SOLUÇÃO EM PARALELO DE UM MODELO TERMO-QUÍMICO-MECÂNICO PARA CONCRETO JOVEM

Iuri Alves Ferreira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

_____
Prof. Eduardo de Moraes Rego Fairbairn, Dr. Ing.

_____
Prof. Fernando Luiz Bastos Ribeiro, DSc.

_____
Prof. Philippe Remy Bernard Devloo, PhD.

_____
Prof. Túlio Nogueira Bittencourt, PhD.

_____
Prof. Álvaro Luiz Gayoso de Azeredo Coutinho, DSc.

_____
Dr. Marcos Martinez Silvoso, DSc.

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 2008

FERREIRA, IURI ALVES

Solução em Paralelo de um Modelo Termo-Químico-Mecânico Para Concreto Jovem

[Rio de Janeiro] 2008

XIV, 129 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Civil, 2008)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Elementos Finitos 2. Computação Paralela 3. Concreto

I. COPPE/UFRJ II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

SOLUÇÃO EM PARALELO DE UM MODELO TERMO-QUÍMICO-MECÂNICO PARA CONCRETO JOVEM

Iuri Alves Ferreira

Julho/2008

Orientadores: Eduardo de Moraes Rego Fairbairn

Fernando Luiz Bastos Ribeiro

Programa: Engenharia Civil

Apresenta-se neste trabalho um modelo numérico de alto desempenho para o problema termo-químico-mecânico do concreto nas primeiras idades, com aplicação na simulação da construção de estruturas de concreto de grande porte tais como barragens. O modelo acopla, por meio da termodinâmica, os diversos fenômenos que intervêm na reação de hidratação, quais sejam: exotermia; termo-ativação e dependência das propriedades térmicas e mecânicas no avanço da reação de hidratação. Desenvolveu-se uma estratégia de solução em paralelo do tipo subdomínio-por-subdomínio para o método dos elementos finitos, visando a utilização de máquinas de memória distribuída. Os sistemas de equações são resolvidos iterativamente, utilizando-se estruturas de dados comprimidas para o armazenamento das matrizes. Dois tipos de particionamento de malhas foram utilizados, *overlapping* e *non-overlapping*, e a comunicação entre processos é feita através do padrão MPI2. Desenvolveu-se também um esquema de reparticionamento dinâmico da malha para a simulação da construção em camadas de uma barragem. Exemplos numéricos demonstram a eficiência da metodologia proposta.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PARALLEL SOLUTION OF A THERMO-CHEMO-MECHANICAL MODEL FOR EARLY AGE CONCRETE

Iuri Alves Ferreira

July/2008

Advisors: Eduardo de Moraes Rego Fairbairn

Fernando Luiz Bastos Ribeiro

Department: Civil Engineering

This work presents a high performance numerical model for the problem of thermo-chemical-mechanical couplings of concrete at early ages, applied to the construction of large scale concrete structures such as dams. The model couples, within the framework of thermodynamics, the several phenomena that intervene in the hydration reaction, namely: exothermicity; thermal activation; and dependence of thermal and mechanical properties on the advancement of the hydration reaction. A parallel solution strategy for the finite element method, based on a subdomain-by-subdomain approach, was developed aiming the utilization of distributed memory architectures. The systems of equations are iteratively solved and compressed data structures are used to store the coefficient matrix. Two types of mesh partitioning were tested, overlapping and non-overlapping, and all inter-process communication is carried out using the MPI2 standard. A dynamic mesh repartitioning scheme was also developed, in order to simulate the construction by layers of a dam.

