



Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

I. Identificación del Proyecto

1.1. Datos Generales del Proyecto

Título:

Diseño, desarrollo e implementación de una arquitectura de software orientada a objetos basada en componentes para aplicaciones en computación científica

Título abreviado:

Software para aplicaciones en computación científica

Unidad Académica Ejecutora

Facultad de Ingeniería - Centro Regional Universitario Córdoba IUA - UNDEF						
Responsable:	José Domingo Cuozzo					
Dirección:	Calle:	Avenida Fuerza Aérea			N°:	6500
Localidad:	Córdoba		C.P.:	5000	Provincia:	Córdoba
Tel.:	0351 - 4335010		Correo Electrónico:	jcuozzo@iua.edu.ar		

Datos de contacto Director de Proyecto

Nombre:	Carlos Gustavo Sacco					
DNI:	20369499					
Dirección:	Calle:	Ricardo Santos			N°:	10500
Localidad:	La Calera		C.P.:	5147	Provincia:	Córdoba
Tel.:	0351-152367521		Correo Electrónico:	csacco@iua.edu.ar		

Otras Facultades de UNDEF u otras instituciones que intervienen

No corresponde						
Responsable:						
Dirección:	Calle:				N°:	
Localidad:			C.P.:		Provincia:	
Tel.:			Correo Electrónico:			

Tipo de línea de investigación¹

Continuidad de líneas de investigación

¹ Aclarar e identificar si se trata de la continuidad de una línea de investigación o una línea prioritaria

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

Características del Proyecto:

Tipo de Actividad ²	Investigación básica y aplicada
Disciplina	Computación científica
Campo de Aplicación	Ciencias Exactas y Naturales, Ingeniería y Tecnología

Palabras clave

Computación científica, Métodos numéricos, Problemas multi-física

1.2. Dirección del Proyecto

Director: (Acompañar CVar o SIGEVA actualizado)

Apellido y Nombres	Categoría				Grado académico alcanzado
	UNDEF	Incentivos	CONICET	RPIDFA	
Carlos Gustavo Sacco	-	-	-	Id-B1	Doctor en ingeniería

Codirector: (Acompañar CVar o SIGEVA actualizado)

Apellido y Nombres	Categoría				Grado académico alcanzado
	UNDEF	Incentivos	CONICET	RPIDFA	
Juan Pablo Giovacchini	-	-	-	-	Doctor en física

1.3. Duración del Proyecto:

Fecha de Inicio	Enero de 2019
Fecha de Finalización	Diciembre de 2019
Duración prevista en meses (máximo 12 meses)	12 meses

² Investigación Básica / Investigación Aplicada / Desarrollo Experimental / Innovación Tecnológica.

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

II. Integrantes Equipo de Trabajo

2.1 Recursos Humanos

Integrantes Equipo de Trabajo (Acompañar CV abreviado de c/u)

Apellido y Nombres	Docente/Investigador (cargo/área de trabajo/facultad)	Estudiante (condición/nivel de carrera)	Personal de Apoyo y Técnico (función/lugar)	Otra Facultad UNDEF	Otras Instituciones (especificar)
Sacco Carlos Gustavo	Director de departamento Mecánica Aeronáutica / Métodos Numéricos / Facultad de Ingeniería	-	-	-	-
Weht Germán	Docente - investigador / Métodos Numéricos / Facultad de Ingeniería	-	-	-	-
Airaudó Facundo Nicolás	-	Ingeniería Aeronáutica – 4to año	-	-	-
Giovacchini Juan Pablo	Docente - investigador / Métodos Numéricos / Facultad de Ingeniería	-	-	-	-

Si correspondiera, consignar becas y/o pasantías relacionadas con el proyecto

Apellido y Nombres	Tipo de Beca / Pasantía	Institución otorgante / Unidad Académica	Período
Airaudó Facundo Nicolás	Ayudante de Investigación	CRUC-IUA	2018-2019

III. Plan de Investigación

3.1. Elaboración del proyecto

Resumen Técnico³

El objetivo del proyecto consiste en diseñar, desarrollar e implementar una arquitectura de software orientada a objetos (object-oriented framework architecture) basada en componentes para la resolución de problemas multi-física. Se exploran, definen y desarrollan metodologías de software para aplicar en el

³ Hasta 500 palabras

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

dominio de la computación científica.

