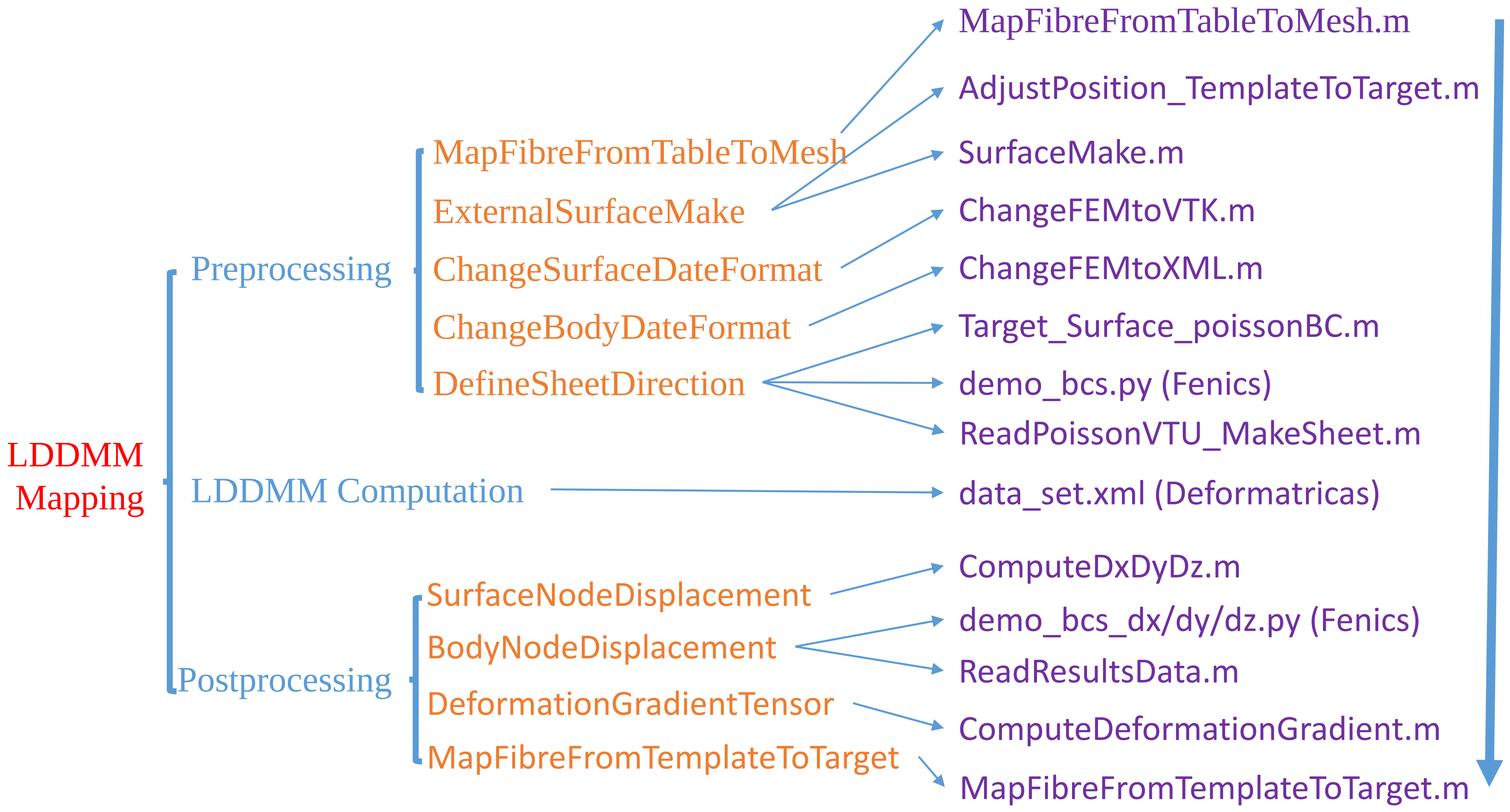
LDDMM Mapping
Preprocessing
MapFibreFromTableToMesh
ExternalSurfaceMake
ChangeSurfaceDateFormat
ChangeBodyDateFormat
DefineSheetDirection
LDDMM Computation
Postprocessing
SurfaceNodeDisplacement
BodyNodeDisplacement
DeformationGradientTensor
MapFibreFromTemplateToTarget
MapFibreFromTableToMesh.m
AdjustPosition_TemplateToTarget.m
SurfaceMake.m
ChangeFEMtoVTK.m
ChangeFEMtoXML.m
Target_Surface_poissonBC.m
demo_bcs.py (Fenics)
ReadPoissonVTU_MakeSheet.m
data_set.xml (Deformatricas)
ComputeDxDyDz.m
demo_bcs_dx/dy/dz.py (Fenics)
ReadResultsData.m
ComputeDeformationGradient.m
MapFibreFromTemplateToTarget.m

- Step 1: MapFibreFromTableToMesh/

MapFibreFromTableToMesh.m: Extract primary eigenvector of DT-MRI diffusion tensor as fibre direction **f** from datasets of Cardiovascular Research Grid, and map it to geometric element through comparing image pixel coordinate and central coordinate of element.

- Step 2: ExternalSurfaceMake

AdjustPosition_TemplateToTarget.m: Change the size and position of template mesh similar to that of target mesh.

SurfaceMake.m: Make surface mesh according to the mesh of geometry, because LDDMM is only can deal with surface mesh format.

- Step 3: ChangeSurfaceDateFormat

ChangeFEMtoVTK.m: Change the surface mesh format from .INP to .VTK because LDDMM algorithm only deals with .VTK file.

- Step 4: ChangeBodyDateFormat

ChangeFEMtoXML.m: Change the body mesh format from .INP to XML because Fenics algorithm only deals with .XML file.

- Step 5: DefineSheetDirection

Target_Surface_poissonBC.m: Sheet direction **s** is defined to be perpendicular to local surface, which is achieved by computing Poisson function when setting epicardium as 1 and endocardium as 0.

demo_bcs.py: Compute the Poisson function at Fenics environment and get gradient value at each node.

ReadPoissonVTU_MakeSheet.m: Compute gradient vector of each element as sheet direction.

- Step 6: LDDMM Computation

data_set.xml: Run LDDMM python code in ‘deformetricas’ conda environment when setting surface mesh of canine heart as initial template and surface mesh of neonatal porcine heart as target template.

- Step 7: SurfaceNodeDisplacement

ComputeDxDyDz.m: Compute the displacement of surface node before and after deformation as boundary condition of Poisson function.

- Step 8: BodyNodeDisplacement

demo_bcs_dx/dy/dz.py : To get displacement of every node in template mesh, run Poisson Function code in ‘fenics’ conda environment using the above boundary condition.

ReadResultsData.m: Summarize the above computing displacement at each node

- Step 9: DeformationGradientTensor

ComputeDeformationGradient.m: Compute the deformation gradient tensor **F** basing on the displacement of every node in the template mesh and then compute fibre direction after deformation according to $\mathbf{f}=\mathbf{F}\mathbf{f}_0$.

- Step 10: MapFibreFromTemplateToTarget

MapFibreFromTemplateToTarget.m: Map the deformed fibre from deformed template mesh to target mesh also through comparing the central coordinate of element, i.e., $\min|X\_{\text{deformed template}}-X\_{\text{target}}|$, where X is the central coordinate of element. After that, compute the sheet-normal direction basing on fibre and sheet direction.