Coupling Procedure

The following document is intended to walk step by step through the process of using the STAR-CCM+ and SERPENT 2 coupling code CONSTELATION.

The document is split into several parts. The first part will detail the codes needed. The second part covers the set-up needed and is split into three subsections; these subsections will cover how to set up each side of the coupling for the code to be executed properly. The third part shows the overall coupling algorithm. The fourth part will dictate how CONSTELATION can be modified to fit different needs. The fifth part will provide a step-by-step guide in how to execute the CONSTELATION code.

1. Codes Required

CONSTELATION is a code written in Python, specifically Python 2.7. Even though Python 3.7 is the most up to date version, there are certain subroutines used by SERPENT 2 that are currently only supported in Python 2.7. Therefore, in order to execute CONSTELATION an installation of Python 2.7 is needed.

An installation of STAR-CCM+ is needed. A specific version is not necessary though the most up to date version should be used. A STAR-CCM+ simulation file is also needed. The only necessary modification to an already working STAR-CCM+ simulation file is the making of a javaMACRO, the inclusion of a derived part, and adding an update table function. This is discussed in more detail in section II.C.

An installation of SERPENT 2 is needed. SERPENT 2 is currently in beta and is constantly being updated. As such, the newest version of SERPENT 2 should be used. A SERPENT 2 base model file is needed.

1. Simulation Setup
   1. Overall File Setup

As CONSTELATION is an external coupling code, it uses a series of external files to communicate with the codes and operating environment it is running in. CONSTELATION is designed to be operated in a Linux Server Cluster. It can be conceivably executed in any operating environment with a few minor tweaks but the computational cost of SERPENT 2 makes a server cluster the best choice. As such CONSTELATION uses a series of submission scripts to execute each of the codes.

Submission scripts within a Linux Server Cluster are essentially small files that submit jobs to the server queue. In these files the basics of the overall simulation are laid out. Those being, the amount of time-requested for the script to be executed, the number of CPUs and Server Nodes requested to be executed on, the software which to execute (i.e STAR-CCM+), and the file in which to execute.

The number of submission scripts needed to execute CONSTELATION is primarily based on the number of STAR-CCM+ simulations being used. This will be expanded upon in Section C. But in general, there should be one STAR-CCM+ simulation for every material interface that is being manipulated within SERPENT 2.

In the simplest case, there is one STAR-CCM+ simulation and one SERPENT 2 simulation. In this case, there are three submission scripts that need to be created. The first submission script is called the “Master” or “Main” submission script. It is inside this script that the overall wrapping code CONSTELATION is submitted to the server queue.

The second submission script is the STAR-CCM+ submission script. Inside this script will be the name of the STAR-CCM+ simulation as well as the associated javaMACRO. These will be discussed in detail in Section B. When considering the HENRI STAR-CCM+ simulation a single server node with 48 cores is requested per simulation. This number of nodes and cores requested can be highly variable depending on the complexity of the STAR-CCM+ simulation.

The third submission script is the SERPENT2 submission script. Inside this script will be the name of the SERPENT 2 base model. When considering the TREAT SERPENT 2 base model 80 server nodes with 48 cores each is requested. As with the STAR-CCM+ simulation the number of nodes and cores requested is highly dependent on the SERPENT 2 simulation and its complexity.

An example of the submission scripts used when coupling the HENRI STAR model to the TREAT SERPENT 2 model are included in the appendix.

When executing CONSTELATION a certain file-setup is expected. This can be modified within the script but as it is. There should exist one overall directory. This is where the STAR-CCM+, SERPENT 2, submission scripts, and CONSTELATION should be placed. Then there a number of sub-directories. The main purpose of these subdirectories is to serve as an archive for files as the code is executed. There should be one overall sub-directory named Archive, it is in here where CONSTELATION will save some key-files every time step so that the user can look at them and post-process if desired. A number of other sub-directories should also be created, these are for STAR-CCM+ to place the csv files it creates while it is executing. An example of the directory layout when only executing one STAR-CCM+ sim is shown here, other files, such as the SERPENT 2 source files and cross section files are omitted but should be placed in this folder too:



* 1. SERPENT 2 File Setup

In order to use CONSTELATION, there are multiple steps that need to be performed regarding SERPENT 2. A general overview of coupling with SERPENT 2 can be found here: <http://serpent.vtt.fi/mediawiki/index.php/Coupled_multi-physics_calculations>. The following list outlines the basic steps:

1. Creation of SERPENT 2 base model
2. Execution of SEREPNT 2 base model in Source mode
3. Creation of detector/outputfile inputs
4. Transient Inputs
5. Decide on type of universal multi-physics interface

The creation of a SERPENT 2 base model can be an extremely involved process and falls outside the scope of this README. However, once a SERPENT 2 base model has been created it is important to remember that the system must be as close to critical as possible. This is a requirement of the code when executing in Source mode as a sub-critical or super-critical system will throw off the way that SERPENT 2 stores neutron information.

