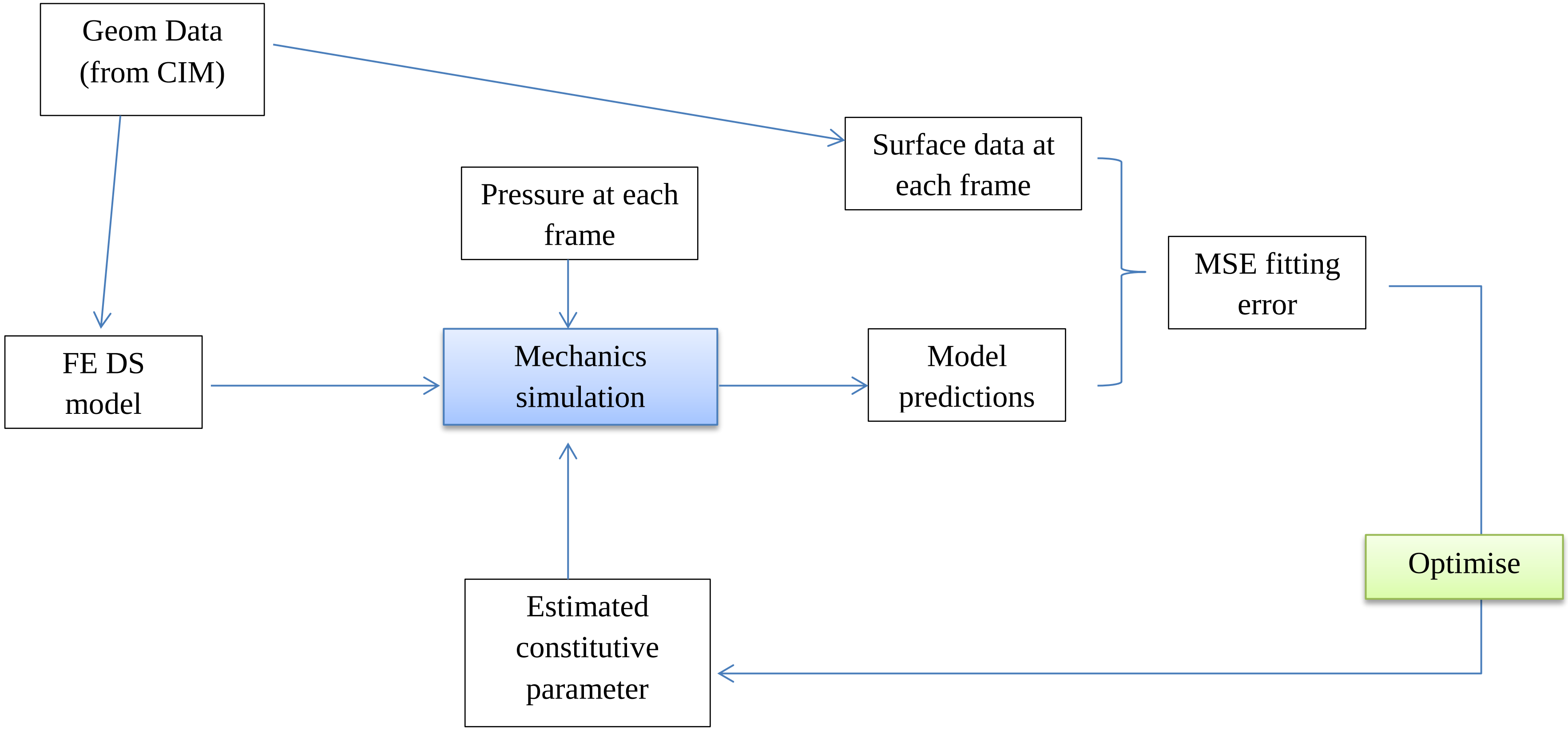
St. Francis Parameter Estimation Framework – May 2015

Program overview



Main.py

This script is the top-level control of the entire simulation.
INPUTS: command line input:

1. Study number: starting from zero, corresponding to the position of the desired study as listed in NYStFranFrameNumber.txt.
2. Debug: toggles between 1 – output written to text file Output.log in study folder, or 0 – output written to screen.
3. Forward solve (passive only): Controls whether to run a forward solve from diastasis to end diastole before optimising. This should be set to 1 when running the study for the first time. When set to 0, the programme skips to optimisation assuming the forward solve has been completed.
4. Active-passive toggle: Controls which parameter estimation is run: 1. Passive parameter estimation only, 2. Active parameter estimation only, 3. Passive followed by active parameter estimation.

The main program is then called which does the following:
Program set up:

- Get current study important frames (**setup_get_frames**).
- Set up directories for current study (**setup_dir**).
- Set up log text file if debug toggle is true (**setup_log_file**).

