



J.L. ALVES

CT2M, UNIVERSIDADE DO MINHO
CEMUC, UNIVERSIDADE DE COIMBRA

MEI, 24 NOVEMBRO DE 2015

VAMOS POR AQUI

1. PROGRAMA **IMPLÍCITO**?
2. TÉCNICAS DE **HPC**?
3. HISTÓRIA DO DESENVOLVIMENTO DO PROGRAMA **DD3IMP**
 1. EM 2003 DISSEMOS QUE...
 2. E EM 2011...
4. O QUE FOI FEITO (DO PASSADO AO PRESENTE)
5. RESULTADOS
6. DESAFIOS (FUTURO)
7. O PMESIMULA (PARA ONDE QUEREMOS IR)

DD3IMP: DEEP-DRAWING 3D IMPLICIT FE SOLVER

Solver de Elementos Finitos

Formulação Elasto-Plástica

Grandes deformações e rotações

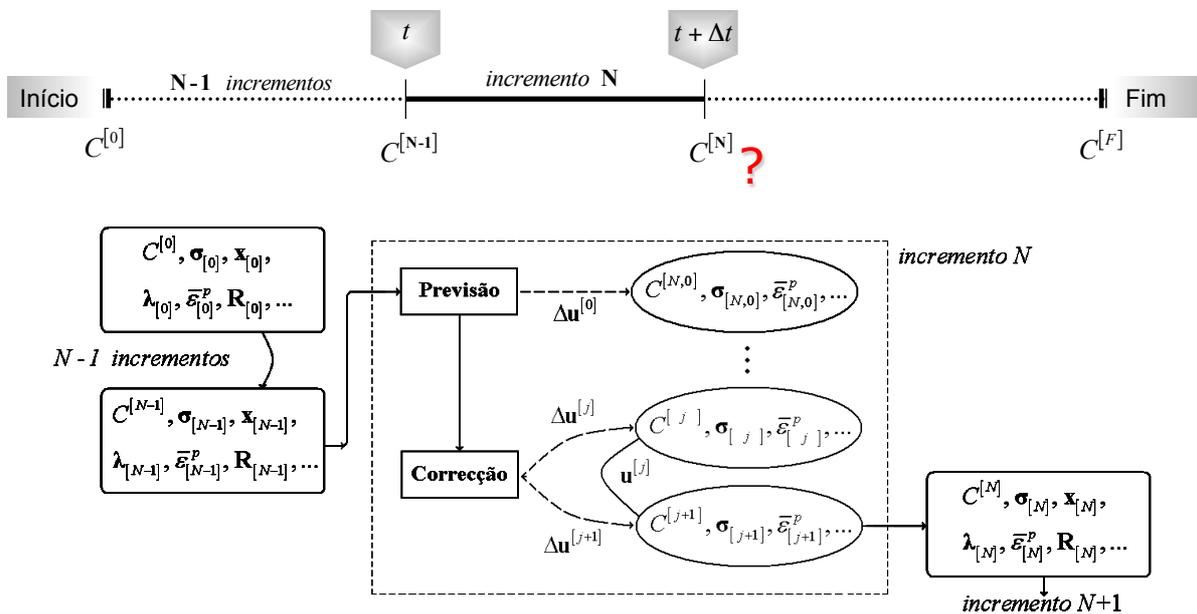
Enorme biblioteca de Modelos Constitutivos

Modelação de ferramentas e contacto com atrito

Esquema de integração temporal totalmente **IMPLÍCITO**

IMPLÍCITO? Quais os desafios? HPC?

