Based on Professor Carlos Felippa Lecture Notes

**Finite Element Programming**

vector-space.com

R. Robert Gajewski  
Division of Applied Computer Science  
Warsaw University of Technology  
www.il.pw.edu.pl/~rg  
R.Gajewski@il.pw.edu.pl

**FE + Numerical Methods**

- Finite Element / Numerical Methods with VectorSpace C++ Library (vs.lib)
- vs.lib is a math library in C++ with a set of linear algebra and integrable / differentiable objects.
- vs.lib is a rapid-proto-typing tool which makes programming in numerical applications as easy as writing mathematical expressions.

**Description of library**

- This library, with special emphasis on applications in object-oriented finite element methods, is in general for applications in advanced numerical analysis such as computational linear algebra, linear programming, unconstrained / constrained optimization, finite difference method, boundary element method, and variational methods. vs.lib is a product of VectorSpace Programming (VSP).

**Book cover**

- Preface and TOC
- C0 Type Objects and Computational Linear Algebra (Matrix Computation)
- C1 and C2 Type Objects and Numerical Optimization
- H0, H1, and H2 Type Objects and Variational Methods
- Finite Element Primer
- Advanced Finite Element
- Index

**O-O FEM World**

- FEACPP.
- OOFElie.
- OFELI.
- Diffpack.
- FEM_Object.
- ZeBuLoN.
- CM2.
- FEMOOP
- FE++.
- BUB++.
- TPROTEUS.
- VfeTools
  - (in DevTools component libraries).
- Kaskade
  - (listed in MGNet).
- FEM2DLIB
  - (Fortran 90).
O-O FEM World

- NAG Faelflo
  - (special purpose language—Fasttalk).
- FreeFEM
  - (special purpose language—Gfem)
- Object-Oriented Finite Element Analysis
- Object-Oriented Finite Element Programming
- ObjectFEM
  - (Ph.D. thesis; Univ. of Aalborg, Denmark).
- FEMLIB
  - (MS & Ph.D thesis; Univ. of Illinois, Urbana-Champaign).
- SOFER
  - (on-going project; Univ. of Wales, Swansea, UK).
- FEM-OO-C++
  - (Univ. of Colorado at Boulder).

Where is the trick/

- An Object-Oriented Finite Element Library
  - (fe.lib; free source code)
- VectorSpace C++ Library—vs.lib:
  - An Object-Oriented Numerical Programming Tool
  - [click here for prices and how to order vs.lib]

Revolutionary step

- The computer languages used in numerical programming nowadays are either Fortran or C.
- They are both type-compiled languages.
- These two languages only support simple arithmetic operations on scalars.
- On the other hand, symbolic languages can be made very expressive.

Advantages

- With the advent of the object-oriented programming paradigm, the high-flown mathematical objects can be modeled in C++.
  - The object-oriented numerical model has the advantages of both worlds.
  - The C++ code can be as expressive as the symbolic codes, while the underlying object models are consistent with the numerical computation algorithms.
  - This makes the C++ program using vs.lib a perfect Rapid-Proto-Typing tool.

Revolutionary step

- However, the symbolic languages are intrinsically too slow for large-scale numerical computation.
- Since the underlying constructs of the two types of languages are so different, a smooth reverse engineering (or optimization) from symbolic codes to type-compiled codes is simply impracticable.

Features of C++

- Two major features in C++, data abstraction and object-oriented method, are used intensively to model the finite element method.
- We show, in the design of fe.lib, that the object oriented analysis is applied to examine the potentially complicated object dependency relationships, and the object oriented design is the discipline enforced to manage the object dependency relationships.
Features of C++

- The result is that the object-oriented model in fe.lib is completely parallel to how a mathematician will describe what the finite element method is.
- The object-oriented finite element model in fe.lib is very general and very inclusive for encompassing ever widely ranging subject areas in finite element method.

Contrast

- In contrast, a plain finite element program in Fortran / C languages is known to easily become ugly and disorganized, after adding, e.g., transient, non-linear, matrix sub-structuring, ... and so on.
- As an object-oriented finite element library, fe.lib shows its beauty of being capable of undertaking much greater "impact of change", while still keeping its program structure simple and coherent.

Re-use

- This advantage is derived from the code-reuse and programming by specification inherited from the object-oriented programming paradigm.
- In fact, fe.lib is the most extensible finite element program known to date.