

Desarrollo, Optimización y Autooptimización de Algoritmos Paralelos para Análisis Cinemático de Sistemas Multicuerpo Basado en Ecuaciones de Grupo

Tesis Doctoral

Autor:

José Carlos Cano Lorente

Directores:

Antonio Javier Cuenca Muñoz

Domingo Giménez Cánovas

Mariano Saura Sánchez

23 de Abril de 2021

Escuela Internacional de Doctorado
Universidad de Murcia



- **Carga computacional** elevada en la solución numérica de problemas científicos complejos. Ejemplo: estudio de sistemas multicuerpo.
- El uso de **paralelismo** permite distribuir cálculos entre varias unidades de computación → reducción de los tiempos de ejecución en plataformas hardware heterogéneas.
- **Diseño eficiente de las aplicaciones** de acuerdo a las especificaciones del hardware y ajuste de las librerías de cómputo → tarea compleja para un usuario no experto en computación.

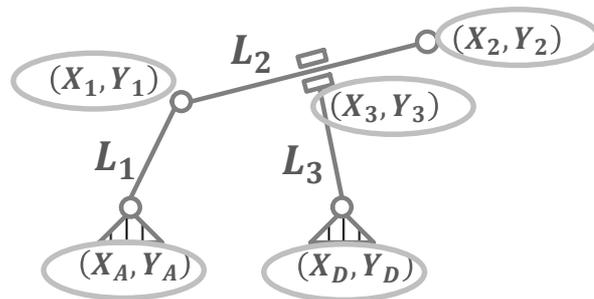
El trabajo realizado para adaptar un código a una plataforma puede no ser válido en otra.

- Estudio de técnicas de **optimización** y autooptimización de los algoritmos de **resolución de sistemas multicuerpo**, mediante el uso de **paralelismo** y una adecuada selección de las librerías de cómputo.
- Implementación de dichas técnicas en un **software** que ayude a un usuario no experto en computación a la correcta configuración del código en un determinado hardware.
- **Extender la metodología** a otros tipos de problemas computacionales que impliquen la resolución simultánea de un conjunto de subproblemas de álgebra lineal.

- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

- 1 **Introducción**
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

- El **modelado** de mecanismos obtiene el conjunto de variables y ecuaciones que representan los elementos que los componen y los movimientos relativos entre ellos.



$$(X_3 - X_D) + (Y_3 - Y_D) - L_3^2 = 0$$

$$(X_2 - X_1)(X_3 - X_D) + (Y_2 - Y_1)(Y_3 - Y_D) = 0$$

- Aplicando métodos numéricos se puede obtener una resolución computacional del estudio de la posición y el movimiento del sistema (análisis cinemático) → alternativa a la costosa creación de prototipos físicos.
- Interés en optimizar los algoritmos de resolución.

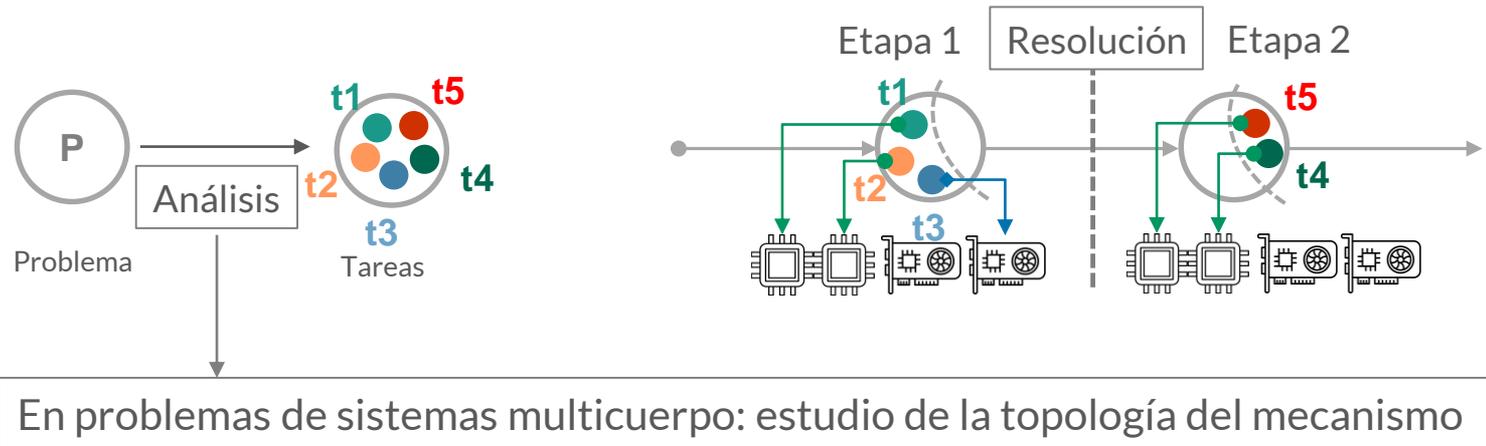
Introducción

1 Modelado

2 Programación paralela

3 Librerías

- Uso de múltiples recursos computacionales para resolver un problema.
- Aplicable a problemas divididos en tareas cuyas instrucciones puedan ejecutarse de manera simultánea en diferentes elementos de proceso.



- Usaremos OpenMP para explotar el paralelismo en sistemas de memoria compartida, combinado con asignaciones de cálculos a una o varias GPU.

- Álgebra lineal: Uso habitual en la solución de problemas científicos, en particular en el análisis de mecanismos.
- Librerías: Contienen subrutinas para operaciones comunes de alto nivel, como la descomposición LU y resolución de sistemas de ecuaciones.
- Implementaciones optimizadas para el manejo de matrices de diferentes tamaños, densas o dispersas, y con adaptaciones a sistemas de memoria compartida, distribuida o empleando GPUs.

- 1 Introducción
- 2 Sistemas Multicuerpo**
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

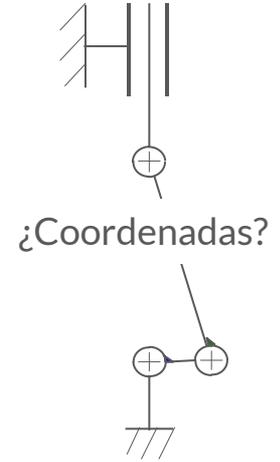
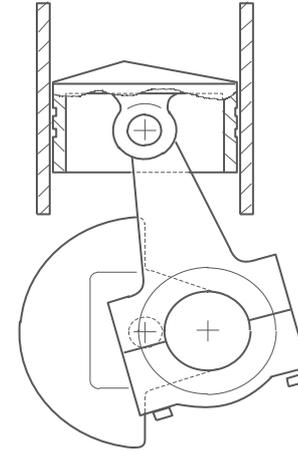
Sistemas Multicuerpo

1 Conceptos

2 Análisis estructural

3 Cinemática computacional

- Sistema multicuerpo: modelado de un mecanismo como un conjunto de sólidos rígidos o flexibles conectados entre sí por un conjunto de uniones.
- Modelado en coordenadas **independientes**: se emplean tantas coordenadas como grados de libertad tiene el sistema.
- Modelado en coordenadas **dependientes**: utiliza un conjunto de coordenadas redundantes, cuyo número es superior al de grados de libertad del sistema, ligadas entre sí por una serie de ecuaciones de restricción → el más habitual.

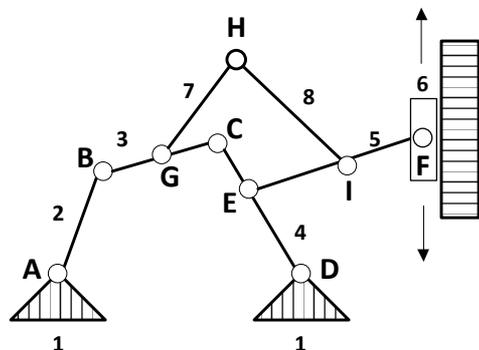


Sistemas Multicuerpo

1 Conceptos

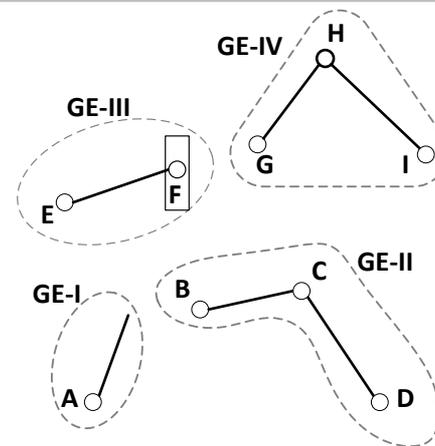
2 Análisis estructural

3 Cinemática computacional



Análisis estructural

Rama de la Teoría de Mecanismos que estudia la estructura cinemática de un mecanismo, es decir, **identifica los grupos o subsistemas** en los que se puede dividir, y el orden en el que se deben analizar.