Geometry set up:

- Set up geometry of reference model and cavity models of the LV (**geom_setup_ref_cavity**).
- Handle file transfer of reference model and DS, ED and ES surface data to study folder (**geom_setup_data**).

Mechanics simulation set up:

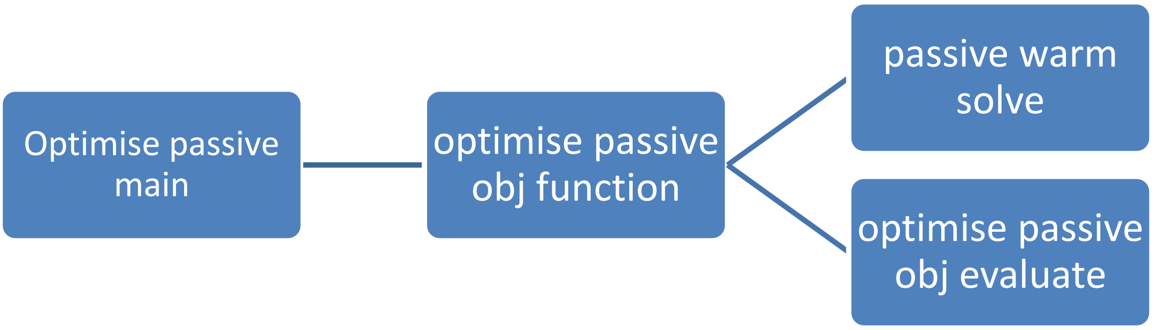
- Get pressure array from the results of haemodynamic processing (**bc_pressure_get**).
- Get data points that will be used to prescribe basal displacement boundary condition (**bc_displacement_get**).
- Handle file transfer of mechanical simulation template files into study folders (**mech_template_setup)**.

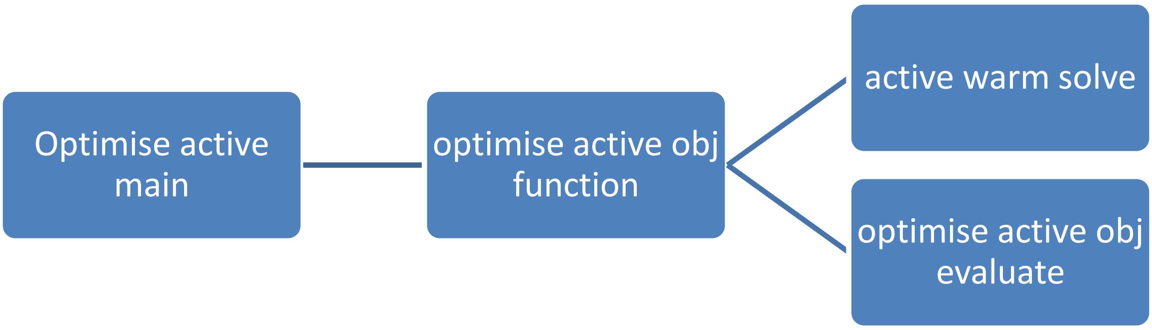
Run simulation:

- Passive parameter estimation (using Guccione transversely isotropic model and estimating bulk stiffness parameter using all frames from DS to ED) (**optimise_passive_main**).

- Active parameter estimation (using Hunter-McCulloch-ter Keurs steady-state active tension model, estimating bulk calcium-dependent muscle activation parameter at each frame from ED to DS) (**optimise_active_main**).

Optimise.py





Optimise_passive_main

This function implements the estimation of a bulk myocardial passive parameter (C1) which is obtained for each study subject by fitting model predictions to subject-specific image-derived geometries.

- Initial simulation from DS to ED taking an initial guess of the material parameter (**material_create_ipmate**). The model predictions at each frame from DS to ED are saved. (**passive_initial_solve**).
- Generates results file containing pressure, volume, endocardial and epicardial fitting errors at each frame of initial simulation. (**results_passive_generate**).
- Optimises bulk stiffness parameter (C1) to minimise fitting error for all frames from DS to ED using python optimiser fmin_l_bfgs_b function. The objective function for this optimisation is calculated in the function **optimise_passive_obj_function.**
- After optimisation is complete, evaluate final solution and fitting errors by running objective function one last time with optimised stiffness parameter (**optimise_passive_obj_function**).

Optimise_active_main

This function implements the estimation of a transient of bulk myocardial activation by estimating a single bulk activation parameter at every frame from ED through systole to DS.