Once, a SERPENT 2 base model has been completed the next step is to execute the code in source mode. NOTE: The files created can be quite large. On the order of 50~100 GB, depending on how many neutrons are being saved. When using CONSTELATION for HENRI a source file of 55 GBs was created, this corresponds to around 763 million neutrons.

This is necessary for executing SERPENT 2 in a transient simulation. Once these source files have been created SERPENT 2 can then be executed in transient mode. There are essentially two transient modes a coupled version and a non-coupled version. As this is a coupling code the coupled transient mode is used. A more detailed overview of the steps needed to create a SERPENT 2 transient setup can be found here <http://serpent.vtt.fi/mediawiki/index.php/Transient_simulations>.

Arguably the most important part of the SERPENT 2 side of this coupling is the detector outputs. This is because the detector inputs found within SERPENT 2 are the main way of extracting information from SERPENT 2. At least for coupling involving materials that do no fission, as is the case with helium-3 within HENRI. If the coupling is using a material that normally fissions, like uranium fuel, SERPENT 2 will automatically output normalized and relaxed power data. CONSTELATION has not currently been tested with this output data but should be able to be easily modified to accommodate it.

However, for coupling involving materials that do not fission, a detector is needed. How to define a detector within SERPENT 2 can be found here <http://serpent.vtt.fi/mediawiki/index.php/Input_syntax_manual>. The CONSTELATION code will then read the file output by SERPENT 2 for the name of the detector and automatically extract the data necessary. It will take this extracted data and then output it in the form a csv file that can be read by STAR-CCM+.

As discussed in the SERPENT 2 wiki on coupling. SERPENT 2 uses a file called a multi-physics interface that can be used to import density and temperature distributions for any material that is defined in the SERPENT 2 base model. The type of multi-physics interface used will change on a case by case basis, but for the HENRI TREAT coupling a regular based mesh interface is used. This essentially means that the density and temperature distributions are read in on a cartesian mesh created from data extracted from STAR-CCM+. It is this file that SERPENT 2 uses to update density and temperature of a material during a transient simulation. CONSTELATION will automatically take a csv created by STAR-CCM+ and turn that csv into a Type 2 multi-physics interface file.

Once SERPENT 2 has completed a single time-step it will create a detector output, this output will be converted into a csv file. Once this csv file has been created CONSTELATION will check to see if the STAR-CCM+ time-step has been completed. It does this by checking for the existence of a file created at the end of the STAR-CCM+ timestep. It will then create its own file saying that SERPENT 2 is done executing.

