GeneralisedElement

Virtual base class for all comph−lib elements.
The minimum requirement is that GeneralisedElements compute their contributions to the global residual vector and the global Jacobian matrix. GeneralisedElements do not necessarily have nodes.

FiniteElement

Virtual base class for all finite elements. This class adds storage for nodes, and pure virtual interfaces for shape functions, etc to the GeneralisedElement base class.

QElement

An example of a geometric element class: Implements the shape functions and the element's representation of its geometry −− the mapping between its local and global coordinates, based on interpolation between its nodal. coordinates.

AdvectionDiffusionEquations

Implements the computation of the element's residual vector and Jacobian matrix, based on the weak form of the advection diffusion equation. This class contains the "abstract" mathematics −− everything is expressed in terms of (pure virtual) basis and test functions.

QAdvectionDiffusionElement

Fully functional, advection diffusion element.
Combines (by inheritance) the representation of the element geometry (from the QElement) with the computation of the element's residual vector and Jacobian matrix (from the AdvectionDiffusionEquations).
Implements the basis and test functions (defined as pure virtual functions in the AdvectionDiffusionEquations class) by defining them as the QElement's geometric shape function −− this corresponds to an iso−parametric Galerkin discretisation since the (same) functions are used:
− as basis functions for the representation of the element geometry
− as basis functions for the represention of the unknown function
− as the test functions in the weak form of the equations.

KEY:

Generic objects.

Equation−specific objects. If you formulate your own elements, you will have to implement these for your specific system of equations.