11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

FEEL++ : A HIGH PERFORMANCE FINITE ELEMENT EMBEDDED LIBRARY INTO C++

Alexandre Ancel[1,4], Vincent Chabannes[1], Cécile Daversin[3,4,5], Guillaume Dollé[3,4], Vincent Doyeux[2], Jean-Marc Gratien[7], Vincent Huber[1,4], Mourad Ismail[2], Pierre Jolivet[1,6], Stéphane Priem[1,2], *Christophe Prud'homme[3,4], Abdoulaye Samake[1,3], Marcela Szopos[3,4], Ranine Tarabay[3,4], Christophe Trophime[5] and Stéphane Veys[1]

[1] Laboratoire Jean Kuntzmann, Université Joseph Fourier Grenoble 1, BP53 38041 Grenoble Cedex 9, France, {vincent.chabannes,pierre.jolivet,stephane.veys,abdoulaye.samake}@imag.fr
[2] Université Grenoble 1 / CNRS, Laboratoire Interdisciplinaire de Physique / UMR 5588. Grenoble, F-38041, France, {Mourad.Ismail,vincent.doyeux,stephane.priem}@ujf-grenoble.fr
[3] Université de Strasbourg / CNRS, IRMA / UMR 7501. Strasbourg, F-67000, France, {prudhomme,szopos}@math.unistra.fr,
[4] CeMoSiS, Center for Modeling and Simulation in Strasbourg, F-67000, France, {alexandre.ancel,vincent.huber}@cemosis.fr,
[5] LNCMI-G, CNRS-UJF-UPS-INSA,25 Av. des Martyrs, Grenoble, F-38042,(France) {cecile.daversin,christophe.trophime}@lncmi.cnrs.fr,
[6] Laboratoire Jacques-Louis Lions, Université de Pierre et Marie Curie. Paris, F-75005
[7] IFP Energies nouvelles, 1 & 4 avenue Bois Prau, 92852 Rueil-Malmaison CEDEX, France, j-marc.gratien@ifpen.fr

Key words: *computational modeling, finite element analysis, high performance computing*

We present our advances in developing a language specific to partial differential equations embedded in C++. We have been developing the Feel++ framework (Finite Element method Embedded Language in C++) to the point where it allows to use a very wide range of Galerkin methods and advanced numerical methods such as domain decomposition methods including mortar and three fields methods, fictitious domain methods or certified reduced basis. We shall present an overview of the various ingredients as well as some illustrations. The ingredients include a very expressive embedded language, seamless interpolation, mesh adaption, seamless parallelisation and automatic differentiation using Frechet differentiation. As to the illustrations, they exercise the versatility of the framework either by allowing the developement and/or numerical verification of (new) mathematical methods or the development of large multi-physics applications[1, 3, 4] — e.g. fluid-structure interaction using either an Arbitrary Lagrangian Eulerian formulation or a levelset based one; high field magnets modeling which involves electro-thermal, magnetostatics, mechanical and thermo-hydraulics model; ... — The range of users span from mechanical engineers in industry, physicists in complex fluids, computer scientists in

biomedical applications to applied mathematicians thanks to the shared common mathematical embedded language hiding linear algebra and computer science complexities.

Feel++ provides a mathematical kernel for solving partial differential equation using arbitrary order Galerkin methods (fem, sem, cG, dG,...) in 1D, 2D and 3D using simplices and hypercubes meshes [2] : (i) a polynomial library allowing for a wide range polynomial expansions including H_{div} and H_{curl} elements, (ii) a light interface to PETSc/SLEPc as well as a scalable in-house solution strategy (iii) a language for Galerkin methods starting with fundamental concepts such as function spaces, forms, operators, functionals and integrals, (iv) a framework that allows user codes to scale seamlessly from single core computation to thousands of cores and enables hybrid computing.

Feel++ takes advantage of the newest C++ standard (C++11) such as type inference and the Boost C++ Libraries such as the Boost.Parameter, Boost.Fusion or Boost.MPL and many more. We shall illustrate how these language enhancements and libraries allow for very concise, robust and expressive C++ codes. Finally we will illustrate on large scale applications the numerical behavior in terms of verification and validation.

We acknowledge PRACE for awarding us access to resource SuperMUC based in Germany at LRZ as well the French National Research Agency (ANR) for the funding of the projects Vivabrain and HAMM.

REFERENCES

[1] C. Caldini Queiros, V. Chabannes, M. Ismail, G. Pena, C. Prud'Homme, M. Szopos, R. Tarabay. Towards large-scale three-dimensional blood flow simulations in realistic geometries., *ESAIM Proc.*, Volume **43**, 195-212, 2013.

[2] C. Prudhomme, V. Chabannes, V. Doyeux, M. Ismail, A. Samake, G. Pena. Feel++: A Computational Framework for Galerkin Methods and Advanced Numerical Methods. *ESAIM Proc.*, Volume **38**, 429-455, 2012.

[3] C. Daversin, S. Veys, C. Trophime and C. Prudhomme. Reduced Basis Framework: Application to large scale non-linear multi-physics problems *Esaim Proc.*, Volume **43** p. 225, 2013

[4] A. Samake, S. Bertoluzza, M. Pennacchio, C. Prudhomme and C. Zaza. A Parallel Implementation of the Mortar Element Method in 2D and 3D *Esaim proc.*, Volume **43** p. 213