- Get ED solution as starting point (**active_get_ED**).
- Loop through all frames from ED to DS and for each frame:
  - Initial simulation using an initial estimate of activation (T_{Ca}) (checking convergence).
  - Set upper and lower bounds for T_{Ca} parameter according to pressure step (**active_bounds_get**).
  - Optimise T_{Ca} parameter by minimising fitting error of model prediction to image-derived geometry at current frame.
- Generate optimised results file containing pressure, volume, fitting error and T_{Ca} value at each frame (**results_active_generate**).

Optimise_passive_obj_function

This function is the objective function which interfaces with the python optimiser for passive parameter estimation. It updates the bulk stiffness parameter and evaluates the overall fitting error objective function value all frames between DS to ED.

- Update current C1 estimate (**material_create_ipmate**).
- Re-run simulation using current C1 estimate to get new model predictions for each frame from DS to ED (**passive_warm_solve**).
- Evaluate mean squared error (MSE) of fitting (**optimise_passive_obj_evaluate**).

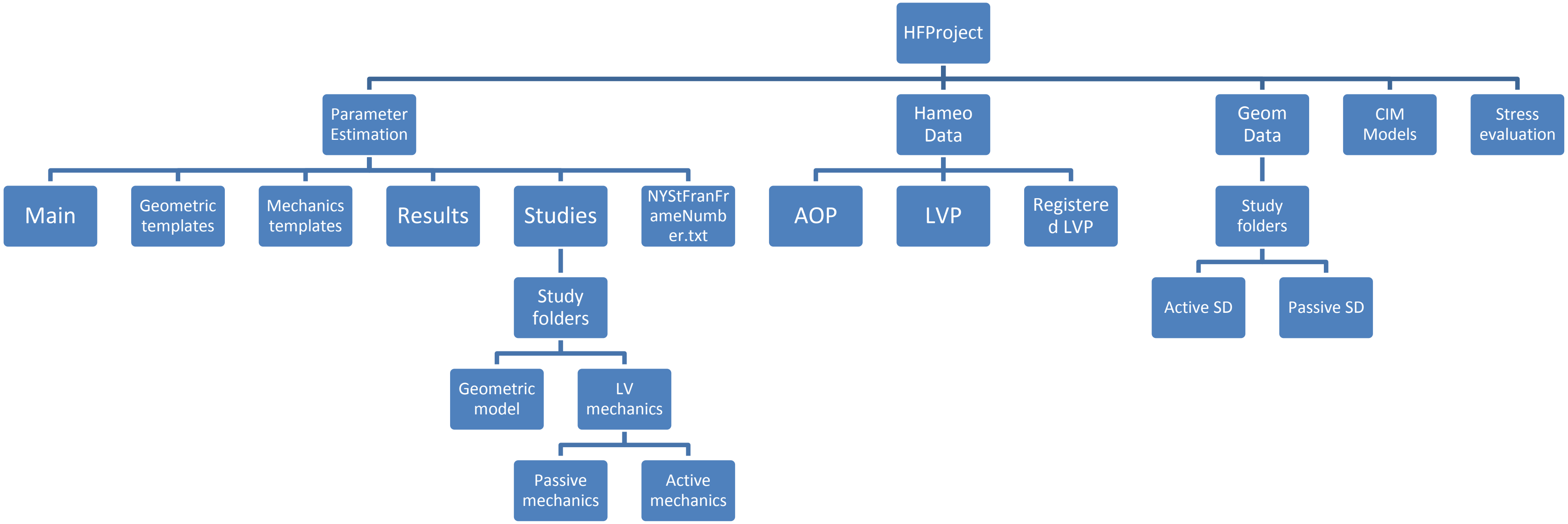
- Generates current optimised results file containing pressure, volume, endocardial and epicardial fitting errors at each frame of initial simulation (**results_passive_generate**).
- Returns MSE.

Optimise_active_obj_function
This function is the objective function which interfaces with the python optimiser for active parameter estimation. It updates the bulk activation parameter and evaluates the fitting error objective function at a specified frame.

- Update current T_{Ca} estimate (**activation_create_ipacti**).
- Re-run simulation using current T_{Ca} estimate to get new model prediction for specified frame (**active_warm_solve**).
- Evaluate MSE of fitting (**optimise_active_obj_evaluate**).
- Returns MSE.

Optimise_passive_obj_evaluate
This function extracts MSE of fitting for both endocardial and epicardial surfaces for all frames from DS to ED and sums them up.

Optimise_active_obj_evaluate
This function extracts MSE of fitting for both endocardial and epicardial surfaces and gets MSE for current frame.


HFProject
Parameter Estimation
Hameo Data
Geom Data
CIM Models
Stress evaluation
Main
Geometric templates
Mechanics templates
Results
Studies
NYStFranFrameNumber.txt
AOP
LVP
Registered LVP
Study folders
Study folders
Geometric model
LV mechanics
Passive mechanics
Active mechanics
Active SD
Passive SD