Structure of Gale

- **Gale/**
  - README, RELEASE_NOTES, INSTALL*
  - MIGRATION, UPGRADE
  - COPYING
  - tools/
    - configure*, *.in, meta-template.c, Sconstruct, config/, python/
  - documentation/
  - input/
    - benchmarks/, cookbook/, examples/
  - src/
src/

- StGermain/
  - Basic utilities
- StgFEM/
  - Finite elements, solvers
- PICellerator/
  - particles
- Underworld/
  - Rheologies
- Gale/
  - Main executable, extras
StGermain/

- Base/
- Discretisation/
- FD/
  - Built, but not used
- SConscript
  - Like a makefile
- compatibility/
  - Handles different versions of Petsc
- doc/
  - Only for automatically generated documentation
StGermain/

- libStGermain/
  - For building a StGermain library
- libStGermainAll/, pyre/, Regresstor/, Services/, src/, Viewers/
  - Not built
StGermain/Base

- Automation/
- Container/
- Context/
- Extensibility/
- Foundation/
- IO/
StGermain/Base

- FlattenXML/
  - Executable to flatten input files into one, big file
- ListPlugins/
  - Executable to list all of the plugins that an input file uses.
- src/
  - For making a StGermainBase library
- Python/, tests/
  - Not built
StGermain/Base/Automation

- SConscript
- src/
- tests/
  - Not built
StGermain/Base/Automation

- Automation.h, Init.c, Init.h, Finalise.c, Finalise.h, types.h, shortcuts.h
  - Code to make a StGermainBaseAutomation library
- offsets.h
  - pointer voodoo
StGermain/Base/Automation

- All of the remaining files come in two's or three's: foo.c, foo.h, foo.meta
- These are meant to be objects as in C++, Java, etc.
- But this is C, not C++. So there are some macro hacks to emulate it.
#define __SetVC
 /* General info */
   __VariableCondition
 /* Virtual info */
 /* Stg_Class info */
 Name                          _dictionaryEntryName;
 SetVC_Entry_Index             _entryCount;
 SetVC_Entry*                  _entryTbl;
 IndexSet*                     _vcset;

 struct _SetVC { __SetVC };
*.meta files

- *.meta files are processed to create components.
- Components are the things that you enumerate in an input file.
StGermain/Base/Automation: components

- CompositeVC
- SetVC
- Stg_Component
- VariableAllVC
- VariableCondition
- Variable
StGermain/Base/Automation: other objects

- CallGraph
- ComponentRepository
- DocumentationComponentFactory
- LiveComponentRegister
- VariableDumpStream
- Variable_Register
- VariableCondition_Register
StGermain/Base/Automation: other objects

- RegisterFactory
- Stg_ComponentMeta
- Stg_ComponentRegister
- Stg_ComponentFactory
- ConditionFunction
- ConditionFunction_Register
- HierarchyTable
StGermain/Base/Container

- Array
- BTree
- HashTable
- Heap
- Map
- Set
- List
- Map
- MPI convenience functions
• AbstractContext
  – This is like a global variable that serves as the root of the object tree.
  – StgFEM, PICellerator, and Underworld all derive from this class to define their own “Context”.
StGermain/Base/Extensibility

- **EntryPoints**
  - These components are a way of having the code execute a list of functions, determined at run time.
  - For example, the Stokes equation adds an EntryPoint that calculates the size of the timestep.

- **Plugins**
  - Allows code to be loaded dynamically from a library depending on the input file
StGermain/Base/Foundation

- Memory routines
- Object, Class
- Other utilities
StGermain/Base/IO

- Read and parse input files
- Formatted output
  - Regular and Debug streams
  - Only one set of output for all processors
StGermain/Discretisation

• Geometry
  – Mesh Topology: Delaunay, IJK
  – Trig, Vector, Tensor, complex math

• Mesh
  – Create Meshes and sync them in parallel
  – SurfaceAdaptor implements distorted upper boundaries.