* 1. STAR-CCM+ Setup

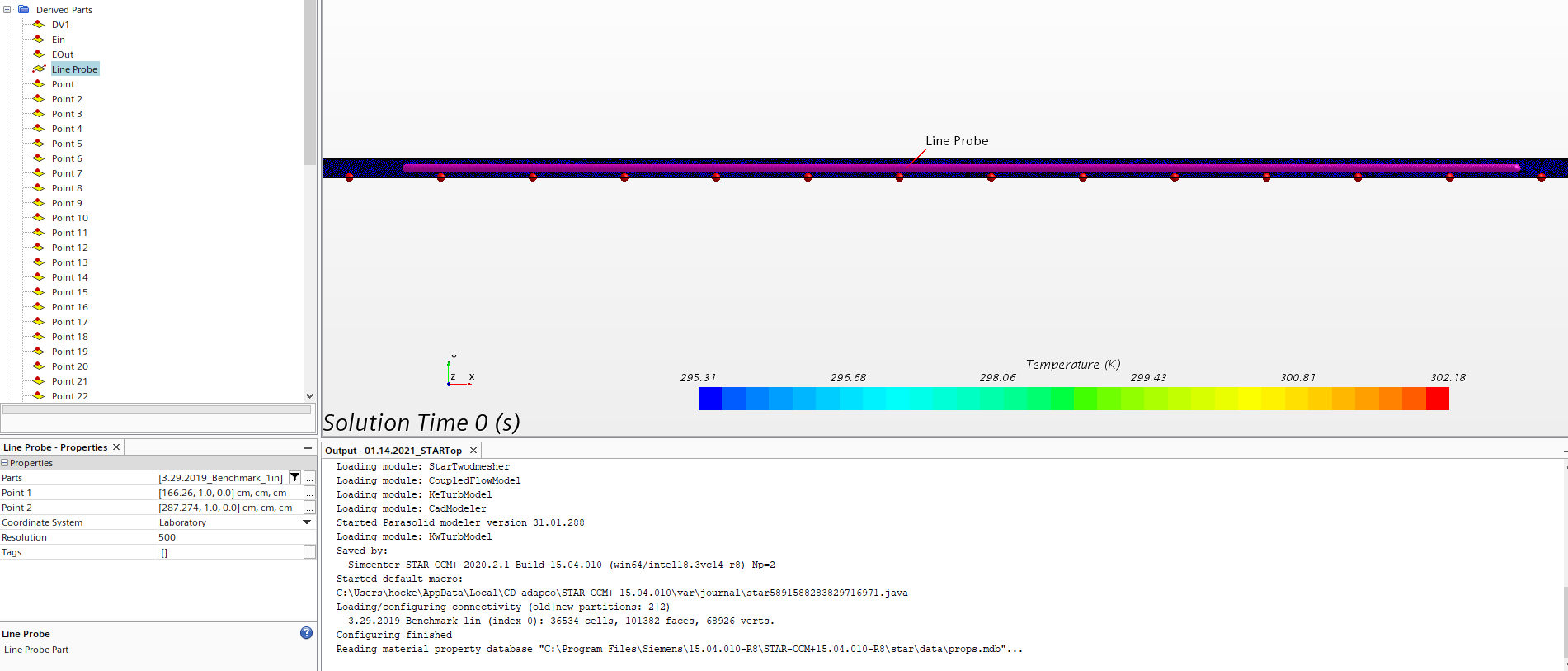
In order for CONSTELATION to communicate with STAR-CCM+ a javaMACRO is needed. A javaMACRO is a script code written in JavaScript that can automatically execute and control a STAR-CCM+ simulation. STAR-CCM+ has a built-in recording feature that will automatically change actions performed in the STAR-CCM+ GUI into JavaScript code. This code can then either be executed within the GUI to automatically perform the actions recorded, or it can be executed outside the GUI using STAR-CCM+’s batch mode. The batch mode is a way of executing STAR-CCM+ on a server environment. An example of executing STAR-CCM+ in batch script mode is included in the STAR-CCM+ submission script.

The main way that CONSTELATION communicates with STAR-CCM+ is using tables, specifically csv files. STAR-CCM+ has the capability of automatically importing and exporting data during an active simulation using tables. These tables consist of tabular data that is defined using field functions. Field functions are how STAR-CCM+ can store data, such as temperature and density throughout a simulation, they can be in the form of scalars or vectors. When exporting data from a STAR-CCM+ simulation there are three things to consider:

1. the field functions being extracted
2. the derived parts from where that data is being extracted
3. when the data should be extracted.

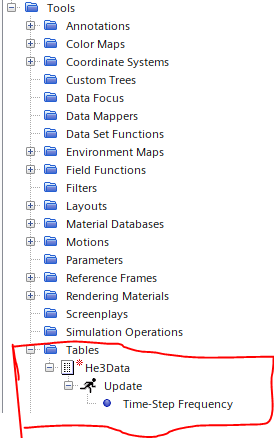
The desired field functions in the HENRI TREAT coupling are the density, temperature, and relative position of the He-3 gas within the HENRI capsule. The density, temperature, and relative position will be used by SERPENT 2 in order to calculate the overall volumetric heating that occurs within the He-3 while it is being injected into the TREAT reactor.