Formulación GLOBAL

Se requieren todas las coordenadas que definen las posiciones de cada cuerpo con independencia del resto, y las ecuaciones de restricción asociadas a cada tipo de unión

No explotan la topología del mecanismo

Muchas variables → **matrices grandes**

Formulación TOPOLÓGICA

Basada en una descomposición de sistemas complejos en problemas más pequeños

Menor número de coordenadas
→ **matrices más pequeñas**

Implementación computacional modular
→ Permite el uso de **técnicas de paralelismo**

Sistemas Multicuerpo

1 Conceptos

2 Análisis estructural

3 Cinemática computacional

Identificar en cada instante t la posición y orientación de los sólidos del sistema

Datos $q = q_i \cup q_d$
 Φ : ecuaciones de restricción

Rutina **general** para el análisis cinemático computacional

1 Análisis de posición

Conocidas las posiciones q_i , se obtienen los valores de q_d resolviendo $\Phi(q, t) = 0$ iterativamente

2 Análisis de velocidades

Conocidas las velocidades \dot{q}_i y las posiciones q obtenidas en ①, se obtienen los valores de \dot{q}_d resolviendo $\dot{\Phi}(q, t) = 0$

3 Análisis de aceleraciones

Conocidas las aceleraciones \ddot{q}_i se obtienen los valores de \ddot{q}_d resolviendo $\ddot{\Phi}(q, t) = 0$

Análisis de un **grupo** estructural

- Se puede resolver usando el mismo esquema de la rutina general.
- Ventajas:
 - ✓ En ciertos casos de grupos sencillos, los sistemas de ecuaciones admiten soluciones directas, sin métodos iterativos.
 - ✓ Se pueden desarrollar y reutilizar librerías para resolver grupos con una topología concreta.

- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales**
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

Herramientas computacionales

1 Software

2 Hardware



Compiladores

JAVA: Interfaz gráfico.

C: Software de gestión de la simulación + OpenMP.

FORTRAN: Cálculos matriciales.

Interfaz FORTRAN \leftrightarrow C



Librerías de cómputo

MKL: Implementación eficiente de rutinas BLAS y LAPACK. **Densas**

MAGMA: rutinas LAPACK para arquitecturas heterogéneas multi CPU + multiGPU. **Densas**

PARDISO, HSL MA27, HSL MA57, HSL MA48, HSL MA86:
Solución de sistemas de ecuaciones dispersos. **Dispersas**

Códigos compatibles con Windows y Linux

Herramientas computacionales

1 Software

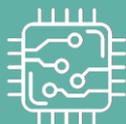
2 Hardware

Ordenadores personales

- CPUs multicore

Clúster Heterosolar

- Cinco nodos de cómputo
- CPUs multicore
- GPUs de distinto tipo



- Desarrollo del software.
- Experimentos en sistemas multiCore en entornos Windows y Linux.

Nodo	Marte	Mercurio	Saturno	Jupiter	Venus	
CPU						
Procesadores	Phenom II X6 1075T	Phenom II X6 1075T	Xeon E7530	Xeon E5-2620	Xeon E5-2620	
Fabricante/Arquitect.	AMD/x86-64	AMD/x86-64	Intel/x86-64	Intel/x86-64	Intel/x86-64	
Reloj	800 MHz	800 MHz	1.87 GHz	2.00 GHz	2.40 GHz	
Cores por CPU	6	6	6	6	6	
CPU por nodo	1	1	4	2	2	
Cores físicos/lógicos	6 / 6	6 / 6	24 / 48	12 / 24	12 / 12	
Memoria RAM	16 GB	16 GB	32 GB	32 GB	64 GB	
GPU						
Procesador NVIDIA	GeForce GTX 480	GeForce GTX 480	Tesla K20c	Tesla C2075	GeForce GTX 590	GeForce GT 640
Arquitectura	Fermi	Fermi	Kepler	Fermi	Fermi	Kepler
GPU por nodo	1	1	1	2	4	1
Reloj	1401 MHz	1401 MHz	706 MHz	574 MHz	608 MHz	902 MHz
Memoria	1536 MB	1536 MB	4800 MB	5375 MB	1536 MB	1024 MB
Streaming MultiProcs.	15	15	13	14	16	2
Streaming Processors	32	32	192	32	32	192
CUDA cores	480	480	2496	448	512	384

- Experimentos multiCore + multiGPU en SATURNO y JUPITER.
- Entorno Linux.

- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización**
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

Optimización y Autooptimización

1 Conceptos

2 Códigos de simulación de sistemas multicuerpo

Objetivo

Obtener el rendimiento óptimo de una aplicación determinada.

Complejidad

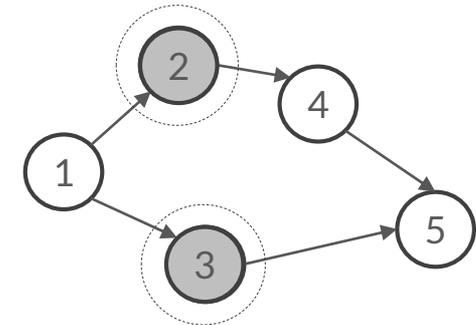
Dependencia de:

- El tamaño del problema.
- Las librerías de cómputo → parámetros ajustables.
- Arquitecturas hardware dispares → diferentes modelos de programación.

Técnicas

- Asignar cálculos a unidades de cómputo para un mejor aprovechamiento del hardware.
 - ↳ Selección de los mejores parámetros de ejecución. En paralelismo: asignaciones de hilos.
- Análisis de rendimientos: captura de tiempos de ejecución.
- Consulta de datos históricos: configurar ejecuciones futuras.

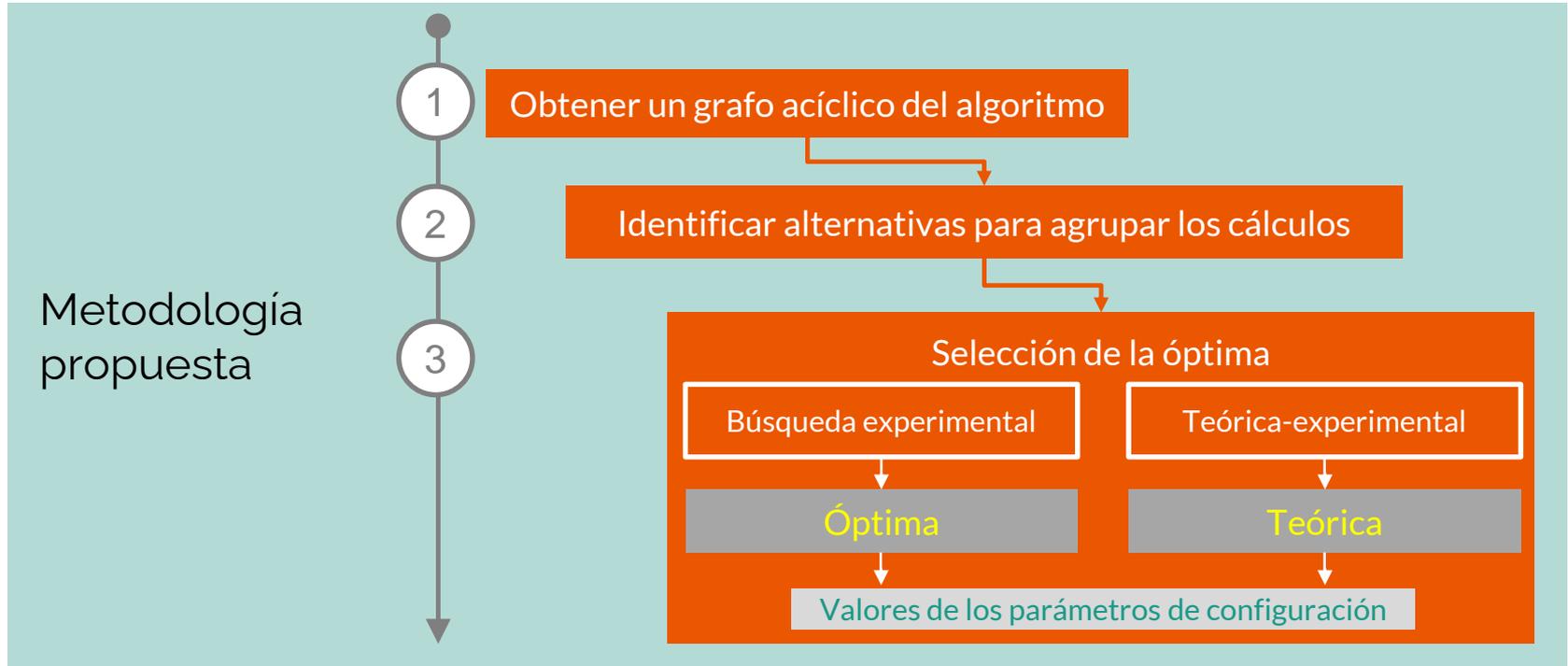
- Agrupar y asignar cálculos a unidades de cómputo → compatible con formulaciones topológicas de análisis cinemático, y en general con problemas planteados como resolución de subproblemas.
- Se requieren herramientas para analizar los algoritmos: **grafo dirigido**.
 - Permite representar la división de un sistema complejo en un conjunto ordenado de determinados subsistemas.
 - Los nodos corresponden a bloques de instrucciones que resuelven un subsistema.
 - Las líneas indican el orden de ejecución de los nodos.
 - Un estudio del grafo permite identificar subsistemas que se pueden resolver en paralelo (ejemplo, nodos 2 y 3).