3



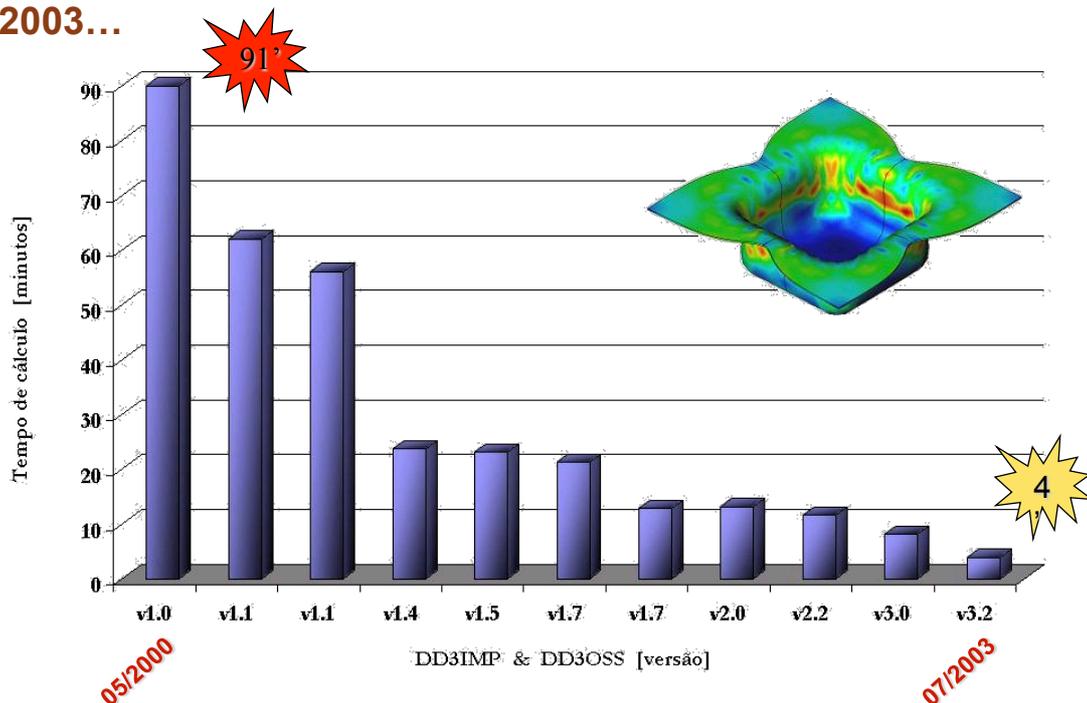
4

O QUE PODEREMOS FAZER/IMPLEMENTAR ?

- 1º, identificar os “gargalos” mais relevantes
- 2º, melhorar os algoritmos sequenciais
- 3º, adoptar bibliotecas externas optimizadas (se disponíveis)
- 4º, paralelizar o algoritmo principal (memória partilhada)
- 5º, decompor o domínio (candidatos?)

5

EM 2003...



6

EM 2011...