El software resultante estará formado por el ensamble versátil de partes elementales denominadas componentes. La combinación de estos componentes permitirán generar una herramienta de simulación dedicada a resolver problemas específicos caracterizados por ecuaciones diferenciales parciales particulares. Se propone analizar y diseñar una arquitectura que simplifique el desarrollo de un software multi-método y multi-física, permitiendo e incrementando la colaboración entre diferentes campos de la computación científica. Se busca desarrollar algoritmos con características de generalidad, que se puedan extender y mantener en forma sencilla y que presente buenas performances numéricas.

Se desarrollarán e implementarán los componentes necesarios para resolver un problema de aplicación particular: transferencia de calor en un material compuesto.

Este problema consiste en analizar la transferencia de calor en un apilamiento de múltiples materiales con propiedades físicas/térmicas disimiles. Como método numérico se utilizará el método de elementos finitos; como modelo físico, el de transferencia de calor en sólidos.

Estado actual del conocimiento sobre el tema⁴

La computación científica es una de las herramientas más importantes para investigación y desarrollo en varias áreas de la ciencia e ingeniería. Dentro del ámbito de la computación científica, la solución de ecuaciones diferenciales parciales (PDE) adquiere notoria relevancia, en especial debido a la gran cantidad de fenómenos físicos que son modelados por estas ecuaciones: mecánica de fluidos, mecánica de sólidos, transferencia de calor, electromagnetismo, por citar algunos. Estas ecuaciones son generalmente complejas o imposible de resolver analíticamente en dominios arbitrarios. Luego una variedad de métodos numéricos, particulares según las ecuaciones diferenciales, son utilizados para obtener soluciones aproximadas.

Un conjunto de PDE con relaciones constitutivas y condiciones de borde apropiadas forman un modelo físico que representa un fenómeno físico o problema específico. Existen problemas complejos donde interactúan múltiples fenómenos, cada uno representado por modelos físicos particulares. Por necesidad de la industria, y dado el incremento de la capacidad de cálculo por evolución de las computadoras, fue posible resolver problemas complejos que involucran múltiples fenómenos interactuando. Estos modelos multi-física fueron incrementando su complejidad, adicionando nuevos modelos, para representar problemas de interés con mayor precisión.

Los métodos numéricos utilizados para discretizar las ecuaciones diferenciales deben ser los apropiados según su característica matemática. De los más utilizados se pueden destacar: método de elementos finitos (MEF) [43, 44, 45, 24, 39, 25], método de diferencias finitas [32, 40], método de volúmenes finitos [33, 22, 21], métodos espectrales [20, 18], entre otros. Las ecuaciones diferenciales que describen los modelos físicos pueden tener en principio diferentes características

⁴ Hasta 2000 palabras

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

matemáticas. Luego en un problema multi-física pueden típicamente coexistir múltiples métodos numéricos, necesarios para discretizar las ecuaciones diferenciales de los diferentes modelos que lo componen.

Las primeras implementaciones de métodos numéricos para resolver problemas en ciencia e ingeniería utilizaron softwares desarrollados ad-hoc que representan en general un único modelo físico. Abordar problemas multi-física incrementa la complejidad de un software, en especial cuando se deben utilizar múltiples métodos numéricos para discretizar las diferentes ecuaciones diferenciales que modelan el problema. Un incremento adicional en la complejidad ocurre cuando el acoplamiento entre los múltiples modelos tiene varias soluciones factibles.

En el ámbito local, los integrantes del equipo de trabajo - pertenecientes al Departamento de Mecánica Aeronáutica (DMA) de la Facultad de Ingeniería del CRUC-IUA [1] - tienen antecedentes en investigación y desarrollo en métodos numéricos, desarrollos de software de computación científica, y uso de estos para la solución de problemas de aplicación en ciencia e ingeniería [42, 9, 8, 6, 14, 17, 16, 15, 7, 26, 12, 3]. Como resultado de estos trabajos, en el DMA se cuenta con diferentes software de cálculo numérico en diversas áreas: mecánica de fluidos, mecánica de sólidos (cálculo de estructuras), transferencia de calor, y electromagnetismo; siendo las tres primeras áreas las que alcanzaron mayor desarrollo. Estos han permitido abordar numerosos trabajos de investigación y desarrollo destinados a la defensa [4, 5, 2] y a la industria. Algunos problemas de aplicación que se abordaron son: determinación de características aerodinámica de automóviles, aeronaves (UAVs), generadores eólicos [19], simulación de transporte de elementos contaminantes [11], análisis de cascos de barcos y submarinos [23], análisis de motores eléctricos de reluctancia variable, análisis de cargas aerodinámicas sobre estructuras civiles, entre los más relevantes.

