Chapter 11

VECFEM-SOLVER FOR NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

Lutz Grosz

Abstract This chapter presents the design and implementation of the VECFEM PSE for a class of systems of partial differential equations (PDEs) that can be formulated in a variational form. The underlying code has been implemented on a variety of supercomputers. The basic modules of VECFEM are the graphical user interface x vem, the code generator, and the kernel library. This system can be combined with other external software tools such as I-DEAS for the definition of the domain and AVS for the analysis of the results. The kernel library can be used independently, e.g., in simulation codes or within other PSEs.

1. INTRODUCTION

It is a big challenge to provide a powerful problem solving environment (PSE) for the solution of PDEs. The typical problem comes in form of a system of PDEs on a three dimensional domain and contains non-linear terms. The non-linearity makes it extremely complicated and sometimes impossible to select suitable solution methods in advance. Advanced CAD techniques have to be used to handle the complexity of specifying the domain. Some applications require large-scale finite element meshes, typically if the domain has a complex shape. To get a solution in reasonable time a supercomputer has to be used. The user expects to get access to this computing power through his PSE without dealing with the special requirements on numerical methods and data structures for a particular platform.

Efficiency on the processor architectures used, and scalability on parallel computers are desirable features of solution techniques provided within the PSE. The user is expecting easy applicability as well as high flexibility of the domain and the PDE. In addition, and perhaps most important, robustness and reliability for a very large class of problems has to be ensured. It is very obvious, that all requirements cannot be met optimally at the same time. In VECFEM a
well-balanced compromise of these requirements has been implemented. With other words: VECFEM does not provide an optimal solver for each PDE, but is an optimal tool to solve a large class of PDEs.

The basic modules of VECFEM are the graphical user interface x vem, the code generator, and the kernel library. This system can be combined with other external software tools such as I-DEAS for the definition of the domain and AVS for the analysis of the results. The kernel library can be used independently, eg. in simulation codes or within other PSEs.

This paper gives an overview on the concept of VECFEM and its graphical user interface x vem. For more details refer to (Grosz et al., 1994; Grosz and Schönauser, 1996) and the VECFEM home page at URL http://wwwmaths.anu.edu.au/vecfem.

2. PARTIAL DIFFERENTIAL EQUATIONS

VECFEM can be applied to a very general class of systems of non-linear PDEs of order two. The PDE may be time dependent. The domain \( \Omega \) may be an arbitrary one-, two- or three-dimensional set. The discretisation method is finite element method (FEM) which allow a high flexibility regarding the geometry of the domain.

The PDE is specified through its equivalent variational formulation. For instance in the case of a three-dimensional system of \( n_c \) steady PDEs the variational formulation for the sought solution \( \tilde{u} \) can have the form: find \( \tilde{u} : \Omega \rightarrow \mathbf{R}^{n_c} \) with

\[
\int_{\Omega} \bar{\nabla} v_i \cdot \tilde{F}_i(x, \tilde{u}, \bar{\nabla} \tilde{u}) + v_i G_i(x, \tilde{u}, \bar{\nabla} \tilde{u}) \, dx + \int_{\partial\Omega} v_i g_i(x, \tilde{u}) \, ds = 0 \tag{11.1}
\]

for all test functions \( v_i : \Omega \rightarrow \mathbf{R} \) and all components \( i = 1, \ldots, n_c \). The operator \( \bar{\nabla} := \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right) \) denotes the Nabla operator. The coefficient functions \( \tilde{F}_i, G_i \) and \( g_i \) may be non-linear. For time dependent problems they may also depend on the time \( t \) and the time derivative \( \tilde{u}_t \).

If the solution and the coefficient functions are sufficiently smooth the variational problem (11.1) is equivalent to the non-linear boundary value problem

\[
- \bar{\nabla} \cdot \tilde{F}_i(x, \tilde{u}, \bar{\nabla} \tilde{u}) + G_i(x, \tilde{u}, \bar{\nabla} \tilde{u}) = 0 \quad \text{on} \quad \Omega
\]

\[
\bar{n} \cdot \tilde{F}_i(x, \tilde{u}, \bar{\nabla} \tilde{u}) + g_i(x, \tilde{u}) = 0 \quad \text{on} \quad \partial\Omega \tag{11.2}
\]

for all \( i = 1, \ldots, n_c \), where \( \bar{n} \) denotes the outer normal field of the boundary \( \partial\Omega \) of the domain, see (Quarteroni and Valli, 1994).

Most modelling approaches base on continuity equations for fluxes \( \tilde{F}_i \) with source terms \( G_i \). The coefficient functions can have discontinuity and/or may contain the Dirac distributions, eg. through nodal forces. For these cases the variational formulation (11.1) is the mathematically correct formulation for the boundary value problem (11.2).