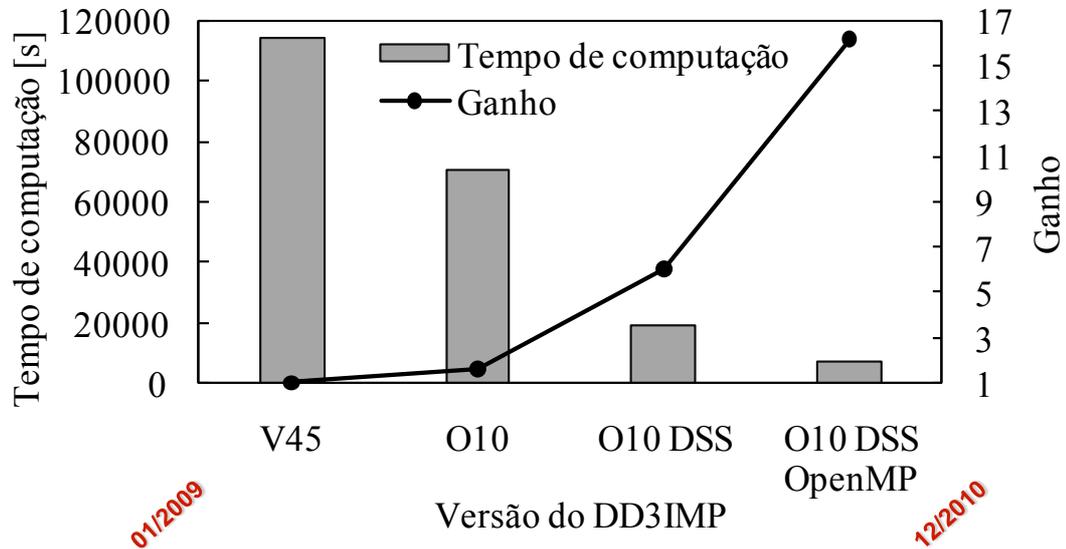


Figura 6. Tempo de computação com várias versões do DD3IMP para a discretização de $65 \times 65 \times 3$ elementos.

7

Resumo:

Alocação dinâmica de todas as variáveis

Identificação e Optimização dos **caminhos críticos**

Implementação de um novo **solver direto**

DSS – Direct Sparse Solver (Intel MKL)

Implementação de directivas **OpenMP** (sempre que possível)

8

Implementação de directivas **OpenMP**

Designação adoptada para identificar as regiões paralelizadas através de directivas OpenMP.			
Designação		Tarefa associada	Número de chamadas em cada incremento
1)	série	nenhuma	nenhuma
2)	rmin	Estratégia de r_{min}	1
3)	predic	Fase de <i>Previsão</i> : cálculo das matrizes elementares e vectores de segundo membro; montagem	1
4)	contact	Tratamento do contacto	$1+1+n^\circ$ de iterações de equilíbrio
5)	correc	Fase de <i>Correcção</i> : cálculo das matrizes elementares e vectores de segundo membro; montagem	n° de iterações de equilíbrio
6)	Todas	Últimas quatro (rmin+prev+contact+correc)	$4+2*n^\circ$ de iterações de equilíbrio

9

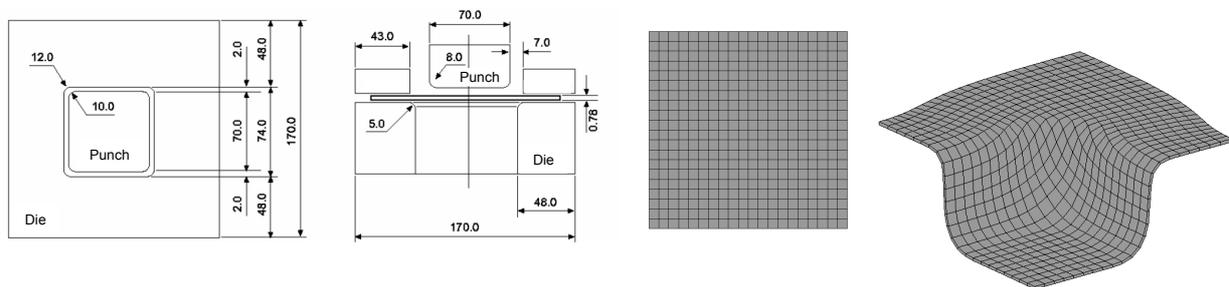
Implementação de directivas **OpenMP**

Código Original	Código Modificado
<pre> DO IE=1, NEDEF NEL=NNODE (IE) CALL DMLOCA (IE, NEL, XL, DUL, MC, NI, NPINT, ITYPE, iMAT) CALL DMELCO (iMAT, ITYPE, IE, NI, NPINT, NEL, XL, DUL, XNEDEF (1, IE), IEMES, IPMES, INDEX) END DO </pre>	<pre> \$OMP PARALLEL \$OMP& PRIVATE (NEL, MC, NI, NPINT, ITYPE, iMAT) \$OMP& PRIVATE (XL, DUL) \$OMP DO DO IE=1, NEDEF NEL=NNODE (IE) CALL DMLOCA (IE, NEL, XL, DUL, MC, NI, NPINT, ITYPE, iMAT) CALL DMELCO (iMAT, ITYPE, IE, NI, NPINT, NEL, XL, DUL, XNEDEF (1, IE), IEMES, IPMES, INDEX) END DO \$OMP END DO \$OMP END PARALLEL </pre>
(a)	(b)