Optimización y Autooptimización

1 Conceptos

2 Metodología



Aplicable a cualquier tipo de problemas computacionales que impliquen la resolución simultánea de un conjunto de subproblemas → **Generalización.**

- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador**
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros

Simulador

1 Modelos

2 Árbol de rutas

3 Simulación

Modelo

Representación en el simulador del conjunto de cálculos necesarios para resolver un determinado problema → definen un **algoritmo**.

Grupo

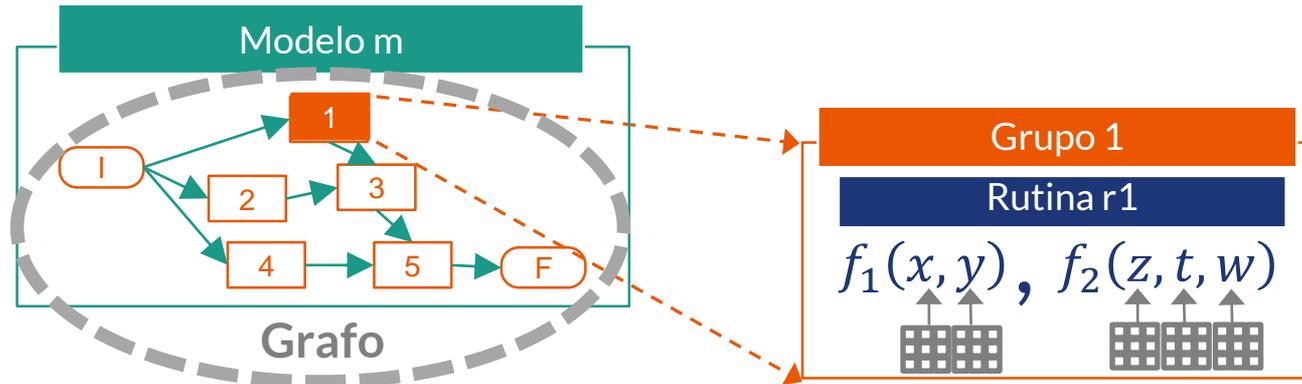
Entidad que representa cada uno de los subproblemas que conforman el problema a resolver.

Rutina

Conjunto de operaciones matriciales y funciones que se ejecutan en secuencia dentro de un grupo.

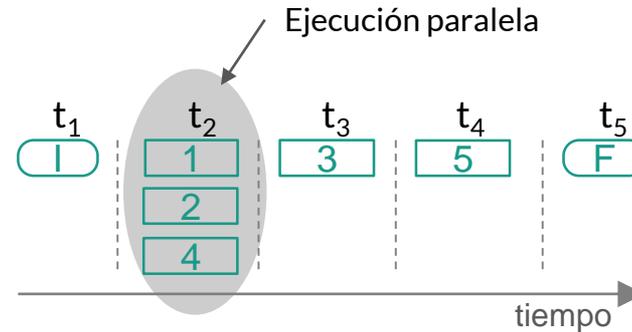
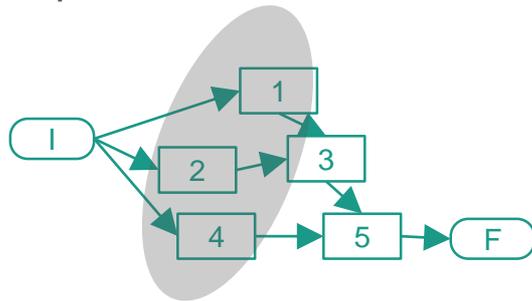
Variables

Matrices necesarias para la resolución numérica del problema → Argumentos de las funciones.

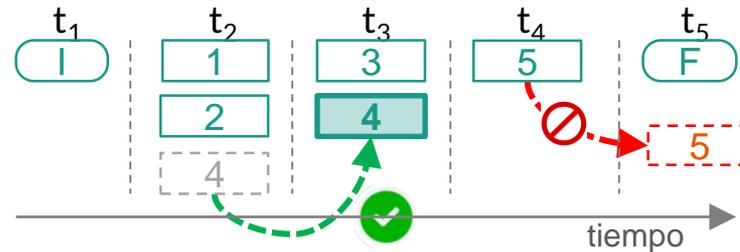


- Ruta: Cada una de las posibles ordenaciones válidas de los grupos de un modelo.

Ejemplo:

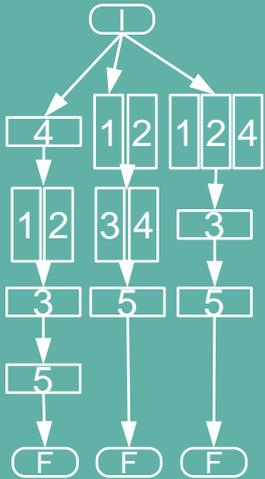


- Las rutas respetan el orden de precedencia que indica el grafo.



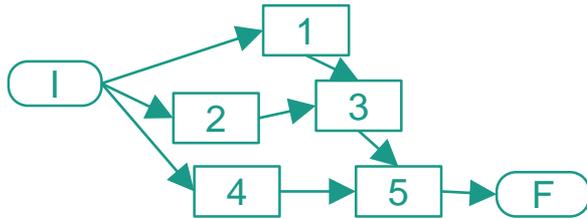
- El conjunto de rutas válidas se puede representar en forma de un **árbol de rutas**.

Propiedades del árbol de rutas:

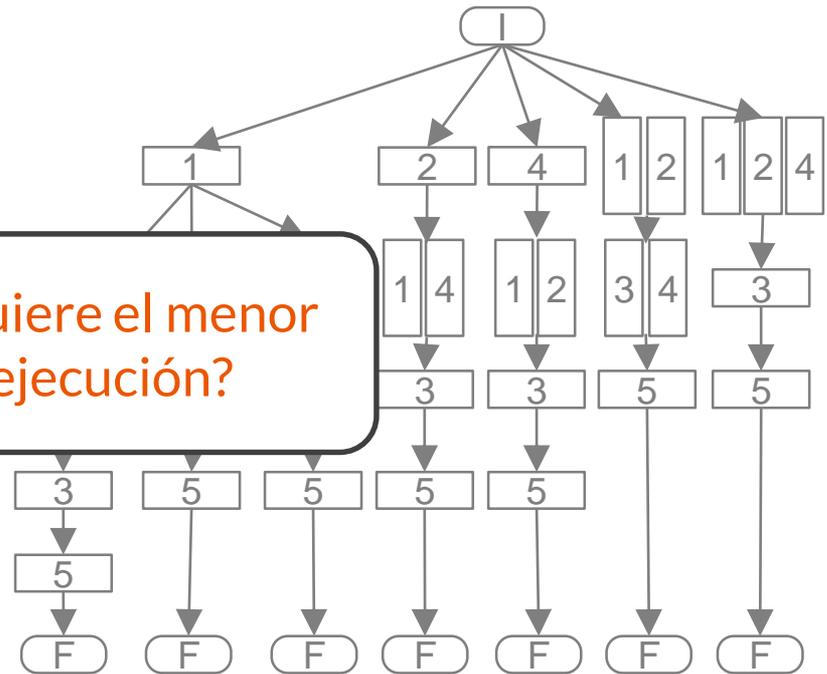


- Cada rama (ruta) está compuesta por nodos que contienen uno o más grupos.
- Una rama representa la solución del modelo completo → todos los grupos deben estar incluidos en todas las ramas.
- No se repiten rutas con el mismo coste computacional → solo una rama por cada agrupación de rutinas, descartando permutaciones.