The derived part option defines a specific part that can track and sample the field function data. The line-probe derived part was used in the HENRI TREAT coupling to extract the density and temperature data of the helium-3 gas. A line-probe is a STAR-CCM+ part that defines a line of points located at a user-defined position and with a user-defined resolution. The resolution corresponds to the number of data points used to make the line-probe. Each data point when asked to export data will export its location, and then the requested field function data. An example of the line-probe used in HENRI-TREAT is shown here:



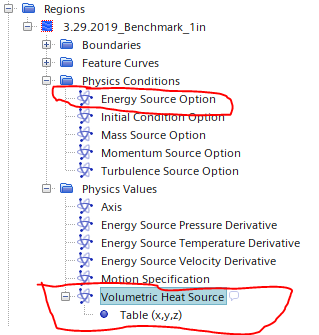
The update function found within the table function can be used to define a time-step, iteration, or delta-time frequency over which the requested data is extracted. The time-step option was chosen to extract the information over. This extracts the density, temperature, and relative position of the He-3 gas after a user-defined number of time-steps.

Tables are defined under the tool function, an example using the HENRI STAR-CCM+ simulation is shown here:



These tables are used to extract information from STAR-CCM+. To import information, specifically Volumetric Heating, the Energy Source Option must be defined. This can be found under Regions>USER DEFINED REGION > Physics Conditions> Energy Source Option. Then the Volumetric Heating Source Option should be selected. Once this is selected an option under the Physics values tab should become available called Volumetric Heat Source. It is here that the name and location of the Volumetric Heating csv file created by SERPENT2 can be defined. This is automatically handled within the javaMACRO included in the appendix.

Volumetric Heating is used as it is the intended power output needed for the coupling to work AND it is the only option that STAR-CCM+ has available that can import data along x,y,z coordinates. An example of how these options are defined is shown here:



Once all these options have been defined STAR-CCM+ will automatically export the requested data in the form of a .csv file. This file can then be read and manipulated by CONSTELATION with the relevant data being passed to SERPENT 2 in the form of a multi-physic interface file.

When this csv file is created it signals the end of the STAR-CCM+ side of the coupling. This is accomplished within the javaMACRO by creating a STAR\_done.txt file once the defined number of time-steps within STAR-CCM+ have been completed. CONSTELATION will then read this file and update the multi-physic interface file so that the next time-step of SERPENT 2 can be executed.

1. CONSTELATION algorithm

The CONSTELATION script performs a few necessary functions. At the start of the simulation, it creates the initial interface files required by SERPENT2. After writing these files it sends an execution signal to both SERPENT2 and STAR-CCM+. Once one time-step is completed by each code, a set of output files is created. The output created by STAR-CCM+ is a csv file containing the temperature and density distribution of the helium-3 gas. The output created by SERPENT2 is a MATLAB file containing the volumetric heating of the helium-3 gas. CONSTELATION then manipulates these output files into respective input files that can be read by STAR-CCM+ and SERPENT2 respectively. The input file for STAR-CCM+ is a csv file containing the volumetric heating calculated by SERPENT 2. The input file for SERPENT2 is an updated interface file with the new density and temperature distribution calculated by STAR-CCM+. The simplified algorithm of CONSTELATION is depicted in the following figure.

Start CONSTELATIONN

Set up coupled Serpent 2 File

Execute Serpent 2

Collect Output

Pause Serpent 2

Transform Serpent 2 Output

Execute STAR-CCM+

Has simulation time elapsed?

End Code

Transform STAR-CCM+ Output

Replace multi-physic file with new output

Yes

No

Prior to the coupled calculation, a standalone STAR-CCM+ calculation is performed for one timestep to obtain an initial density and temperature distribution. Similarly, an initial volumetric heating profile is created by SERPENT 2 to be read by STAR-CCM+ to initialize the simulation.

1. CONTSTELATION modification

As it currently stands CONSTELATION only works for a model of HENRI within the TREAT reactor. However, the overall steps used in the coupling process can be conceivably expanded to any STAR-CCM+ simulation with any SERPENT 2 base model file. The main things that would need to be changed are the multi-physic interface files inputs created with CONSTELATION, the name of all the files, the time-step, and the number of x,y,z points used.

1. Step by Step process
2. Acquire SERPENT 2, STAR-CCM+, and Python 2.7
3. Acquire STAR-CCM+ model and SERPENT 2 base model
4. Modify STAR-CCM+ to output csv of density and temperature distribution after desired number of timesteps
5. Modify SERPENT 2 model to output desired data (most likely Power of some kind) either through detector or multi-physic output file
6. Execute SERPENT 2 in source generation mode
7. Create set of transient inputs that match time-step of STAR-CCM+ output.
8. Modify CONSTELATION script to match STAR-CCM+ and SERPENT 2 outputs
9. Execute CONSTELATION