A framework for developing finite element codes for multi-disciplinary applications

Pooyan Dadvand

2007

ACTA DE QUALIFICACIÓ DE LA TESI DOCTORAL

Reunit el tribunal integrat pels sota signants per jutjar la tesi doctoral:

Títol de la tesi: .....

Autor de la tesi: .....

Acorda atorgar la qualificació de:

- ☐ No apte
- ☐ Aprovat
- ☐ Notable
- ☐ Excel·lent
- ☐ Excel·lent Cum Laude

Barcelona, …… de/d'…..…….. de .....….

El President

.....
(nom i cognoms)

El Secretari

.....
(nom i cognoms)

El vocal

.....
(nom i cognoms)

El vocal

.....
(nom i cognoms)

El vocal

.....
(nom i cognoms)

To Hengameh

Abstract

The world of computing simulation has experienced great progresses in recent years and requires more exigent multidisciplinary challenges to satisfy the new upcoming demands. Increasing the importance of solving multi-disciplinary problems makes developers put more attention to these problems and deal with difficulties involved in developing software in this area.

Conventional finite element codes have several difficulties in dealing with multi-disciplinary problems. Many of these codes are designed and implemented for solving a certain type of problems, generally involving a single field. Extending these codes to deal with another field of analysis usually consists of several problems and large amounts of modifications and implementations. Some typical difficulties are: predefined set of degrees of freedom per node, data structure with fixed set of defined variables, global list of variables for all entities, domain based interfaces, IO restriction in reading new data and writing new results and algorithm definition inside the code. A common approach is to connect different solvers via a master program which implements the interaction algorithms and also transfers data from one solver to another. This approach has been used successfully in practice but results duplicated implementation and redundant overhead of data storing and transferring which may be significant depending to the solvers data structure.

The objective of this thesis is to design and implement a framework for building multi-disciplinary finite element programs. Generality, reusability, extendibility, good performance and memory efficiency are considered to be the main points in design and implementation of this framework. Preparing the structure for team development is another objective because usually a team of experts in different fields are involved in the development of multi-disciplinary code.

Kratos, the framework created in this work, provides several tools for easy implementation of finite element applications and also provides a common platform for natural interaction of its applications in different ways. This is done not only by a number of innovations but also by collecting and reusing several existing works.

In this work an innovative variable base interface is designed and implemented which is used at different levels of abstraction and showed to be very clear and extendible. Another innovation is a very efficient and flexible data structure which can be used to store any type of data in a type-safe manner. An extendible IO is also created to overcome another bottleneck in dealing with multi-disciplinary problems. Collecting different concepts of existing works and adapting them to coupled problems is considered to be another innovation in this work. Examples are using an interpreter, different data organizations and variable number of dofs per node. The kernel and application approach is used to reduce the possible conflicts arising between developers of different fields and layers are designed to reflect the working space of different developers also considering their programming knowledge. Finally several technical details are applied in order to increase the performance and efficiency of Kratos which makes it practically usable.

This work is completed by demonstrating the framework's functionality in practice. First some classical single field applications like thermal, fluid and structural applications are implemented and used as benchmark to prove its performance. These applications are used to solve coupled problems in order to demonstrate the natural interaction facility provided by the framework. Finally some less classical coupled finite element algorithms are implemented to show its high flexibility and extendibility.

Resumen

El mundo de la simulación computacional ha experimentado un gran avance en los últimos años y cada día requiere desafíos multidisciplinares más exigentes para satisfacer las nuevas demandas. El aumento de la importancia por resolver problemas multidisciplinares hizo poner más atención a la resolución de estos problemas y a los problemas que éstos implican en el área de desarrollo de software.

Los códigos convencionales de elementos finitos tienen varias dificultades para enfrentar se con problemas multidisciplinares. Muchos de estos códigos se diseñan y desarrollan para solucionar ciertos tipos de problemas, implicando generalmente un solo campo. Ampliar estos códigos para resolver problemas en otros campos del análisis, normalmente es difícil y se necesitan grandes modificaciones. Los ejemplos más comunes son: grados de libertad predefinidos para los nodos, estructura de datos capaz de guardar sólo una serie de variables definidas, lista global de las variables para todas las entidades, interfaces basadas en los dominios, capacidad del Input/Ouput para leer nuevos datos o escribir nuevos resultados y definición del algoritmo dentro del código. Un método común para resolver estos problemas es conectar varios modulos de calculo a través de un programa principal que implemente los algoritmos de la interacción y también transfiera datos de un modulo de calculo a otro. Este método se ha utilizado en la práctica con éxito, pero resulta en muchas duplicaciones del código y exceso de almacenamiento y tiempo de ejecución, dependiendo de la estructura de datos de los modulos de calculo.

El objetivo de esta tesis es diseñar e implementar un marco general para el desarrollo programas de elementos finitos multidisciplinares. La generalidad, la reutilización, la capacidad de ampliación, el buen rendimiento y la eficiencia en el uso de la memoria por parte del codigo son considerados los puntos principales para el diseño e implementación de este marco. La preparación de esta estructura para un fácil desarrollo en equipo es otro objetivo importante, porque el desarrollo de un código multidisciplinar generalmente requiere expertos en diferentes campos trabajando juntos.

Kratos, el marco creado en este trabajo, proporciona distintas herramientas para una fácil implementación de aplicaciones basadas en el método de los elementos finitos. También proporciona una plataforma común para una interacción natural y de diferentes maneras entre sus aplicaciones. Esto no sólo está hecho innovando, sino que además se han recogido y usado varios trabajos existentes.

En este trabajo se diseña y se implementa una interface innovadora basada en variables, que se puede utilizar a diferentes niveles de abstracción y que ha demostrado ser muy clara y extensible. Otra innovación es una estructura de datos muy eficiente y flexible, que se puede utilizar para almacenar cualquier tipo de datos de manera "type-safe". También se ha creado un Input/Ouput extensible para superar otras dificultades en la resolución de problemas multidisciplinares. Otra innovación de este trabajo ha sido recoger e integrar diversos conceptos de trabajos ya existentes, adaptándolos a problemas acoplado.Esto incluye el uso de un intérprete, diversas organizaciones de datos y distinto número de grados de libertad por nodo. El concepto de núcleo y aplicación se utiliza para separar secciones del codigo y reducir posibles conflictos entre desarrolladores de diversos campos. Varias capas en la estructura de Kratos han sido diseñadas considerando los distintos niveles de programación de diferentes tipos de desarrolladores. Por último, se aplican varios detalles técnicos para aumentar el rendimiento y la eficacia de Kratos, convirtiendo lo en una herramienta muy útil para la resolución de problemas prácticos.

Este trabajo se concluye demostrando el funcionamiento de Kratos en varios ejemplos prácticos. Primero se utilizan algunas aplicaciones clásicas de un solo campo como prueba patrón de rendimiento. Después, estas aplicaciones se acoplan para resolver problemas multidisciplinares, demostrando la facilidad natural de la interacción proporcionada por Kratos. Finalmente se han implementado algunos algoritmos menos clásicos para demostrar su alta flexibilidad y capacidad.

Contents