Grafo



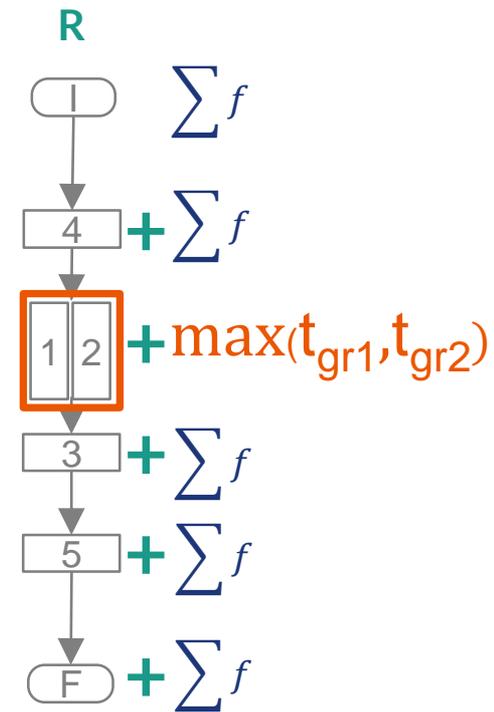
Árbol de rutas

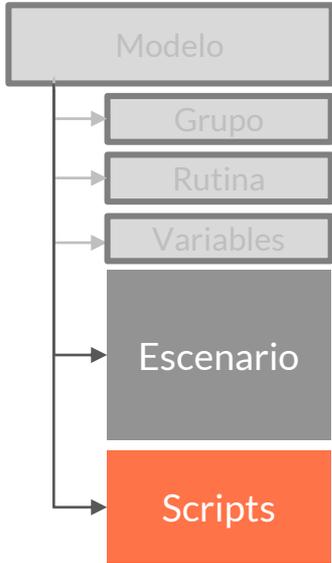


¿Qué ruta requiere el menor tiempo de ejecución?

Cálculo del tiempo de ejecución de una ruta R

- Tiempo de ejecución de una **ruta**: sumatorio de los tiempos de sus **nodos**.
- Los grupos contenidos en un **nodo** se calculan simultáneamente:
 - Tiempo de ejecución de un **nodo**: Mayor de los tiempos de los **grupos** que lo forman.
- Las funciones en un grupo se ejecutan en secuencia:
 - Tiempo de ejecución de un **grupo**: Sumatorio de los tiempos de ejecución de las **funciones**.





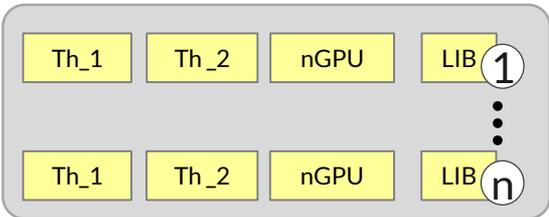
Simulación: ejecución de los cálculos especificados en un **modelo** siguiendo una o varias **rutas**.

Definen el **tamaño** del problema (dimensión de matrices, tipología y factor de dispersión) → Tipo de datos de las variables usadas como argumentos de las funciones a la hora de ejecutar una rutina.

Define los valores de los parámetros que son ajustables (**parámetros algorítmicos**) que se aplicarán durante la simulación.

Lista o rango de valores

- **Th_1** threads paralelismo de grupos
- **Th_2** threads paralelismo de librerías
- **nGPU** número de GPU a usar
- **LIB** identificador de la librería



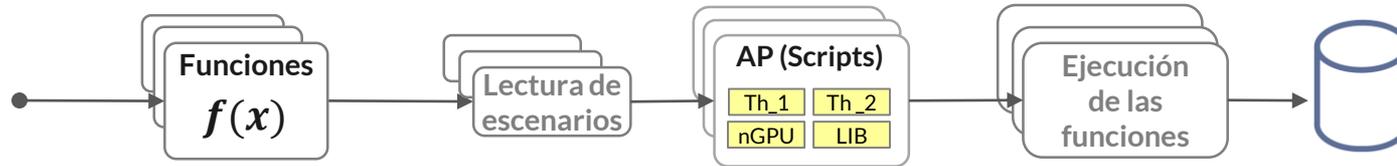
Simulador

1 Modelos

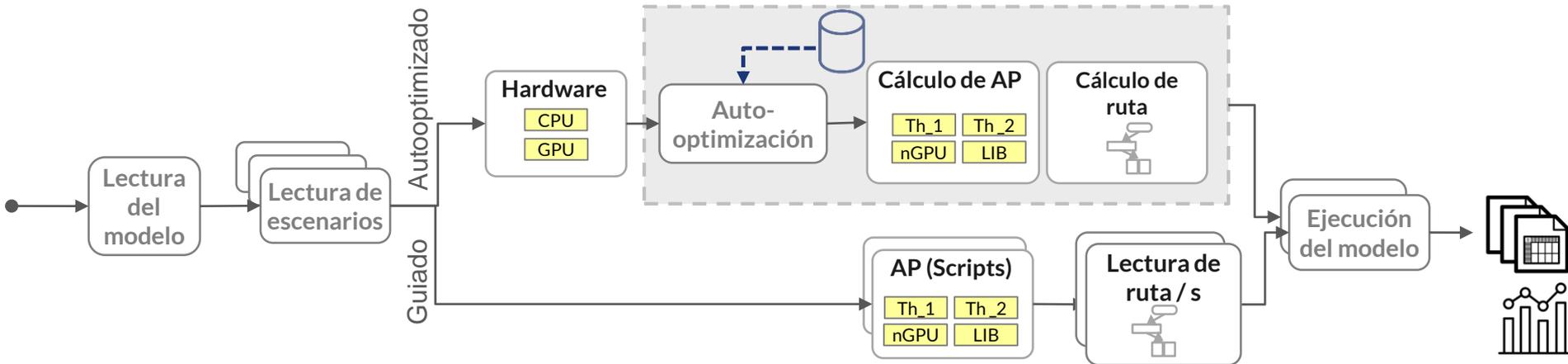
2 Árbol de rutas

3 Simulación

Modo entrenamiento: Ejecuta las funciones usando diferentes tipos de matrices (**escenarios de entrenamiento**) y valores de los parámetros algorítmicos (**AP de entrenamiento**).



Modo simulación: Proceso que realiza los cálculos especificados en un **modelo** y **escenario**, aplicando diferentes valores de los parámetros algorítmicos (**AP**), siguiendo una o varias **Rutas**.



- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos**
- 7 Conclusiones y Trabajos Futuros

- Mostrar su utilidad para:
 - Decidir el **orden de los cálculos** (ruta) y sus parámetros algorítmicos, tanto en modo guiado como en ejecución autooptimizada.
 - Estudiar el comportamiento de **rutinas equivalentes** para resolver problemas con diferentes plataformas y parámetros ajustables.
 - Usar la información obtenida para construir **tablas de decisión** que pueden ser enlazadas en rutinas que incluyan opciones de **autooptimización**.

Experimentos

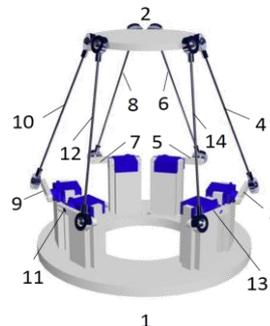
1 Objetivos

2 MBS

3 Multiplicación de matrices

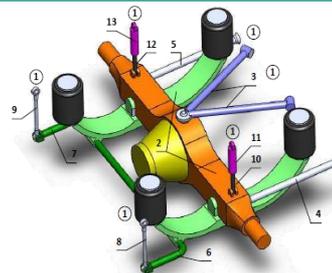
Plataforma de Stewart

Robot manipulador paralelo donde la posición de un terminal se controla mediante seis actuadores independientes que tienen la misma topología, y se pueden resolver simultáneamente.

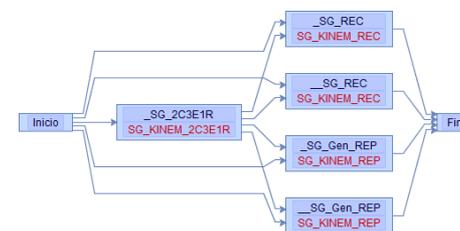
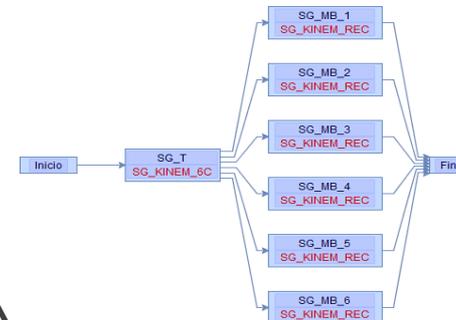


Sistema de suspensión de un eje de un camión

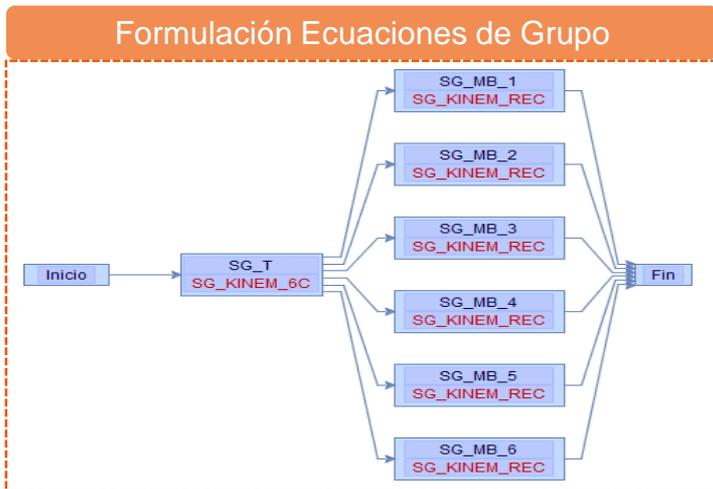
Trece cuerpos y diferentes articulaciones.
Escalable: permite incluir un número creciente de ejes en el modelo.