Estos desarrollos fueron aislados en su mayoría, no tuvieron una base común, y no fueron concebidos como algoritmos con la generalidad y documentación suficiente para ser reutilizados, adaptados, extendidos y mantenidos con facilidad. Dadas estas características, se tiene localmente las limitaciones típicas asociadas a la no utilización de metodologías de ingeniería de software apropiadas para resolver problemas que involucren el acoplamiento de múltiples fenómenos físicos.

Existen diferentes estrategias para resolver problemas multi-física. Una aproximación estándar es utilizar un software que gestione la interacción entre diferentes algoritmos que resuelven algún modelo físico puntual. Esta aproximación es limitada y generalmente ineficiente, con bajas performances computacionales. Frecuentemente se duplican implementaciones y la transferencia de datos suele ser engorrosa. Para acoplar los problemas puede ser necesario reescribir parte de los algoritmos implementados, siendo esto particularmente costoso ya que generalmente se utilizan estructuras de datos rígidas en el desarrollo de los softwares que se combinan.

Otra perspectiva, algo más moderna, es concebir el diseño y desarrollo utilizando una arquitectura de software reutilizable, extensible y flexible a nuevas implementaciones. Para lograr estos objetivos se utilizan diferentes metodologías de ingeniería de software (análisis y diseño orientado a objetos, design patterns, frameworks, arquitectura basada en componentes, entre otros.) que involucran una cantidad de conceptos abstractos interconectados. La metodología de programación orientada a

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

objetos [34, 31] es una técnica de ingeniería de software para modelar un sistema como un grupo de objetos y sus interacciones. Una clase define un grupo de objetos que tienen las mismas propiedades y comportamientos. En esta metodología se analizan las diferentes partes del sistema como cajas negras y se diseñan las interacciones entre ellas. Cuando el sistema es pequeño, las clases y objetos son suficientes para describir su arquitectura. Cuando el sistema es de una escala mayor, son necesarios otros niveles de abstracción que ayuden a reducir y manejar la complejidad. Design patterns [13] describe la relación e interacción entre entidades de menor escala, clases. Un framework es una abstracción superior, un conjunto de clases que cooperan de una forma abstracta y concreta para construir un software reutilizable. Reducen el tiempo de desarrollo de muchas aplicaciones aunque el costo de desarrollo inicial es considerable. Los componentes del software son unidades de desarrollo independientes que interactúan para formar un sistema [41, 34]. Son utilizados para incrementar la modularidad y proveer un conjunto de funcionalidades que se reutilizan en diferentes aplicaciones.

Existe una variedad de desafíos en el desarrollo e implementación de software para computación científica utilizando estas metodologías: Identificar la segmentación más conveniente del software en sus diferentes componentes; encontrar un diseño de la arquitectura del software que reduzca o minimice la complejidad de las interacciones entre los diferentes componentes, facilitando la extensibilidad del software y permitiendo adicionar nuevos componentes a un costo de desarrollo aceptable; desarrollar e implementar componentes que traten de manera independiente los métodos numéricos y el modelo físico, permitiendo combinaciones entre las diferentes opciones factibles; pueden ser algunos de los desafíos iniciales. Existen también requerimientos de eficiencia numérica. Los problemas típicos que se abordan utilizan en general métodos iterativos, estos requieren frecuentemente una gran cantidad de iteraciones para encontrar una solución. Dependiendo de la magnitud del problema, en ocasiones es posible que se requieran de días o semanas para encontrar una solución. Luego la eficiencia numérica es de extrema importancia y un requisito fundamental que deben satisfacer los algoritmos que se implementan.

Se pueden encontrar en la literatura diferentes aproximaciones y experiencias en el desarrollo de software para computación científica [27, 38, 30, 29, 28, 37, 35, 36, 10] que utilizan combinaciones apropiadas de estas metodologías; donde resuelven, al menos parcialmente, algunos de los problemas mencionados. Los software comerciales (ANSYS, COMSOL, etc) utilizan de manera casi exclusiva estas metodología de desarrollo.

Encontrar soluciones de diseño apropiadas para desarrollar e implementar en un software que permita abordar problemas multi-física, discretizados por múltiples métodos numéricos en dominios de geometrías arbitrarias complejas, y utilizando algoritmos computacionalmente eficientes es un desafío que sigue vigente.

Retornando al ámbito local, DMA (CRUC-IUA), basados en la experiencia previa, y con la necesidad de disponer de una herramienta numérica para resolver modelos multi-física de manera eficiente - utilizando múltiples métodos numéricos para discretizar diferentes modelos físicos en dominios de geometrías complejas - se pretende abordar, utilizando metodologías de desarrollo mencionadas, el diseño e implementación de un software que posea las características deseables citadas.

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

Referencias:

- [1] Centro Regional Universitario Córdoba - IUA, Universidad de la Defensa Nacional.
- [2] Piddef 035/12 - desarrollo e implementación de métodos basados en el modelo de lattice Boltzmann para aplicaciones en mecánica de fluidos.
- [3] C. G. Sacco S. C. Chan Chang A. Figueroa, G. Weht. Aerodinámica compresible inestacionaria. Tercer Congreso Argentina de Ingeniería Aeronáutica, Noviembre 2014.
- [4] Sacco Carlos G. Piddef 015/10 - desarrollo de un código de flujo compresible. 2010-2012.
- [5] Sacco Carlos G. Piddef 041/12 - aerodinámica de flujo compresible. 2012-2014.
- [6] Sacco Carlos G., Gonzalez Esteban, and Giuggioloni Franco. Análisis de la aerodinámica de un automóvil de competición. *Mecánica Computacional*, XXV:83–94, Noviembre 2006.
- [7] Antonelli Dino, Sacco Carlos G., and Tamagno José P. Flow simulations with ultra-low Reynolds numbers over rigid and flexible airfoils subject to heaving and flapping motions. *Journal of Applied Fluid Mechanics*, 10:165–183, March 2017.
- [8] Ortega E. and Sacco Carlos G. Solución de las ecuaciones de flujo compresible mediante el método de puntos finitos. *Mecánica Computacional*, XXIV:1493–1507, Noviembre 2005.
- [9] C. Sacco E. Oñate, S.R. Idelsohn and J. García. Stabilization of the numerical solution for the free surface wave equation in uid dynamics. *Proceeding of the IV Ecomas Computational Fluid Dynamics Conference*, September 1998.
- [10] Elmer. Csc - it center for science. elmer an open source multiphysical simulation software.
- [11] Gonzalez Esteban and Sacco Carlos G. Simulación numérica del transporte de contaminantes por el método de los elementos finitos. 09 2004.
- [12] Gonzalez Esteban, Sacco Carlos G., R Vrech, V Renzi, D García Lambas, and Pablo Recabarren. Numerical simulation of winds behaviour in macón site. 01 2009.
- [13] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [14] Carlos G. Sacco Mario A. D'Errico Germán Weht, Juan P. Giovacchini. Elementos finitos aplicado a flujo compresible con gas en equilibrio. *Mecánica Computacional*, XXX:547–562, Noviembre 2011.
- [15] Juan P. Giovacchini. Sedimentation analysis of small ice crystals by lattice boltzmann method. *Quarterly Journal of the Royal Meteorological Society*. QJ-17-0139.R2.
- [16] Juan P. Giovacchini and Omar E. Ortiz. Flow force and torque on submerged bodies in lattice-boltzmann methods via momentum exchange. *Phys. Rev. E*, 92:063302, Dec 2015.
- [17] Juan P. Giovacchini, Omar E. Ortiz, and Carlos Sacco. El método de lattice boltzmann con condiciones de borde en geometrías arbitrarias no regulares. *Mecánica Computacional*, XXXI:165–183, 11 2012.
- [18] David Gottlieb and Steven A. Orszag. *Numerical Analysis of Spectral Methods : Theory and Applications (CBMS-NSF Regional Conference Series in Applied Mathematics)*. Society for Industrial and Applied Mathematics, 1987.
- [19] Jeandrevín Griselda, Sacco Carlos G., Paoletti Carlos, and Preidikman Sergio. Simulación en cfd de los ensayos en túnel de viento de una pala en condición de operación rotor

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

estático. VI World Wind Energy Conference & Exhibition, 10 2007.

[20] Jan S. Hesthaven, Sigal Gottlieb, and David Gottlieb. Spectral Methods for Time-Dependent Problems (Cambridge Monographs on Applied and Computational Mathematics). Cambridge University Press, 2007.

[21] Charles Hirsch. Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows. Wiley, 1990.

[22] Charles Hirsch. Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics. Butterworth-Heinemann, 2007.

[23] Sergio R. Idelsohn, Eugenio Oñate, and Carlos Sacco. Finite element solution of free-surface ship-wave problems. International Journal for Numerical Methods in Engineering, 45(5):503–528, jun 1999.

[24] Reddy J. N. An Introduction to the Finite Element Method. MCGRAW HILL SERIES IN MECHANICAL ENGINEERING. McGraw-Hill Education, 2005.

[25] Jian-Ming Jin. The Finite Element Method in Electromagnetics. Wiley-IEEE Press, 2014.

[26] Inaudi Jose and Sacco Carlos G. Stochastic wind-load model for building vibration estimation using large-eddy cfd simulation and random turbulent flow generation algorithms. Mecánica Computacional, XXXV(12):595–618, Noviembre 2017.

[27] Geradin M Klapka I, Cardona A. Interpreter oofelie for pdes. European congress on computational methods in applied sciences and engineering (ECCOMAS), 2000.

[28] Andrea Lani, Tiago Quintino, Dries Kimpe, Herman Deconinck, Stefan Vandewalle, and Stefaan Poedts. The coolfluid framework: Design solutions for high performance object oriented scientific computing software. 2005.

[29] Andrea Lani, Tiago Quintino, Dries Kimpe, Herman Deconinck, Stefan Vandewalle, and Stefaan Poedts. Reusable object-oriented solutions for numerical simulation of PDEs in a high performance environment. Scientific Programming, 14(2):111–139, 2006.

[30] Andrea Lani, Nadege Villedie, Khalil Bensassi, Lilla Koloszar, Martin Vymazal, Sarp M. Yalim, and Marco Panesi. COOLFluid: an open computational platform for multi-physics simulation and research. jun 2013.

[31] Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). Prentice Hall, 2004.

[32] Randall LeVeque. Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied Mathematics). SIAM, Society for Industrial and Applied Mathematics, 2007.

[33] Randall J. LeVeque. Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[34] Bertrand Meyer. Object-Oriented Software Construction (Book/CD-ROM) (2nd Edition). Prentice Hall, 2000.

[35] OOFELIE. Open engineering, multidisciplinary and multiphysics virtual virtual prototyping, 2008.

[36] OpenFoam. Open source cfd software.

[37] OpenFoam. Object-oriented tools for solving pdes in complex geometries, 2008.

[38] Dadvand Pooyan, Rossi Riccardo, and Oñate Eugenio. An object-oriented environment for developing finite element codes for multi-disciplinary applications. Archives of

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

Computational Methods in Engineering, 17(3):253–297, September 2010.

[39] J. N. Reddy and D.K. Gartling. The Finite Element Method in Heat Transfer and Fluid Dynamics, Third Edition. Applied and Computational Mechanics. CRC Press, 2010.

[40] John Strikwerda. Finite Difference Schemes and Partial Differential Equations. Society for Industrial and Applied Mathematics, 2007.

[41] Clemens Szyperski. Component Software: Beyond Object-Oriented Programming (paperback) (2nd Edition) (Addison-wesley Component Software). Addison-Wesley Professional, 2002.

[42] Carlos Sacco y Sergio R. Idelsohn. Un método de puntos finitos en mecánica computacional. III Congreso de Métodos Numéricos en Ingeniería, 2:1255–1268, 1996.

[43] O. C. Zienkiewicz and R. L. Taylor. Finite Element Method: Volume 1, Fifth Edition. Butterworth-Heinemann, 2000.

[44] O. C. Zienkiewicz and R. L. Taylor. Finite Element Method: Volume 2, Fifth Edition. Butterworth-Heinemann, 2000.

[45] O. C. Zienkiewicz and R. L. Taylor. Finite Element Method: Volume 3, Fifth Edition. Butterworth-Heinemann, 2000.

Objetivos de la Investigación

Disponer de una herramienta numérica con todas las características deseadas, mencionadas en la sección anterior, es un objetivo que se pretende alcanzar en el largo plazo. El objetivo específico que se persigue en este programa está circunscrito a una etapa de iniciación del proyecto, donde se realizarán definiciones sobre la arquitectura básica del software, implementaciones de prototipos y test elementales de las diferentes partes que lo componen, entre otros. Se propone además el abordaje de un problema particular, de interés en ingeniería, con cierta complejidad tal que permita verificar el correcto funcionamiento y la evaluación de performances numéricas de las implementaciones.

Por motivo de claridad y simplicidad, se detallan debajo y se diferencian los objetivos en: **generales de largo plazo**, y en **específicos del proyecto** a los que se persiguen en los plazos previstos por este programa.

Objetivos generales de largo plazo

- Diseñar, desarrollar e implementar una arquitectura de software orientada a objetos (object-oriented framework architecture) basada en componentes para computación científica. Se busca maximizar tres cualidades del software: eficiencia, flexibilidad y usabilidad.
- Crear una arquitectura que simplifique el desarrollo de un software multi-método y multi-

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

física, permitiendo e incrementando la colaboración entre diferentes campos de la computación científica.

- Explorar, definir y desarrollar metodologías de software (OOA/D, framework architectures, Component based software construction, Generative Programming and Domain Engineering) para aplicar en el dominio de la computación científica. Muchas de las metodologías de interés fueron desarrolladas en otras áreas de aplicación y necesitan ser adaptadas para un uso efectivo en computación científica. Buscar soluciones a nuestro problema dentro del espectro de metodologías disponibles está fuera del alcance de este trabajo. Basados en experiencias previas y resultados de proyectos similares reducimos el espacio solución a algunas metodologías, luego se intenta encontrar dentro de estas cuales son las más efectivas para alcanzar los objetivos.
- Analizar alternativas y metodologías para una definición adecuada de los diferentes componentes del software. Explorar y definir estrategias para vincularlos de manera eficiente. Los componentes son una descomposición del sistema en bloques funcionales con interfaces bien definidas utilizadas para comunicarse entre sí. La idea principal es utilizar estos componentes como bloques elementales para construir diferentes aplicaciones, reduciendo de esta forma el esfuerzo de desarrollo. También se busca minimizar el número de interacciones entre los diferentes componentes del sistema. Esto reduce el costo de implementaciones para expandir e incluir nuevas funcionalidades.
- Explorar técnicas para reducir o eliminar la dependencia o acoplamiento entre los métodos numéricos y modelos físicos. El objetivo es desarrollar técnicas donde múltiples métodos puedan discretizar múltiples modelos físicos desarrollados de manera independiente.
- Formar recursos humanos en áreas de métodos numéricos. El objetivo es vincular este proyecto con trabajos finales de estudiantes de carreras de grado y posgrado afines.
- Disponer de una herramienta numérica que contribuya, con fines educativos, a la enseñanza de métodos numéricos. En la misma línea, que sirva como generadora de nuevos proyectos de investigación y desarrollo específicos en áreas de computación científica y en la solución de problemas de aplicación en ciencia e ingeniería.

Objetivos específicos del proyecto

- Diseñar, desarrollar e implementar un prototipo (o primera aproximación) de una arquitectura de software orientada a objetos basada en componentes para computación científica.

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

- Desarrollar los componentes necesarios para resolver un problema particular. Algunos componentes elementales a desarrollar están relacionados a: método numérico, modelo físico, dominio, sistema lineal, entre otros.
- Explorar estrategias y métodos genéricos (independientes del método numérico y modelo físico) para vincular los diferentes componentes.
- Diseñar, desarrollar e implementar los prototipos de componentes necesarios para resolver un problema particular: transferencia de calor en un material compuesto, un apilamiento de múltiples materiales con propiedades físicas/térmicas disimiles. En este problema cada material se trata como un subdominio. Estos subdominios se resuelven como problemas aislados, vinculados con el resto a través de condiciones de borde particulares impuestas por una interface apropiada.
- Implementar como método numérico el método de elementos finitos. Como modelo físico se utiliza el de transferencia de calor en sólidos.

Metodología

El proyecto que se presenta en este programa se divide en etapas como se detalla debajo. Si bien las secciones se solapan temporalmente (como se indica en el cronograma de actividades) con tareas secundarias, el comienzo y fin de cada una marcan una serie de hitos en la evolución del plan de trabajo propuesto.

En primer lugar, previo a la iniciación de cada etapa se arma un esquema detallado de tareas secundarias, se asignan responsables y se estipulan tiempos de ejecución. Para el comienzo de cada etapa y durante la evolución de la misma se proponen reuniones de proyecto para lanzar, controlar su evolución, y finalizar la etapa. Cada una de las etapas se organiza de manera estándar, tienen una introducción o iniciación donde se analiza y estudia la bibliografía disponible de cada tema relevante, una transición donde se ejecutan y realizan implementaciones y verificaciones propuestas. Finalmente un cierre donde se efectúan revisiones de diseño y documentación de las tareas realizadas junto con el código desarrollado. Son tres las metodologías de desarrollo de software aplicada frecuentemente: Modelo de Cascada, Prototipado e Incremental. Siendo el Modelo de Cascada el más utilizado.

Como lenguaje de programación se utilizarán una combinación de, o el que oportunamente se considere más conveniente entre: Fortran 90/95/2003, C++, y Python; no se descarta utilizar otros lenguajes de programación. Incluso hacer uso de los nuevos estándares de programación que

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

permiten código embebido, lenguaje C en Fortran o viceversa. Los software necesarios, compiladores, editores y librerías que se utilicen serán software con licencia open source o versiones libres para desarrollistas de software comerciales.

Para facilitar y gestionar el trabajo colaborativo de programación en equipo se implementa un software de control de versiones, siendo git el adoptado por el grupo de trabajo. Se trabajará sobre una plataforma de desarrollo colaborativo, como puede ser github, y/o en repositorios locales según disponibilidad y posibilidad de acceso externo.

Todas las implementaciones y software que se desarrolle se documentará con herramientas de documentación estándar, como doxygen, siguiendo un formato de buenas prácticas prefijado para el grupo de trabajo.

Todo el trabajo realizado se documentará en informes técnicos, donde se utilizará exclusivamente un editor de texto Latex. Se realizará al menos un informe técnico por etapa del proyecto, dando cuenta de las tareas realizadas y el cumplimiento de las tareas estipuladas. Además se realizará un manual de usuario, donde se buscará solucionar problemas típicos.

Definición de etapas con detalle de actividades del proyecto

Por motivo de claridad y organización, el cronograma de actividades del proyecto se segmenta en las etapas explicitadas a continuación.

- ETAPA 1 - Etapa de definiciones. Diseño y desarrollo de la arquitectura preliminar del software. Diagrama de clases y UML preliminares. Análisis, definición y diseño de componentes elementales y su contenido. Análisis y definición preliminar de la estructura de datos de componentes básicos. Definición e implementación de test de pruebas básicos de funcionamiento. Definición de la gestión y documentación del software. Informes y documentación preliminar del software.
- ETAPA 2 - Implementación de prototipos de componentes. Verificación de funcionamiento y performances numéricas utilizando casos de prueba típicos de la literatura. Test de patrones y esquemas elementales. Revisión de diseño de prototipos de componentes elementales. Informes y documentación preliminar del software.
- ETAPA 3 - Análisis, diseño e implementación de interfaces para ensamblar múltiples componentes. Análisis de métodos y estrategias para acoplar, de manera eficiente, diferentes subdominios del problema. Verificación de funcionamiento y performances numéricas

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

utilizando casos de prueba típicos de la literatura. Revisión de diseño de prototipos. Informes y documentación preliminar del software.

- ETAPA 4 - Implementación del problema de interés particular que se quiere resolver. Análisis y verificación de resultados. Revisión de diseño de prototipos. Informes y documentación preliminar del software.
- ETAPA 5 - Cierre del proyecto. Informes y documentación del software.

Indicadores (cuantitativos y/o cualitativos)

Los indicadores del proyecto que reflejan el avance y cumplimiento de las tareas propuestas serán: informes técnicos y documentación del software. Se consideran diferentes tipos de documentación, informes técnicos de avance e informes técnicos de finalización de cada etapa prevista en este proyecto.

