- read problem-specific boundary conditions as algebraic expressions

  ```
  sigma = 5.670374419e-8  # W m^2 / K^4 as in wikipedia
  e = 0.98      # non-dimensional
  T0 = 1000     # K
  Tinf = 300    # K

  BC left  T=T0
  BC right q=sigma*e*(Tinf^4-T(x,y,z)^4)
  ```

- access shape functions and its derivatives evaluated either at Gauss points or at arbitrary locations for computing elementary contributions to

  - stiffness matrix
  - mass matrix
  - right-hand side vector

  For example, this snippet would build the elemental stiffness matrix for the Laplace problem:

  ```c
  int build_laplace_Ki(element_t *e, unsigned int q) {
    double wdet = feenox_fem_compute_w_det_at_gauss(e, q);
    gsl_matrix *B = feenox_fem_compute_B_at_gauss(e, q);
    feenox_call(feenox_blas_BtB_accum(B, wdet, feenox.fem.Ki));
    return FEENOX_OK;
  }
  ```

  The calls for computing the weights and the matrices with the shape functions and/or their derivatives currently support first and second-order iso-geometric elements, but other element types can be added as well. More complex cases involving non-uniform material properties, volumetric sources, etc. can be found in the examples, tutorials and tests.

- solve the discretized equations using the appropriate PETSc (Balay et al., 1997, 2023) or SLEPc (Hernandez et al., 2005; Roman et al., 2023) objects, i.e.

  - KSP for linear static problems
  - SNES for non-linear static problems
  - TS for transient problems
  - EPS for eigenvalue problems

The particular functions that implement each problem type are located in subdirectories `src/pdes`, namely

- `laplace`
- `thermal`
- `mechanical`
- `modal`
- `neutron_diffusion`
- `neutron_sn`

Researchers with both knowledge of mathematical theory of finite elements and programming skills might, with the aid of the community, add support for other PDEs. They might do that by using one of these directories (say `laplace`) as a template and

1. replace every occurrence of `laplace` in symbol names with the name of the new PDE
2. modify the initialization functions in `init.c` and set
   - the names of the unknowns
   - the names of the material properties
   - the mathematical type and characteristics of problem
   - etc.