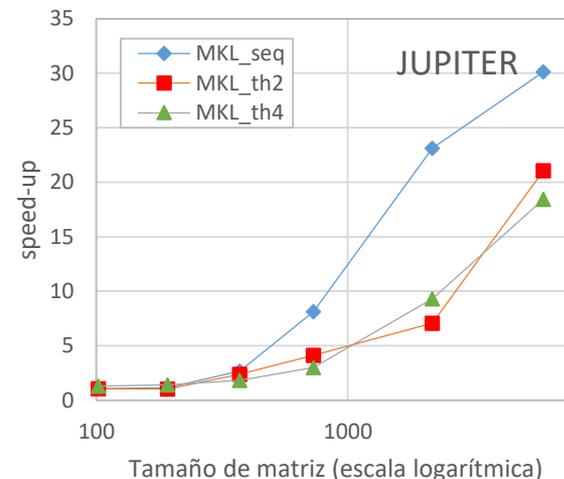
Análisis estructural



Utilidad: Análisis de diferentes formulaciones (Ecuaciones de Grupo vs Global).



nEQ_T	nEQ_{MB}	nEQ_{Global}
12	15	102
12	30	192
12	60	372
12	120	732
12	300	2172
12	1000	6012
12	2000	12015
12	3000	18015



$$nEQ_{Global} = nEQ_T + 6 * nEQ_{MB}$$

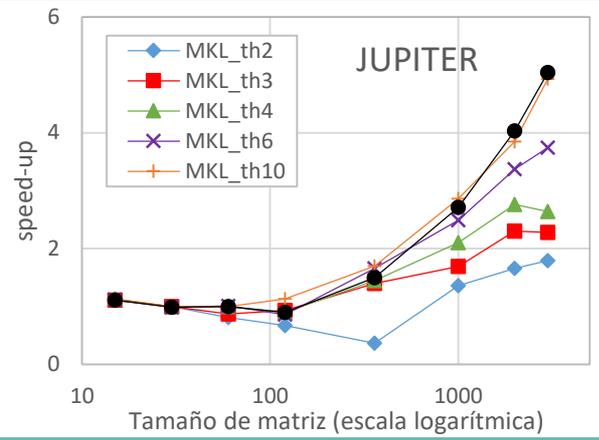
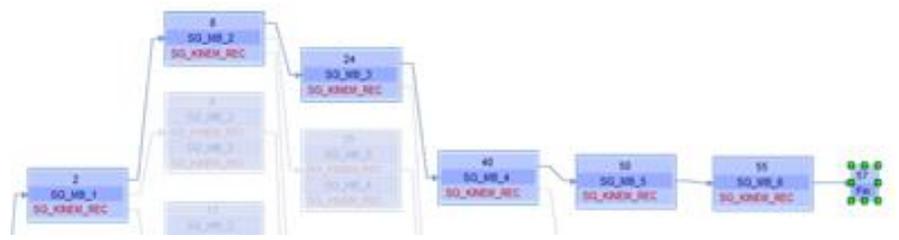
- Conclusiones: Ecuaciones de grupo ofrece mejores tiempos. Con MKL, muy optimizada para matrices grandes, el speed-up se reduce al aumentar el número de threads dado el mayor tamaño de matrices manejadas en la formulación global.

Experimentos

- 1 Objetivos
- 2 MBS
- 3 Multiplicación de matrices

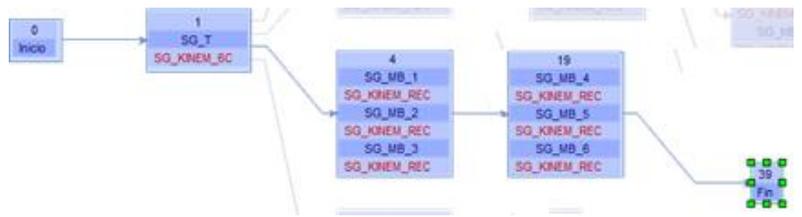
Utilidad: Análisis de diferentes rutas de cálculo.

- Ejecución **en secuencia** de los seis grupos en Stewart.



Variando el número de threads asignados a la librería MKL.

- Ejecución **simultánea** de tres grupos → Introducción de paralelismo de grupos.



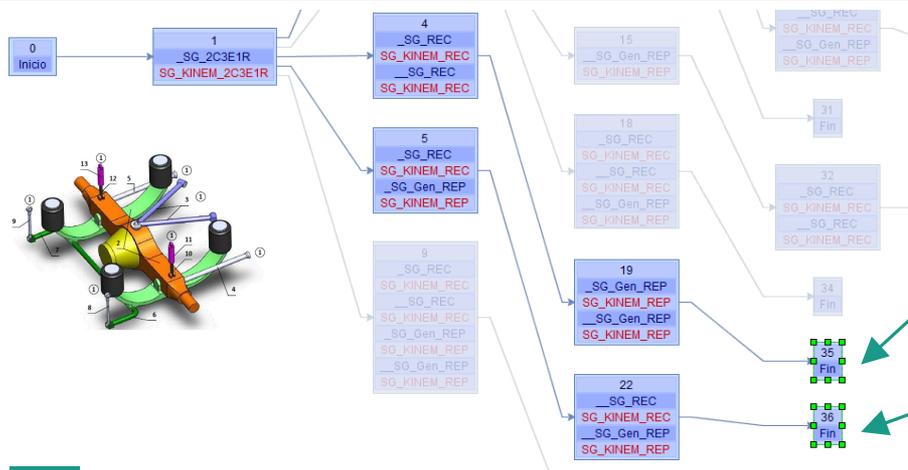
Variando el número de threads asignados a las librerías MKL y PARDISO.

nEQ_MB	MKL		th. OMP × th. MKL			
			3 × 1	3 × 2	3 × 3	3 × 4
15	1 × 4	0.02071	0.01315	0.01330	0.01356	0.01355
30	1 × 4	0.02670	0.01265	0.01297	0.01273	0.01339
60	1 × 6	0.03452	0.01511	0.01488	0.01561	0.01621
120	1 × 10	0.05967	0.02682	0.02395	0.03618	0.02820
360	1 × 10	0.23340	0.17734	0.17993	0.14890	0.13786
1000	1 × 10	2.08138	2.66616	1.74596	1.52307	1.81382
2000	1 × 12	10.37525	15.63358	10.02083	8.77579	7.86532
3000	1 × 12	26.61387	49.72425	30.33689	26.10797	23.40896

nEQ_MB	PARDISO		th. OMP × th. PARDISO			
			3 × 1	3 × 2	3 × 3	3 × 4
15	1 × 1	0.05032	0.03500	0.03541	0.03544	0.03807
30	1 × 1	0.06695	0.02965	0.03583	0.03684	0.04110
60	1 × 1	0.10100	0.04200	0.04726	0.05062	0.06806
120	1 × 6	0.14795	0.06431	0.06267	0.06736	0.06947
360	1 × 6	0.62256	0.46196	0.40550	0.38145	0.34323
1000	1 × 6	3.49814	2.70817	2.23300	2.15049	2.32183
2000	1 × 10	15.46708	12.36325	9.78210	10.11429	8.99634
3000	1 × 10	36.11017	27.40348	25.44521	21.73778	21.49892

Experimentos

- 1 Objetivos
- 2 MBS
- 3 Multiplicación de matrices



Utilidad: Estudiar paralelismo de grupos con diferentes rutinas en el sistema de un eje.

A Ruta de ejecuciones simultáneas de dos grupos con la misma rutina.

B Ruta de ejecuciones simultáneas de dos grupos con distinta rutina.

A

nEQ_MB	MKL		th. OMP × th. MKL					
			2 × 1	2 × 2	2 × 3	2 × 4	2 × 8	2 × 12
37	1 × 8	0.04973	0.03578	0.03219	0.03043	0.03099	0.03203	0.03833
60	1 × 8	0.06713	0.04331	0.04061	0.04067	0.03991	0.07803	0.12226
120	1 × 4	0.08719	0.10462	0.05796	0.05956	0.05535	0.06388	0.11576
360	1 × 8	0.34809	0.43377	0.37073	0.25038	0.24390	0.21624	0.34104
1000	1 × 16	3.06832	6.65778	4.06421	3.49995	2.76865	3.06329	2.58186
2000	1 × 24	18.10030	46.45289	32.70827	19.34889	15.80933	13.27327	11.64035
3000	1 × 24	49.33507	136.36582	80.03931	58.31936	47.11536	33.43908	31.74696

B

nEQ_MB	MKL		th. OMP × th. MKL					
			2 × 1	2 × 2	2 × 3	2 × 4	2 × 8	2 × 12
37	1 × 8	0.04973	0.03477	0.03329	0.03087	0.03102	0.03234	0.03923
60	1 × 8	0.06713	0.04729	0.04521	0.04519	0.04557	0.05499	0.05234
120	1 × 4	0.08719	0.09702	0.06574	0.06837	0.06852	0.07647	0.08516
360	1 × 8	0.34809	0.58851	0.44067	0.35714	0.35258	0.29864	0.31047
1000	1 × 16	3.06832	10.78664	6.43909	5.83582	4.82973	4.02271	4.05059
2000	1 × 24	18.10030	66.63464	41.59162	31.82747	26.66708	19.40707	18.44834
3000	1 × 24	49.33507	219.85583	131.32784	92.80238	85.23057	50.86643	48.17318

Conclusiones en este modelo concreto:

- Mejores tiempos al paralelizar grupos con la misma rutina.