• Shape
  – Box, Cylinder, Sphere, Union, Intersection, etc.
StGermain/Discretisation

• Swarm
  – Particle layout, particle exchange between processors, particle variables

• Utils
  – Initial condition, boundary conditions, and fixed internal degrees of freedom
StgFEM

• Apps
  – Input files: not guaranteed to work. They often require gLucifer

• Templates
  – License templates for individual files

• src
  – Not built

• libStG_FEM
  – Makes StgFEM library
StgFEM

- Assembly
  - Assembles Gradient, Isoviscous, Laplacian, and Thermal Buoyancy terms

- Discretisation
  - Finite Element Mesh, constant and linear elements, FE variables, FE-Swarm variables

- SLE

- plugins
StgFEM/SLE

• **LinearAlgebra**
  - Mostly an interface to Petsc

• **MultiGrid**
  - Broken

• **SystemSetup**
  - Sets up the system of linear or nonlinear equations
StgFEM/SLE/ProvidedSystems

- AdvectionDiffusion
- Energy
- StokesFlow
  - Uzawa
  - Penalty – untested
  - UpdateDt
StgFEM/plugins

- LevelSetPlugin, MultiGrid, Application
  - Not built
- Document
  - Prints out info about all components
- CompareFeVariableAgainstReferenceSolution, VelicAnalyticSolutions
  - Analytic solutions
- FeVariableImportExporters
  - Interchange with Abaqus: Untested
StgFEM/plugins

• Output
  – Print out various statistics: CPUPTime, Memory, FE variables.

• StandardConditionFunctions
  – A large number of mathematical convenience functions (e.g. for initial conditions)
PICellerator

• MaterialPoints
  – Interpolating between grid and particles
  – particle advection

• PopulationControl
  – Creates and destroys particles as needed

• Voronoi
  – Voronoi cells are created from the particles positions and particle values are weighted accordingly

• Weights
  – Handles how particles are weighted.
PICellerator

- Apps
  - Probably non-working input files
- Utils
  - Buoyancy Force
  - Hydrostatic Term
    - Subtracts out a background hydrostatic pressure
- Plugins
  - CalculateParticleDisplacement
  - Output/MaterialCentroid
Underworld

- **InputFiles**
  - Probably broken

- **Rheology**
  - Von Mises, Drucker Prager, Anisotropic, etc.

- **Utils**
  - Radiogenic heating
  - Pressure & Temp output for particles
  - DensityField, StressField, and ViscosityField
Underworld

• plugins
  – EulerDeform
    • Moves the mesh after every timestep so that it covers the new location of the particles.
    • This would be the place to start if you want a poor man's adaptive meshing.
  – Output
    • VTKOutput: the workhorse output routine
  – Everything else is untested.
Gale

- src/
  - Contains main()

- plugins
  - SurfaceProcess
    - Does not work fully in parallel yet

- Utils
  - Q1-Q1 stabilization term
  - Hydrostatic Correction
  - Divergence Forces
  - Stress BC
  - Kinetic Friction
Important Parts

- StGermain
  - Base
    - Automation
    - Container
    - Context
    - Extensibility
    - Foundation
    - IO
  - Discretisation
    - Geometry
    - Mesh
    - Shape
    - Swarm
    - Utils

- StgFEM
  - Assembly
  - Discretisation
  - SLE
    - LinearAlgebra
    - SystemSetup
    - ProvidedSystems
      - AdvectionDiffusion
      - Energy
      - StokesFlow
  - plugins
    - StandardConditionFunctions
Important Parts

• PICellerator
  – MaterialPoints
  – PopulationControl
  – Voronoi
  – Weights
  – Utils
  – plugins

• Underworld
  – Rheology
  – Utils
  – plugins
    • EulerDeform
    • VTKOutput

• Gale
  – src
  – Plugins
    • SurfaceProcess
  – Utils
Control Flow

- It all starts in src/Gale/src/main.c
  - Initializes MPI
  - Reads the input file
  - Dynamically determines what to do next
  - Following the execution in a debugger is basically impossible