Figura 1. Exemplo de aplicação de directivas *OpenMP* ao código do DD3IMP
 (a) código original; (b) código modificado com directivas *OpenMP*.

Benchmark: estampagem de uma taça quadrada

	Discretização no plano da chapa ($N \times N$)							
	33×33	45×45	55×55	65×65	75×75	85×85	95×95	110×110
$L=2$	10404	19044	28224	39204	51984	66564	82944	110889
$L=3$	13872	25392	37632	52272	69312	88752	110592	-



11

Benchmark: estampagem de uma taça quadrada

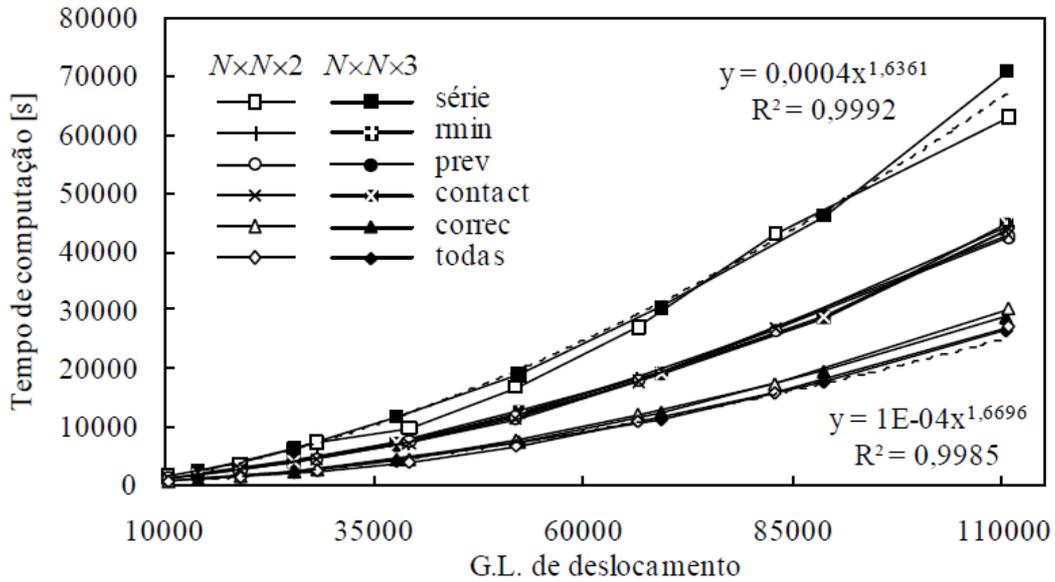
- As simulações foram realizadas em dois computadores:
 - Intel® Core™ i7-950 (3.07 GHz) CPU
 - Windows7 Professional (64-bits platform) OS
- Todas as versões do DD3IMP foram compiladas com a opção (/fast).
- Performance avaliada com base na taxa de aceleração (*speedup*):

$$S_p = T_1/T_p$$

- Cada simulação é repetida 3 vezes nas mesmas condições, de modo a determinar o valor médio dos tempos de computação bem como a dispersão.