!! Puede ser diferente en otros modelos !!
→ necesidad de autooptimización

Experimentos

- 1 Objetivos
- 2 MBS
- 3 Multiplicación de matrices

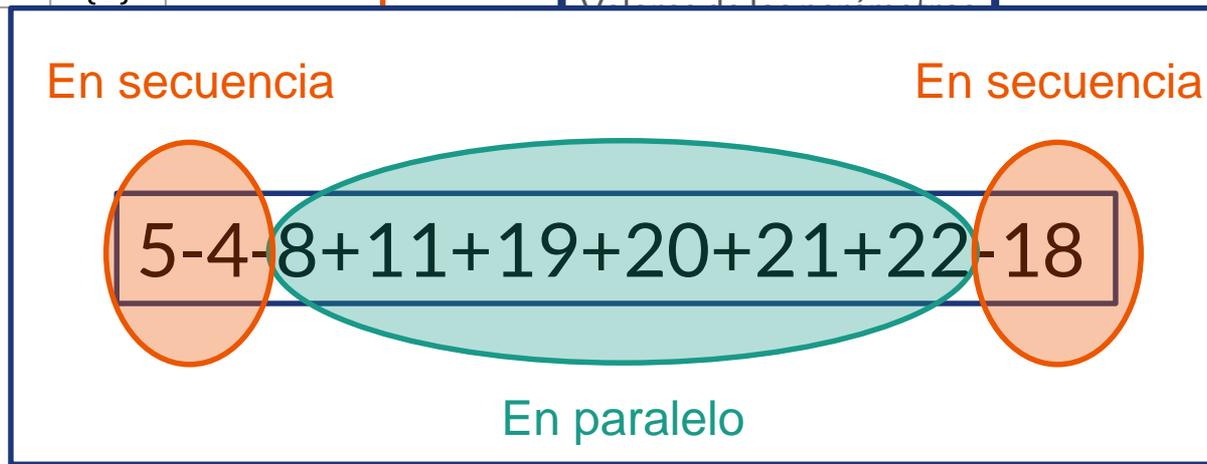
Utilidad: Ejecución autooptimizada de un modelo con un determinado escenario.

Especificaciones del hardware

Ejecución autooptimizada

Número de cores CPU	{24}
Número de GPUs	{1}

Ruta teóricamente más rápida.



ups sequence
4-8+11+19+20+21+22-18

3	SG_MB_2
3	SG_MB_3
3	SG_MB_4
3	SG_MB_5
3	SG_MB_6
4	Fin

pards	4	0

Autooptimizado
Óptimo

Experimentos						Autooptimizada
3 x 8		2 x 12		4 x 6		6 x 4
MKL	PARD	MKL	PARD	MKL	PARD	PARD
48.29932	34.66773	38.08665	43.07153	35.43077	30.14942	31.436998

Experimentos

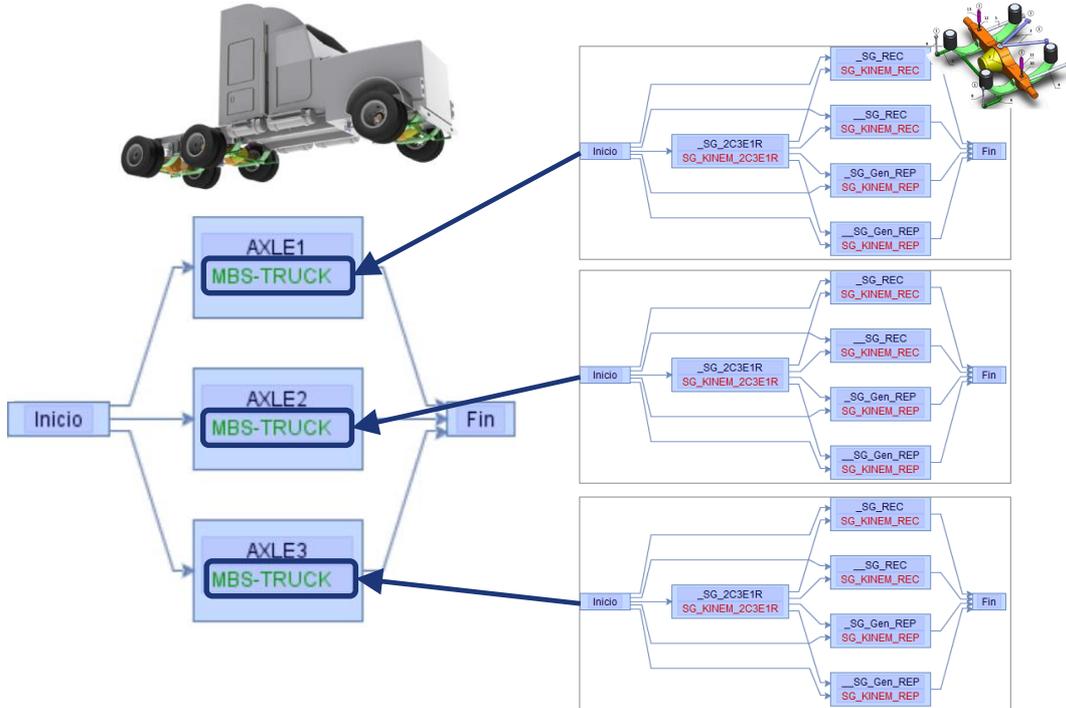
1 Objetivos

2 MBS

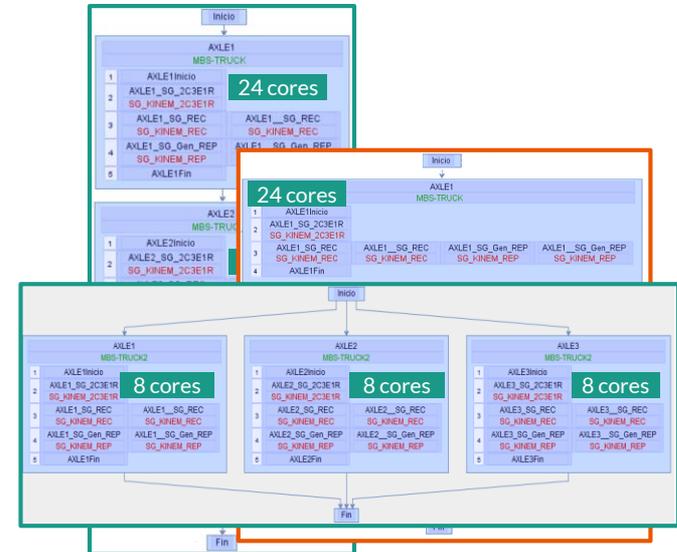
3 Multiplicación de matrices

Utilidad: Construir sistemas complejos anidando modelos estudiados previamente.

- Nueva problemática: Diferentes combinaciones de paralelismo, aplicables tanto en el modelo multieje como en los ejes que lo integran.



La información de optimización obtenida en el estudio del modelo de un eje se usa para optimizar el modelo de tres ejes

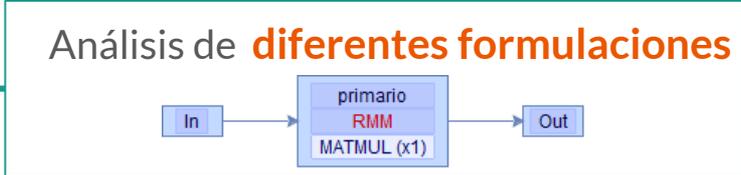


Experimentos

1 Objetivos

2 MBS

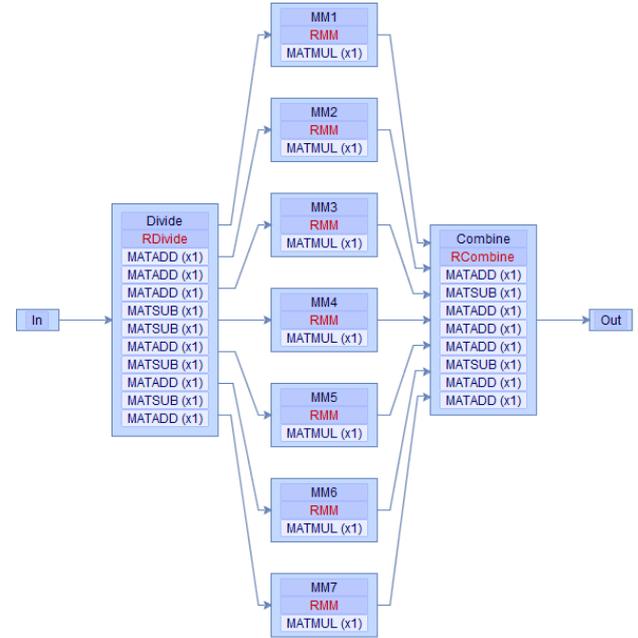
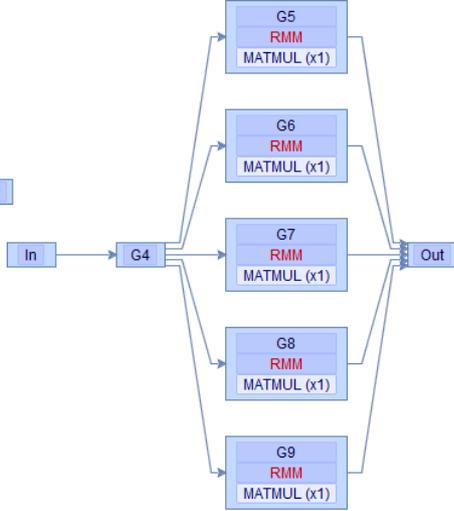
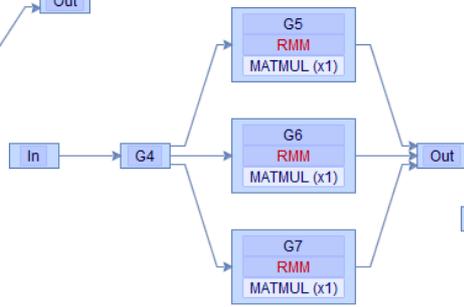
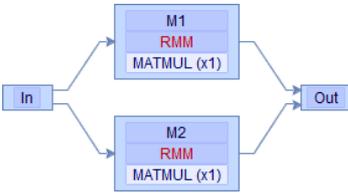
3 Multiplicación de matrices



Por bloques

Algoritmo de Strassen

Diferentes tamaños de bloque →

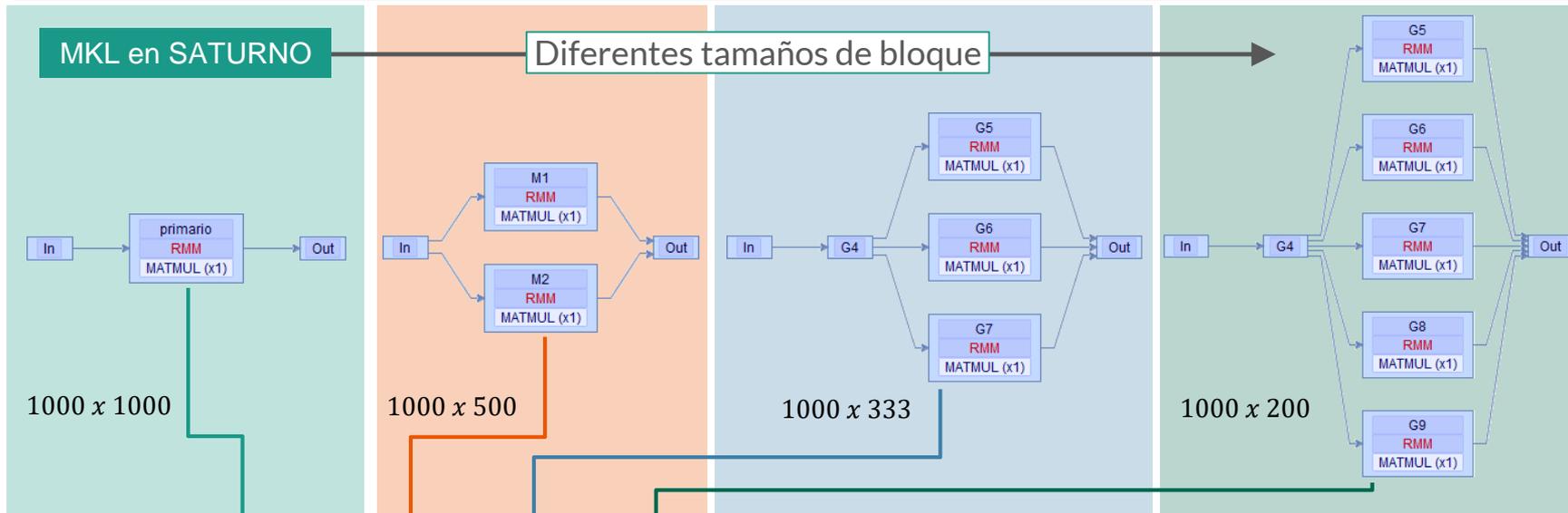


Experimentos

1 Objetivos

2 MBS

3 Multiplicación de matrices



nROWS	MATMUL		COLS50	COLS35	COLS20
	1x4	1x20	2x12	3x8	5x4
100	0.00043	0.00412	0.00065	0.00058	0.00072
500	0.01099	0.00643	0.01770	0.01010	0.00702
1000	0.12732	0.06158	0.03780	0.04037	0.03883
2000	0.60477	0.27485	0.27300	0.24414	0.28912
3000	1.96197	0.78933	0.74721	0.68834	0.57356

Diferentes estrategias paralelas analizadas:

- Matrices pequeñas → mejor MKL sin división.
- Matrices grandes → mejor más divisiones, con ejecuciones en paralelo y más threads asignados conforme aumenta el tamaño.

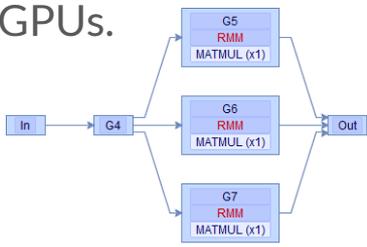
Experimentos

1 Objetivos

2 MBS

3 Multiplicación de matrices

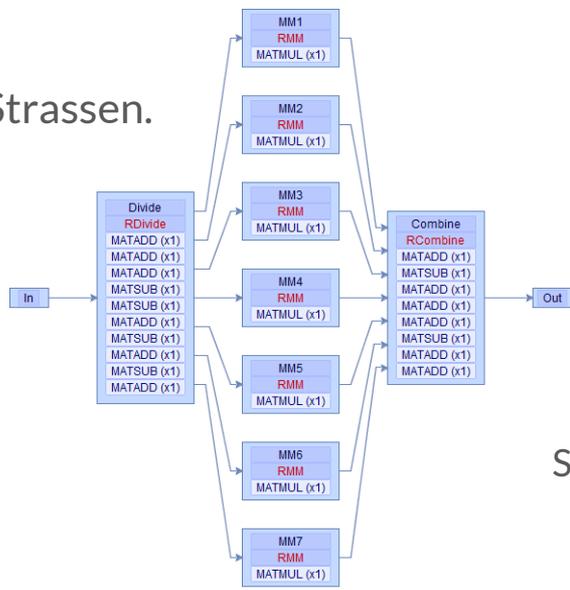
Rendimientos en GPUs.



nROWS	MKL				MAGMA
	3 x 1	3 x 2	3 x 3	3 x 4	3 GPUs
500	0.01019	0.00635	0.00350	0.00403	0.01055
1000	0.05926	0.02977	0.03592	0.02950	0.01796
2000	0.36205	0.16690	0.12079	0.13858	0.07952
3000	1.09032	0.86692	0.72478	0.58676	0.22992

Matrices grandes → mejor rendimiento empleando GPUs.

Algoritmo de Strassen.



nROWS x nCOLS	JUPITER		SATURNO	
	MATMUL	STRASSEN	MATMUL	STRASSEN
6000 x 6000	22.56067	21.20005	54.36038	50.13080
8000 x 8000	53.34177	47.80797	130.18680	114.81141
12000 x 12000	179.10999	160.26036	428.79211	379.44492

Strassen: 12% de mejor rendimiento en la plataforma SATURNO, cercano al 12.5% teórico

Experimentos

- No es el objetivo principal de la tesis optimizar modelos, son muchas comparativas las que se podrían llevar a cabo.
- Ejemplo:** Plataforma de Stewart, tiempo de ejecución en Windows con diversas librerías.
- Ejemplo:** Speedups para ejecución en Linux (JUPITER) y Windows (ORION 6 cores físicos, 12 lógicos).

nEQ_MB	Lib	80%							
		OMP2_sq	OMP2.th2	OMP2.th3	OMP2.th4	OMP2.th6	OMP2.th10	OMP2.th12	
15	MKL	0.036948	0.030273	0.037579	0.031866	0.044365	0.042407	0.045019	
	PARD	1.100252	0.138666	0.136854	0.081287	0.071711	0.121914	0.175256	
	MA27	0.047704	-	-	-	-	-	-	
	MA57	0.051514	-	-	-	-	-	-	
	MA48	0.045734	-	-	-	-	-	-	
	MA86	0.116287	0.113897	0.141051	0.221863	0.131635	0.175339	0.188972	
30	MKL	0.167111	0.047657	0.066623	0.039093	0.043609	0.070596	0.051768	
	PARD	0.085391	0.09304	0.084811	0.094401	0.097374	0.127513	0.202368	
	MA27	0.062337	-	-	-	-	-	-	
	MA57	0.069772	-	-	-	-	-	-	
	MA48	0.056202	-	-	-	-	-	-	
	MA86	0.115093	0.138579	0.177673	0.147586	0.175143	0.186452	0.197812	
60	MKL	0.192772	0.09599	0.164638	0.082488	0.072492	0.072052	0.086723	
	PARD	0.111931	0.118862	0.125578	0.133693	0.128583	0.160901	0.26254	
	MA27	0.193365	-	-	-	-	-	-	
	MA57	0.137059	-	-	-	-	-	-	
	MA48	0.130167	-	-	-	-	-	-	
	MA86	0.178939	0.213688	0.262391	0.305623	0.194055	0.187781	0.233921	0.270405
	0.241285	0.222191	0.285308	0.333992	-	-	-	-	
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	
	0.412753	0.564996	0.625475	0.63957	1.181641	1.296121	1.537612	1.620883	
	1.745631	1.665574	2.013542	2.168771	-	-	-	-	
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	
	3.308794	2.990206	3.866456	4.063697	9.524478	9.868568	11.034251	11.424513	
	11.746181	11.61982	13.005348	13.09405	-	-	-	-	
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	
	26.664694	26.969526	29.187355	29.656067	40.907253	40.580276	40.981945	42.401672	
	45.384052	45.657536	46.703342	48.313442	-	-	-	-	
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	
	117.241501	119.146042	118.919708	118.715355	-	-	-	-	

LINUX: Ejecución con MKL				WINDOWS: Varias librerías				
nEQ_MB	SpdUp: Global vs Global MKL (1 x 1)			nEQ_MB	SpdUp: Global vs Global MA27 (1 x 1)			
	MKL(1 x 2)	MKL(1 x 4)	MKL(1 x 6)		MA57(1 x 1)	MKL(1 x 1)	MKL(1 x 6)	MKL(1 x 12)
102	1.1193	0.9312	0.9312	102	1.75038	2.0701	1.94800	1.72271
192	1.1329	0.8219	0.8219	192	0.72670	2.00704	1.83569	1.03086
372	0.9072	1.4758	1.4758	372	0.97713	2.64072	2.05503	2.46644
732	1.3257	2.4485	2.4485	732	1.17475	2.71945	2.80219	2.21749
2172	1.2282	3.5975	3.5975	2172	1.61393	3.91812	4.60101	4.86324
6012	1.9424	3.4387	3.4387	6012	2.25572	5.82413	9.20582	9.57055

nEQ_MB	SpdUp: Grupos vs Global MKL (1 x 1)			nEQ_MB	SpdUp: Grupos vs Global MA27 (1 x 1)		
	GR(1 x 1)	GR(1 x th2)	GR(th1 x th2)		GR(1 x 1)	GR(1 x th2)	GR(th1 x th2)
102	1.0577	1.1956	1.8829	102	3.61256	4.40910	4.40911
192	1.1406	1.1427	2.4119	192	2.2182	3.53718	4.63619
372	2.6621	2.6767	6.2097	372	5.39285	8.32527	17.75133
732	8.1112	9.143	22.7791	732	11.03549	13.75045	49.23770
2172	23.1199	39.3067	66.5471	2172	28.1596	34.3617	97.81748
6012	30.1423	86.1958	117.7924	6012	13.63001	16.95825	227.02850

- 1 Introducción
- 2 Sistemas Multicuerpo
- 3 Herramientas Computacionales
- 4 Optimización y Autooptimización
- 5 Simulador
- 6 Experimentos
- 7 Conclusiones y Trabajos Futuros**

Conclusiones y Trabajos Futuros

1 Conclusiones

2 Difusión

3 Trabajos futuros

- Se ha presentado un **software** como asistente a científicos para crear **códigos optimizados**, con un interfaz visual para representar los algoritmos y generación automática de las rutas de resolución de los mismos.
- El software identifica los **parámetros de paralelismo ajustables** y **librerías** que mejor se adaptan a cada tamaño de problema planteado y hardware.
- Un simulador permite **verificar experimentalmente** la configuración propuesta a través de ejecuciones que la comparan con otras no optimizadas.
- Utilidad de dicho software en problemas de **análisis cinemático** y hemos generalizado a otro problema del **álgebra lineal**.
- **Herramienta didáctica** en la enseñanza de asignaturas de programación paralela (póster en el workshop EduPar-21).

Conclusiones y Trabajos Futuros

1 Conclusiones

2 Difusión

3 Trabajos futuros

- P. Segado, J. C. Cano, J. Cuenca, D. Giménez, M. Saura, P. Martínez. Eficiencia de la paralelización multihilo en el análisis cinemático de sistemas multicuerpo basado en Ecuaciones de Grupo. **XXI Congreso Nacional de Ingeniería Mecánica**, Elche, España, (2016).
- Gregorio Bernabé, José-Carlos Cano, Javier Cuenca, Antonio Flores, Domingo Giménez, Mariano Saura-Sánchez and Pablo Segado-Cabezas. Exploiting Hybrid Parallelism in the Kinematic Analysis of Multibody Systems Based on Group Equations. International Conference on Computational Science, **ICCS 2017**, 12-14 June 2017, Zurich, Switzerland.
- José-Carlos Cano, Javier Cuenca, Domingo Giménez, Mariano Saura-Sánchez and Pablo Segado-Cabezas. A Parallel Simulator for the Kinematic Analysis of Multibody Systems Based on Group Equations. Proceedings of the 18th International Conference on Computational and Mathematical Methods in Science and Engineering, **CMMSE**, July 9–14, 2018, Cádiz, Spain.
- José-Carlos Cano, Javier Cuenca, Domingo Giménez, Mariano Saura-Sánchez, and Pablo Segado-Cabezas. A parallel simulator for multibody systems based on group equations. **The Journal of Supercomputing**, 75, pp. 1368–1381, 2018.
- Jesús Cámara, José-Carlos Cano, Javier Cuenca, Domingo Giménez, Mariano Saura-Sánchez. Adapting a multibody system simulator to auto-tuning linear algebra routines (póster). **SIAM Conference on Parallel Processing for Scientific Computing (PP20)**, Seattle, Washington, U.S., 12-15 Febrero, 2020.

Conclusiones y Trabajos Futuros

1 Conclusiones

2 Difusión

3 Trabajos futuros

Funcionalidad general

- Gestión de **familias de GPUs** con diferentes prestaciones y memoria, incorporando en el software una heurística concreta de reparto de trabajo.
- Incorporar la asignación de cálculos a otro tipo de **coprocesadores** y el manejo de paralelismo basado en **paso de mensajes**.
- Añadir técnicas de **modelado de tiempos de ejecución** para reducir el tiempo de experimentación de funciones básicas y guiar la elaboración del árbol de rutas priorizando las configuraciones más prometedoras.
- Permitir a un usuario añadir **rutinas propias** y enlazar nuevas librerías.
- Generación de un **código fuente optimizado** que pueda ser reutilizado.
- **Integración con herramientas** que incluyan procesos de optimización.

Conclusiones y Trabajos Futuros

1 Conclusiones

2 Difusión

3 Trabajos futuros

Comunidad multibody

- Integrar el simulador con un **módulo de análisis estructural** computacional de sistemas multicuerpo.
- Definición de **estándares** en la creación de rutinas y de modelos de grupos estructurales de topologías determinadas para permitir su intercambio entre diferentes grupos de investigación.
- Para cada grupo estructural, poder **leer las matrices necesarias para el análisis** que se pretende llevar a cabo (cinemático, dinámico o de síntesis óptima)
- Integrar los resultados numéricos en una herramienta de **representación tridimensional**.

23 de Abril de 2021

Tesis:

Desarrollo, Optimización y Autooptimización de Algoritmos Paralelos para Análisis Cinemático de Sistemas Multicuerpo Basado en Ecuaciones de Grupo



Directores:

Antonio Javier Cuenca Muñoz
Domingo Giménez Cánovas
Mariano Saura Sánchez

Tribunal:

D. Vicente E. Vidal Gimeno (Presidente)
D. Juan Luis Aragón Alcaraz (Secretario)
D. Francisco J. González Varela (Vocal)