYTHING UNDER SYSTEM VARIABLES.**
   6. Click “Ok”
4. Run CMake
5. Browse for srcdir as source code and builddir as build.
6. Click “Configure” and select “Visual Studio 14 2015 Win64”.
7. This will generate a number of adjustable variables:
   1. Deselect: BUILD\_IDA, BUILD\_IDAS, BUILD\_KINSOL and BUILD\_STATIC\_LIBS
   2. Select: BUILD\_SHARED\_LIBS
   3. Change: CMAKE\_INSTALL\_PREFIX to “…/SUNDIALS/instdir”.
8. Click “Generate”
9. Run “VS 2015 x64 Native Command Prompt” **as administrator.** Run following commands
   1. cd c:\Users\...\Documents\SUNDIALS\builddir
   2. msbuild ALL\_BUILD.vcxproj
   3. msbuild INSTALL.vcxproj

# STEP 2: accessing the library from GroIMP

1. Open GroIMP
2. Add the cvode.dll to the path and load to the system by running following script (change the paths accordingly):

private void loadCVODElib(){

String libpath = System.getProperty("java.library.path");

if(libpath.contains("cvode")){

println("Shared CVODE library already in path.");

} else {

try {

libpath = libpath + ";c:\\Users\\user\\Documents\\SUNDIALS\\instdir\\lib\\sundials\_cvode.dll";

System.setProperty("java.library.path", libpath);

println("Shared CVODE library added to path.");

} catch (Throwable e){

println("Error adding CVODE library to java path./n" +e);

}

}

try {

String cvodepath = "c:\\Users\\user \\Documents\\SUNDIALS\\instdir\\lib\\sundials\_cvode.dll";

System.load(cvodepath);

println("Shared CVODE library loaded.");

//sundials\_cvode cvode = (sundials\_cvode) Native.LoadLibrary(cvodepath, sundials\_cvode.class);

} catch (Error e) {

String error = (String)e;

if(error.contains("already loaded")){

println("Shared CVODE library already loaded.");

} else {

println("Native code library failed to load.\n" + e);

}

}

String libpath2 = System.getProperty("java.library.path");

if(libpath.contains("nvecserial")){

println("Shared nvecserial library already in path.");

} else {

try {

libpath2 = libpath2 + ";c:\\Users\\user \\Documents\\SUNDIALS\\instdir\\lib\\sundials\_nvecserial.dll";

System.setProperty("java.library.path", libpath2);

println("Shared nvecserial library added to path.");

} catch (Throwable e){

println("Error adding nvecserial library to java path./n" +e);

}

}

try {

System.load("c:\\Users\\user\\Documents\\SUNDIALS\\instdir\\lib\\sundials\_nvecserial.dll");

println("Shared nvecserial library loaded.");

} catch (Error e) {

String error = (String)e;

if(error.contains("already loaded")){

println("Shared nvecserial library already loaded.");

} else {

println("Native nvecserial library failed to load.\n" + e);

}

}

}

protected void init()

[

{

loadCVODElib();

}

]

1. Make sure this function is called before you call the solver or an error will be received.
2. At the top of your script, add:
   1. import de.grogra.numeric.\*;
   2. import de.grogra.numeric.cvode.\*;
3. The solver can then be changed by running setSolver(new CVodeAdapter());
4. You can manually set the tolerances with:
   1. CVODE.setAbsTolDefault(1e-12);
   2. CVODE.setRelTolDefault(1e-12);
5. Within your initialisation script, it should look like this:

