Preface

This book covers parametric finite element analysis (FEA), which has been in wide use since the early 1960’s, in combination with symbolic Matlab analysis for solving algebra and calculus problems. Parametric FEA employs a non-dimensional polynomial as a piecewise approximation of the spatial solution of some differential equation that governs an engineering application. Parametric FEA also uses the same or a similar polynomial and geometric control points on a geometric region to define the shape of the region. The essential boundary conditions are applied at points on the geometric region (and interpolated on its boundary).

This contrasts to the newer isogeometric FEA where the shape of a geometric region is first exactly modeled using Non-Uniform Rational B-Splines (NURBS). An isogeometric FEA assumes that the spatial solution is approximated by the same NURBS using solution values at the NURBS control points. For detailed information about that newer approach the reader should see “Isogeometric Analysis: Toward Integration of CAD and FEA” by J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, John Wiley, 2009.

In teaching the classic parametric FEA for more than four decades, I have observed that most of the difficulties and mistakes arise because students have forgotten some aspects of calculus, matrix algebra, and differential equation terminology that they typically studied in their first two years in college. Therefore, this presentation will begin with a review of those concepts. Many students also have difficulty in understanding how the parametric polynomials are derived and how various combinations of them are integrated in integrals with physical interpretations. Symbolic software has been around for about fifty years. However, such tools have not been user friendly until recently when it was included as a tool in the Matlab environment. Now Matlab can and will herein be utilized to execute both symbolic and numerical solutions formulated as finite element simulations.

The classic linear FEA theory and common applications are covered in detail, including their implementation with Matlab. The chapter on error analysis and h-adaptive mesh adaptivity introduces those advanced subjects and illustrates their revision of examples presented in the prior chapters.

While the FEA can yield exact or approximate analytic solutions, it usually requires an implementation on a digital computer for application to practical problems. Therefore, the reader should understand various programming approaches require by a numerical FEA. Unfortunately, most undergraduates today do not have training in efficient engineering programming. However, students do often have experience with the Matlab numerical environment, its matrix operations, and graphical outputs. Thus, the FEA algorithms necessary to solve several classes of engineering applications will be presented using the Matlab environment included in each application. In addition, a large library of heavily commented Matlab scripts for processing typical FEA applications is made available.