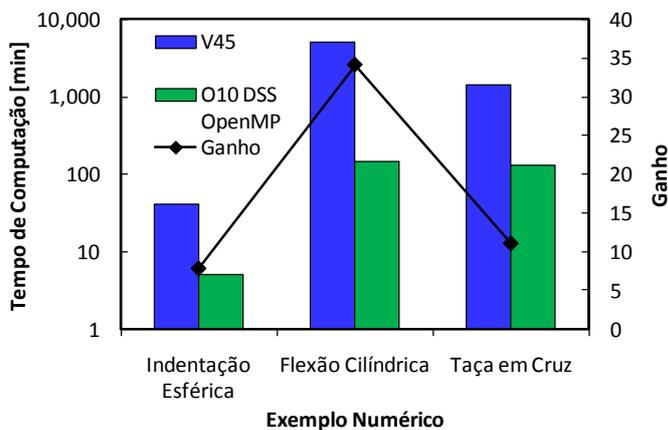
12

Benchmark: estampagem de uma taça quadrada

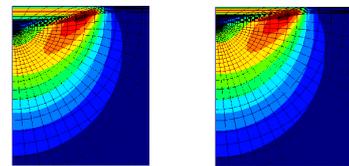


13

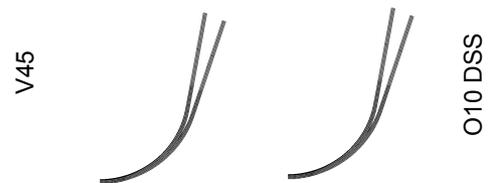
Outros benchmarks:



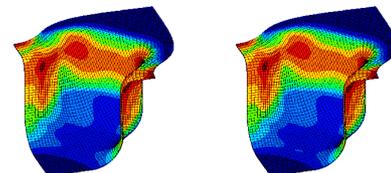
Indentação Esférica



Flexão Cilíndrica



Taça em Cruz



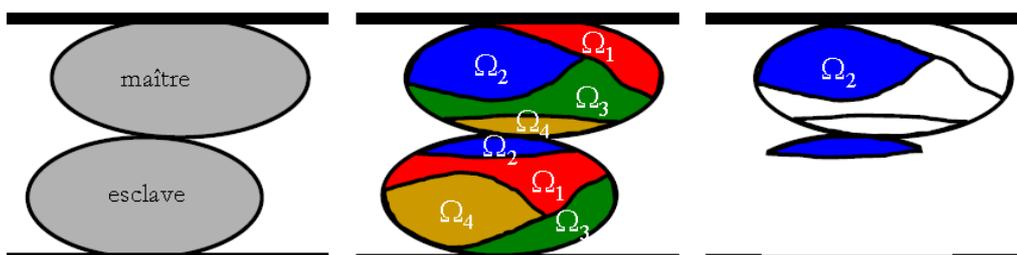
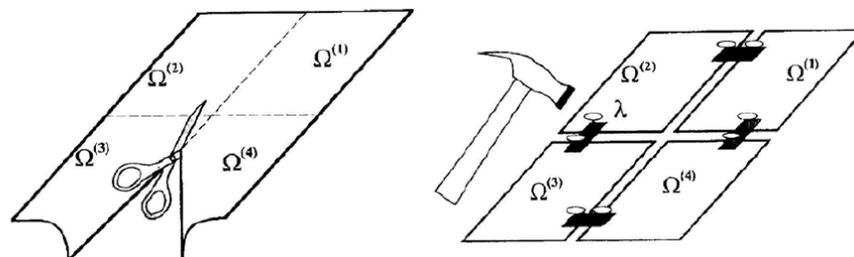
14

- Os resultados apresentados mostram que:
 - A adopção do método de resolução de sistemas de equações lineares, **Direct Sparse Solver**, permite resolver problemas de dimensões maiores, devido à gestão de memória mais eficiente.
 - A implementação de **directivas OpenMP** exige pouca, embora cuidadosa, programação e preserva a estrutura sequencial do código, o que permite a sua aplicação de forma faseada.
 - A implementação de directivas OpenMP conduzem a uma melhoria significativa na **eficiência computacional** do algoritmo global, inclusive em computadores pessoais.

15

1º, CPUs + GPUs

2º, DDM



16