3. modify the contents of the elemental matrices in `bulk.c` in the FEM formulation of the problem being added
4. modify the contents of how the boundary conditions are parsed and set in `bc.c`
5. re-run `autogen.sh`, `./configure` and `make` to get a FeenoX executable with support for the new PDE.

The addition of non-trivial PDEs is not straightforward, but possible. The programming guide contains further details about how to contribute to the code base.

## Conclusions

FeenoX's main goal is to keep things simple as possible from the user's point of view without sacrificing flexibility. There exist other tools which are similar in functionality but differ in the way the problem is set up. For example, FeniCSx uses the Unified Form Language where the PDE being solved has to be written by the user in weak form (Alnæs et al., 2014). This approach is very flexible, but even simple problems end up with non-trivial input files so it does not fulfill the first requirement stated in the summary. As simple as it is, FeenoX is still pretty flexible. A proof of this fact is that its applications range from coupling neutronics with CFD in nuclear reactors (Vasconcelos et al., 2018) to providing a back end to web-based thermo-mechanical solvers.

## References

Alnæs, M. S., Logg, A., Ølgaard, K. B., Rognes, M. E., & Wells, G. N. (2014). Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Softw.*, *40*(2). https://doi.org/10.1145/2566630

Anderson, R., Andrej, J., Barker, A., Bramwell, J., Camier, J.-S., Cerveny, J., Dobrev, V., Dudouit, Y., Fisher, A., Kolev, Tz., Pazner, W., Stowell, M., Tomov, V., Akkerman, I., Dahm, J., Medina, D., & Zampini, S. (2021). MFEM: A modular finite element methods library. *Computers & Mathematics with Applications*, *81*, 42—74. https://doi.org/10.1016/j.camwa.2020.06.009

Ayachit, U. (2015). *The ParaView guide: A parallel visualization application*. Kitware. ISBN: 9781930934306

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., … Zhang, J. (2023). *PETSc/TAO users manual* (ANL-21/39 - Revision 3.19). Argonne National Laboratory. https://doi.org/10.2172/1968587

Balay, S., Gropp, W. D., McInnes, L. C., & Smith, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, & H. P. Langtangen (Eds.), *Modern software tools in scientific computing* (pp. 163–202). Birkhäuser Press. https://doi.org/10.1007/978-1-4612-1986-6_8

Baratta, I. A., Dean, J. P., Dokken, J. S., Habera, M., Hale, J. S., Richardson, C. N., Rognes, M. E., Scroggs, M. W., Sime, N., & Wells, G. N. (2023). *DOLFINx: The next generation FEniCS problem solving environment*. preprint. https://doi.org/10.5281/zenodo.10447666

Finite Element Methods & Standards (Great Britain), N. A. for. (1990). *NAFEMS: The standard NAFEMS benchmarks*. NAFEMS. https://books.google.com.ar/books?id=--qwHAAACAAJ

Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical*