

Máster Universitario en Ingeniería Informática

Computación de Altas Prestaciones

Curso 2022-23

José Miguel Alonso
Ignacio Blanquer
Víctor M. García

Profesores

- ◆ Víctor Manuel García Mollá (vmgarcia@dsic.upv.es):
 - Despacho 2D02, Edificio DSIC (1F).
 - Tutorías a demanda.
- ◆ José Miguel Alonso Ábalos (jmalonso@dsic.upv.es):
 - Despacho 1D02, Edificio DSIC (1F).
 - Instituto de Instrumentación para Imagen Molecular (I3M), CPI, edificio 8B, acceso N, nivel 1, laboratorio 088.
 - Tutorías a demanda.
- ◆ Ignacio Blanquer Espert (iblanque@dsic.upv.es):
 - Despacho 1D04, Edificio DSIC (1F).
 - Instituto de Instrumentación para Imagen Molecular (I3M), CPI, edificio 8B, acceso N, nivel 1, laboratorio 088.
 - Tutorías a demanda.

Computación de Altas Prestaciones

- ◆ **Supercomputación:** Se realiza en los computadores más potentes para resolver los problemas científicos con mayores necesidades computacionales.
- ◆ **Computación Paralela:** Utiliza sistemas computacionales paralelos (todos en la actualidad). ¿Cuántos núcleos tiene nuestro móvil, tablet, portátil, PC de sobremesa, ...?
- ◆ **Computación de Altas Prestaciones:** Pretende obtener las máximas prestaciones del sistema computacional con el que se trabaja.

Competencias

- ◆ Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- ◆ Capacidad para diseñar y evaluar sistemas operativos y servidores, y aplicaciones y sistemas basados en computación distribuida.
- ◆ **Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.**
- ◆ **Capacidad para comprender y poder aplicar conocimientos avanzados de computación de altas prestaciones y métodos numéricos o computacionales a problemas de ingeniería.**

Organización

- ◆ Sesiones teóricas: (15 sesiones)
Profesor José M. Alonso, Ignacio Blanquer, Víctor M. García
Martes, de 19:30 a 20:30 , aula 1G 0.1
- ◆ Seminarios: (14 sesiones)
Profesores: José M. Alonso, Ignacio Blanquer, Víctor M. García
Horario: Jueves, de 17:30 a 19:30. Lugar: Laboratorio 7 del DSIC
- ◆ Sesiones Laboratorio: (10 sesiones)
Profesores: Ignacio Blanquer, Víctor M. García
Horario: Miércoles, de 17:30 a 19:00. Laboratorio 7 del DSIC

Calendario

Sesiones teóricas: Martes

Febrero: 14, 21, 28

Marzo: 7, 14, 21, 28

Abril: 4, 25

Mayo: 2, 9, 16, 23, 30

Sesiones prácticas: Miércoles

Marzo: 22, 29

Abril: 5, 19, 26

Mayo: 3, 10, 17, 24

Seminarios: Jueves

Febrero: 16, 23

Marzo: 2, 9, 23, 30

Abril: 20, 27

Mayo: 4, 11, 18, 25

Junio: 1, 15

*sesiones de evaluación

Computación de Altas Prestaciones:

Sesiones teóricas

◆ Temporización

Sesión 1. Presentación y motivación

Sesión 2. Algoritmos matriciales básicos; notación, Evaluación, optimización

Sesión 3. Optimización de código

Sesión 4. Algoritmos a bloques, memorias cache

Sesión 5. Computación con precisión finita. Representación en coma flotante.

Normas vectoriales y matriciales. Optimización de códigos matriciales

Sesión 6. Seminario Cloud computing

Sesión 7. Introducción Parallel Studio

Sesión 8. Seminario GPUs

Sesión 9. Sistemas triangulares, descomposición LU.

Sesión 10. Descomposición LU pivotación, aplicaciones

Sesión 11. Librerías Blas+Lapack, Optimización códigos Matlab.

Sesión 12. Matrices dispersas

Sesión 13. Problemas de mínimos cuadrados

Sesión 14. Descomposición SVD, aplicaciones

Computación de Altas Prestaciones:

Seminarios + Sesiones laboratorio

◆ Herramientas, entornos y librerías de Altas Prestaciones

1) Seminarios sobre temas generales de Computación de altas prestaciones: Matlab, Herramientas de altas prestaciones. Optimización de código, Vectorización, BLAS, LAPACK...

2) Seminarios temáticos (A impartir por profesor especializado)

2.1) Cloud Computing: Ignacio Blanquer 21, 22, 23, 28, 30 de Marzo, evaluación el 5 de Abril.

2.2) Programación GPUs (CUDA): Víctor García, 4, 19, 20, 26 y 27 de Abril, evaluación el 3 de Mayo.

2.3) Parallel Computing Toolbox, Víctor García, 4, 10, 11, 17 de Mayo, evaluación el 24 de Mayo

Evaluación de la asignatura

- ◆ Seminarios especializados: Se evalúan mediante trabajos en clase de laboratorio. *Se dedicarán a dicha evaluación tres sesiones de laboratorio (5 de Abril Cloud, 3 de Mayo GPUs, 24 Mayo parallel Toolbox). Al comienzo de cada una de esas tres sesiones al alumno se le propondrá la realización de un trabajo, que idealmente se debería poder acabar en la misma sesión. Si el profesor lo considera necesario al acabar dicha sesión podrá formular algunas preguntas al alumno sobre el trabajo llevado a cabo. Si no se ha podido acabar el trabajo en esa sesión, el alumno podrá acabar el trabajo a lo largo del día siguiente (con reducción de nota). Cada trabajo valdrá un 20% de la nota final.*
- ◆ Prueba escrita teoría **Peso en la calificación final: 20%**
+examen seminarios generales **Peso en la calificación final: 20%:**
 - **Examen: 15 Junio 2023**

Infraestructura

- ◆ Matlab (Windows, Linux) y programación en C, Cuda (Linux).
- ◆ Plataforma cloud Microsoft Azure, para el seminario de Cloud, utilizando unas credenciales que se os suministrarán antes de la sesión.
- ◆ Máquinas disponibles, a través de Polilabs o a través de Rdesktop:
 - Polilabs DSIC Windows (windesktop.dsic.upv.es)
 - Polilabs Linux (linuxdesktop.dsic.upv.es)
 - knights.dsic.upv.es (2 xeon, 24 cores, 1 Tesla K20)
 - gpu.dsic.upv.es (1 core i9, 16 cores, 2 Quadro RTX 5000)

Bibliografía

◆ Básica:

- Matrix Computations. G.Golub, C. Van Loan. Baltimore, London. Johns Hopkins University Press, 1996.

◆ Complementaria:

- Guide to Scientific Computing in C++. J. Pitt-Francis, J. Whiteley. London. Springer, cop. 2012. Recurso en línea.
- Numerical methods using Matlab. G. Lindfield & J. Penny. United States. Academic Press, 2012.

- ◆ En cada sesión se proporcionará bibliografía específica cuando sea necesaria.

Motivación

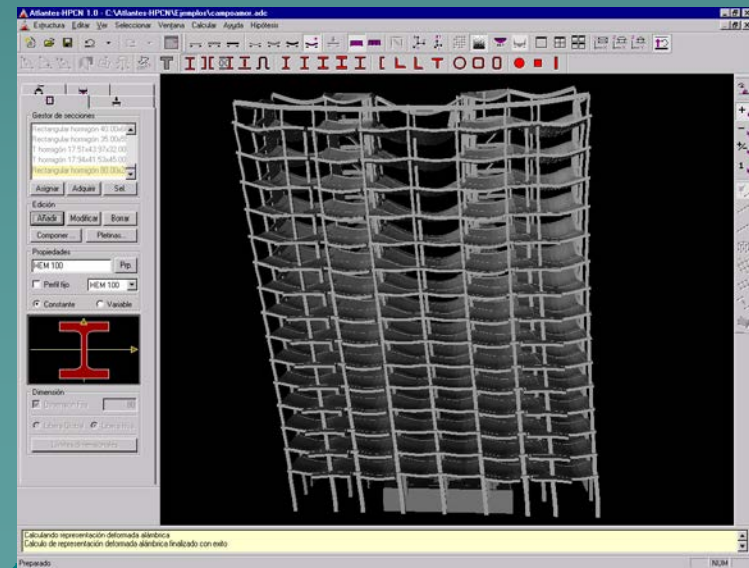
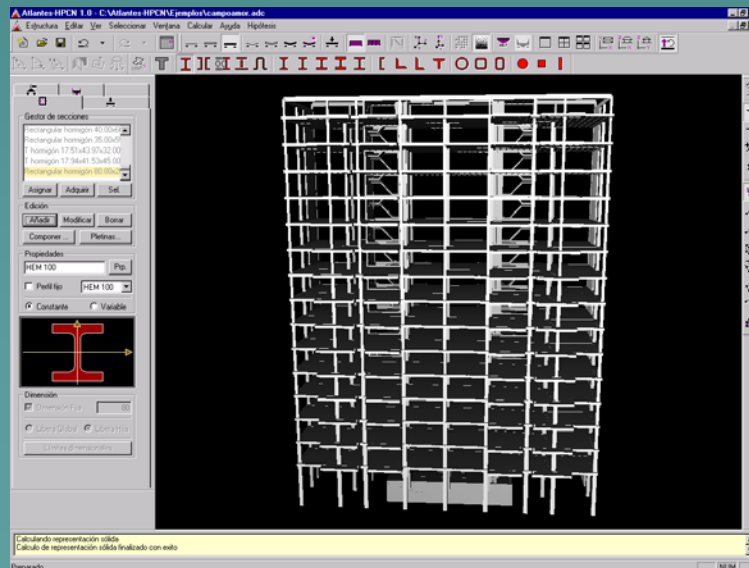
- ◆ Existen multitud de problemas en Ingeniería, Ciencia, Economía, ... que solo se pueden resolver usando ordenadores.
- ◆ Con frecuencia los problemas los plantean ingenieros y matemáticos, que formulan el problema en forma matemática. Pero (también con frecuencia) carecen de la formación informática necesaria para obtener soluciones de forma eficiente.

Objetivos del Ingeniero Informático especialista en Ciencias Computacionales

- ◆ Entender los modelos matemáticos de forma adecuada para ser tratados eficientemente por un computador
- ◆ Conocer herramientas y tecnologías (Librerías numéricas y matriciales de altas prestaciones) que permitan resolver los problemas computacionales planteados
- ◆ Ser capaz de resolver los problemas planteados utilizando las herramientas de altas prestaciones

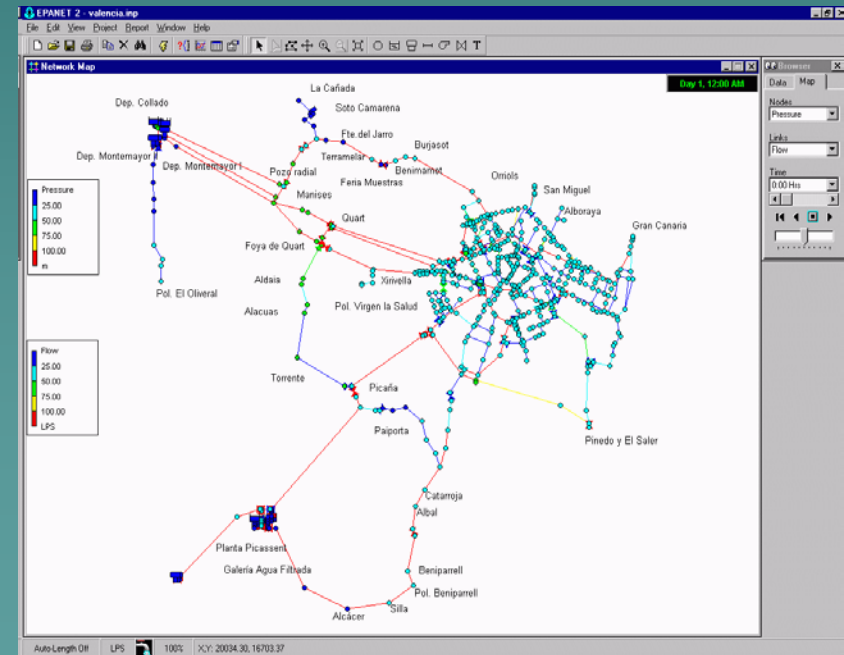
Ejemplo 1

- ◆ Cálculo de estructuras de edificación e ingeniería civil:
 - Resolución de sistemas de ecuaciones lineales y problemas de valores propios con cientos de miles de incógnitas. Se usan librerías para matrices dispersas.
 - La resolución precisa de estos sistemas permite reducir costes y aumentar la seguridad.



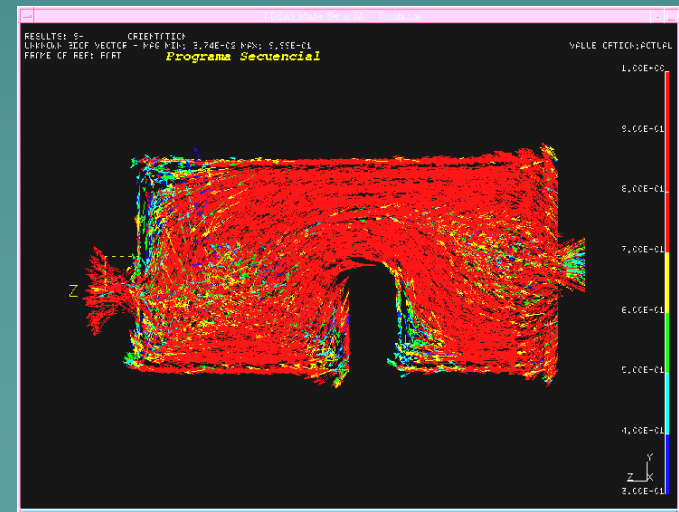
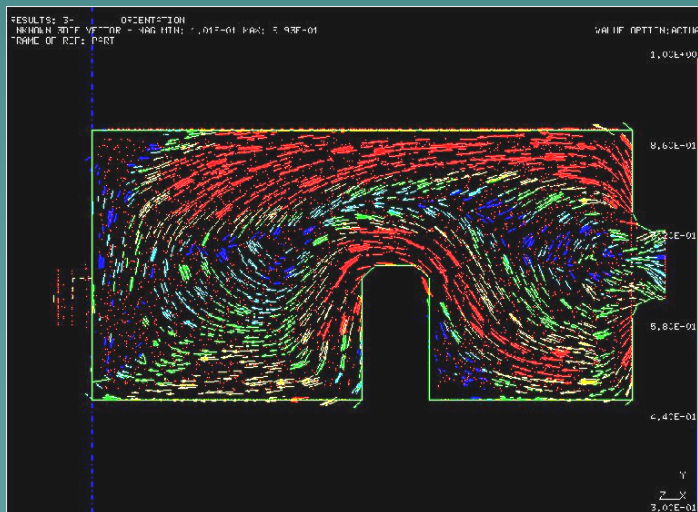
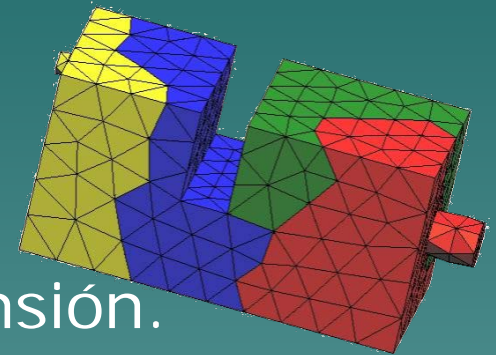
Ejemplo 2

- ◆ Redes de distribución de agua potable:
 - Simulación hidráulica: resolución de sistemas de ecuaciones no lineales.
 - Calidad del agua: formulación matemática.
 - Minimización de costes energéticos y de fugas de agua: optimización.



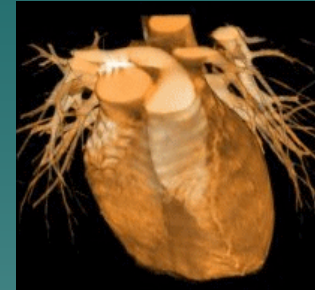
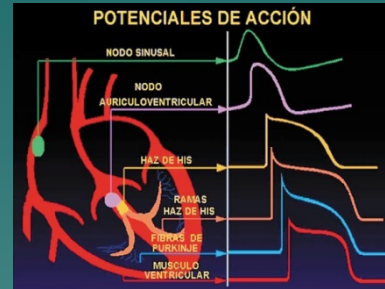
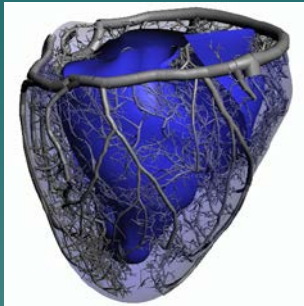
Ejemplo 3

- ◆ Simulación de inyección de plásticos:
 - La simulación de materiales visco-elásticos requiere modelos muy complejos.
 - Resolución de sistemas de ecuaciones (matrices) de gran dimensión.



Ejemplo 4

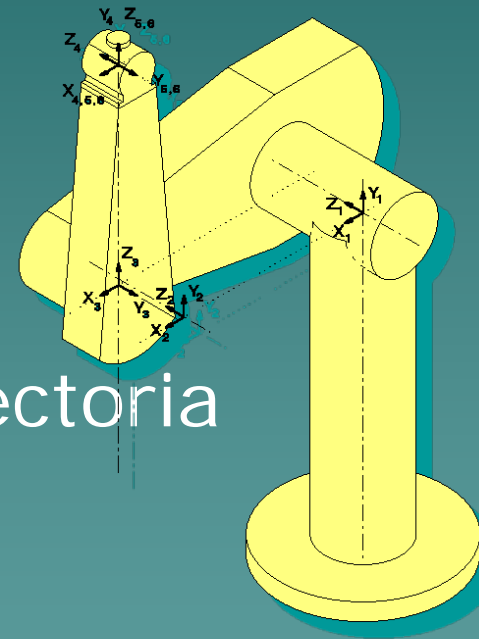
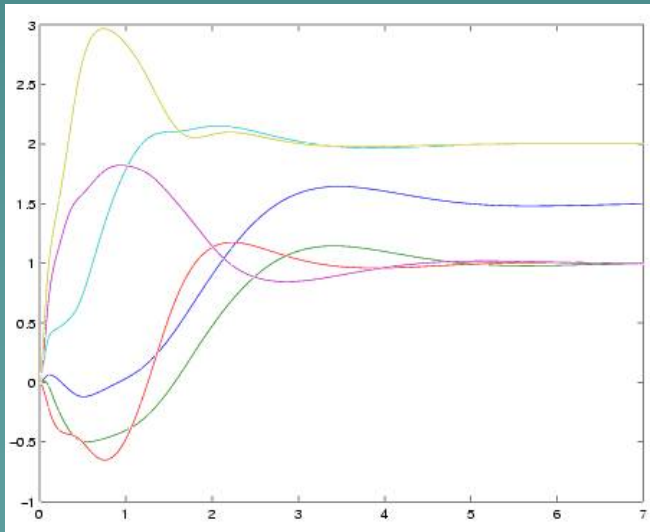
◆ Simulación de la actividad eléctrica cardiaca:



- Obtención del potencial de acción. Indicativo de patologías cardiacas. Desarrollo de nuevos fármacos.
- La simulación se divide en múltiples pasos de tiempo. Cada paso implica resolver un sistema de ecuaciones.
- Gran complejidad computacional: Una simulación de un tejido de 1 cm x 1 cm x 1 cm durante 250 milisegundos (con pasos de tiempo de 8 microsegundos) implica días de cómputo.
 - ◆ Resultados que ocupan 232 GBytes.

Ejemplo 5

- ◆ Control de Robots Manipuladores:
 - Resolver gran cantidad de problemas en un tiempo corto.
 - Es necesario resolver la ecuación dinámica del robot y calcular el control que se ajuste a una trayectoria determinada.



$$E \cdot \ddot{x}(t) = F(x, u, t)$$

Ejemplo 6

- ◆ Big Data: Sistemas de Recomendación
- ◆ Empresas como NetFlix utilizan Análisis de Componentes Principales para mejorar sus sistemas de recomendación. Esta técnica depende de descomposiciones matriciales como la SVD (descomposición en valores singulares)
- ◆ Estas descomposiciones de matrices enormes solo se pueden calcular usando técnicas de paralelismo y computación de altas prestaciones, y, en la actualidad, se suelen hacer mediante Cloud Computing.

Conclusiones

- ◆ Para poder colaborar como informático en problemas de estos tipos, se necesita:
 - Adquirir los conceptos básicos de computación de altas prestaciones.
 - Conocer los algoritmos matriciales básicos
 - Saber utilizar las librerías donde están implementados los algoritmos matriciales.