El cumplimiento de las actividades previstas para cada etapa se reflejará con una serie de informes y resultados de la evaluación de casos de prueba típicos previstos.

Se confeccionarán informes técnicos que documenten las implementaciones relacionadas al software, e informes técnicos relacionados a la evaluación de casos de prueba típicos de la literatura.

3.2. Impacto del proyecto

Contribución al avance del conocimiento científico y tecnológico y/o transferencia al medio

Las tareas a desarrollar y el producto final de este proyecto tiene una contribución relevante a nivel local (DMA CRUC-IUA) ya que no se dispone de herramientas numéricas como las que se propone. La disponibilidad de esta herramienta permitirá a futuro abordar problemas más complejos de interacción entre múltiples fenómenos físicos. Se mejoran las capacidades de desarrollo de software gracias a la utilización de nuevas metodologías de trabajo colaborativo y el uso de estándares de programación.

Por otra parte este trabajo dará lugar a publicaciones y presentaciones en congresos que permitirán difundir las tareas de investigación que se llevan a cabo en la institución. Se prevé publicar los resultados en congresos de mecánica computacional: MECOM, ENIEF y CAIA por citar algunos de interés. En caso de obtener avances significativos y soluciones originales se publicarán los resultados en revistas especializadas con referato internacionales.

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

Contribución a la formación de recursos humanos

El proyecto que se presenta contribuye de manera directa a la formación de recursos humanos en áreas de método numéricos y computación científica. En este sentido, uno de los integrantes del equipo de trabajo es un estudiante de la carrera de Ingeniería Aeronáutica cursando los últimos años. Es factible y se estima que se incorporen alumnos de carreras de grado y posgrado afines que realicen sus trabajos finales enmarcados en tópicos incluidos en los objetivos de este proyecto. Disponer de una herramienta numérica para resolver los problemas que se proponen, no solo contribuye a las facilidades locales del DMA, sino que es un aporte valorable para ser incluido como parte herramienta de soporte en el dictado de materias de grado y posgrado de carreras afines.

Beneficiarios/Usuarios directos e indirectos de la propuesta

Los beneficiarios y usuarios directos de esta propuesta son los investigadores del DMA del CRUC-IUA y las demás facultades y departamentos de la Universidad de la Defensa Nacional que deseen utilizar estas herramientas de computación científica que se propone.

3.3. Cronograma de Actividades

Actividades	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
Etapa 1	x	x	x	x								
Etapa 2				x	x	x	x					
Etapa 3							x	x	x			
Etapa 4										x	x	x
Etapa 5												x

3.4. Conexión/Intercambio del proyecto con otros grupos de investigación de Facultades UNDEF y/u otras instituciones

En principio no se tiene prevista una conexión o intercambio con grupos de investigación de otras instituciones.
 Se tendrán conexiones con otros grupos de investigación de la Facultad de Ingeniería del CRUC-IUA. En especial se tienen previstas colaboraciones con grupos de investigación en áreas relacionadas a informática e ingeniería de software del departamento de Informática del CRUC-IUA.

IV. Presupuesto detallado del financiamiento solicitado y monto total que se necesita para viabilizar el proyecto

Programa de Acreditación y Financiamiento de Proyectos de Investigación

Formulario Guía para la presentación de proyectos

<i>Rubros elegibles</i>	<i>Concepto (desagregar gastos)</i>	<i>Monto Solicitado UNDEF</i>	<i>Otros aportes</i>	<i>Monto Total</i>
Insumos	Elementos de oficina, hojas, toner de impresora.	\$ 5000	-	\$5000
Bibliografía	Libros relacionados con el tema propuesto en este trabajo. Se estima la compra de 10 libros.	\$40000	-	\$40000
Servicios y/o Asistencias Técnicas Especializadas	No corresponde	-	-	-
Viajes y Viáticos	Viajes y viatico a congreso ENIEF 2019 para 2 personas.	\$10000	-	\$10000
Inscripción a Congresos y eventos.	Inscripción a congreso ENIEF 2019.	\$10000	-	\$10000
Equipamiento	Servidor de cálculo (WorkStation) para aplicaciones HPC.	\$60000		\$60000
Monto Total		\$125000		\$125000

(Nota: Los gastos de Viajes y Viáticos no podrá superar el 40% del presupuesto total solicitado)

Firma del Director

Aval Institucional