Índice do Texto

Índice do Texto v

Índice de Figuras viii

Índice de Tabelas xiii

Índice de Algoritmos xiv

1 Introdução 1

1.1 Relevância e objetivos 1

1.2 Metodologia e revisão bibliográfica 3

2 Acoplamento termo-químico-mecânico: quadro teórico 6

2.1 Modelagem do material 8

2.2 Quadro termodinâmico - Energia 9

2.2.1 Equações de estado 10

2.3 Acoplamento termo-químico 11

2.3.1 Cinética da reação de hidratação 11

2.3.2 Determinação da cinética de hidratação por experimento macroscópico 14

2.4 Acoplamentos químico-mecânicos 15

2.4.1 Plasticidade 16

2.4.2 Fluência 19

3 Modelo numérico 21

3.1 Módulo termo-químico 21

3.1.1 Validação: teste adiabático 23

3.2 Módulo mecânico 24

3.2.1 Cálculo de tensões 26

3.2.2 Fluência 27

3.2.3 Ativação das superfícies de plastificação 27

3.2.4 Mapeamento de retorno às superfícies 29

3.2.5 Validação: plasticidade 31
3.2.6 Validação: fluência 32
3.3 Construção por etapas de concretagem 34
4 Implementação em paralelo 36
4.1 Estrutura de dados 36
4.2 Particionamento da malha 38
4.2.1 Particionamento “non-overlapping” 38
4.2.2 Particionamento “overlapping” 40
4.3 Solução em paralelo 41
4.3.1 Método “non-overlapping” 42
4.3.2 Método “overlapping” 47
4.4 Comparação entre métodos de particionamento 49
4.5 Paralelização do modelo numérico acoplado para concreto 60
4.5.1 Particionamento com construção em camadas 61
5 Simulação de um vertedouro 63
5.1 Estrutura 63
5.2 Materiais 66
5.3 Plano de construção 69
5.4 Resultados 69
5.5 Desempenho 71
6 Simulação de uma casa de força 93
6.1 Estrutura 93
6.2 Materiais 96
6.3 Resultados 97
6.4 Desempenho 100
7 Conclusão 122

8 Referências Bibliográficas..... 125

Índice de Figuras

Figura 2.1: Processo de micro-difusão de água pelas camadas de hidtratos. 12

Figura 2.2: Superfícies de carregamento. 17

Figura 2.3: Variação de propriedades com o grau de hidratação. 19

Figura 3.1: Teste adiabático para validação do modelo cinético. 24

Figura 3.2: Método iterativo não-linear (NRM) de integração de forças. 25

Figura 3.3: Regiões usadas no mapeamento de retorno. 28

Figura 3.4: Mapeamento de retorno da região II. 30

Figura 3.5: Teste uniaxial com os dois critérios de plasticidade. 32

Figura 3.6: Fluência específica para concreto de referência. 33

Figura 3.7: Fluência específica para concreto blenda 34

Figura 4.1: Particionamento “Non-overlapping” 39

Figura 4.2: Particionamento “Overlapping”. 40

Figura 4.3: Mapeamento de equações local -> fronteira. 44

Figura 4.4: Mapeamento de equações fronteira -> local. 44

Figura 4.5: Problema dos poços de injeção e produção. 51

Figura 4.6: Speedups malha 1: non-overlapping x overlapping. 53

Figura 4.7: Speedups malha 1: core1 x core2, por processo. 54

Figura 4.8: Speedups malha 2: non-overlapping x overlapping. 54

Figura 4.9: Speedups malha 2: core1 x core2, por processo. 55

Figura 4.10: Speedups malha 3: non-overlapping x overlapping. 55

Figura 4.11: Speedups malha 3: core1 x core2, por processo. 56

Figura 4.12: Speedups malha 1: core1 x core2, por máquina 57

Figura 4.13: Speedups malha 2: core1 x core2, por máquina 58

Figura 4.14: Speedups malha 3: core1 x core2, por máquina 58

Figura 4.15: Speedups malha 3 non-overlapping por máquina. 59

Figura 4.16: Eficiências malha 3 non-overlapping..... 60

Figura 5.1: Partes da estrutura do vertedouro: solo, fundação, pilares e salto de esqui, rampa, curva do vertedouro, blocos munhão..... 64

Figura 5.2: Detalhe da estrutura do vertedouro: galeria. 65

Figura 5.3: Malha da estrutura do vertedouro. 65

Figura 5.4: Curvas de elevação adiabática de temperatura. 67

Figura 5.5: Propriedades x grau de hidratação, fundação..... 68

Figura 5.6: Propriedades x grau de hidratação, pilares, superfícies e munhão..... 68

Figura 5.7: Temperatura, corte transversal passando pela galeria..... 69

Figura 5.8: Temperatura, corte transversal passando pelo munhão. 69

Figura 5.9: Temperatura, corte longitudinal passando pelo pilar da direita. 70

Figura 5.10: Regiões plastificadas pelo critério Drucker-Prager. 71

Figura 5.11: Região plastificada pelo critério tension cut-off. 71

Figura 5.12: Particionamentos com 8 processos (camadas 10, 20 e 30). 73

Figura 5.13: Particionamentos com 64 processos (camadas 10, 20 e 30). 73

Figura 5.14: Número de coeficientes da matriz armazenados (nad) por camada. 74

Figura 5.15: Número de equações (neq) por camada. 74

Figura 5.16: Tempo seqüencial (n=1, p=1) por camada [s]. 75

Figura 5.17: Tempo seqüencial (n=1, p=1) acumulado [s]. 76

Figura 5.18: Tempo paralelo (n=1, p=2) por camada [s]. 76

Figura 5.19: Tempo paralelo (n=1, p=2) acumulado [s]. 77

Figura 5.20: Tempo paralelo (n=2, p=4) por camada [s]. 77

Figura 5.21: Tempo paralelo (n=2, p=4) acumulado [s]. 78

Figura 5.22: Tempo paralelo (n=4, p=8) por camada [s]. 78

Figura 5.23: Tempo paralelo (n=4, p=8) acumulado [s]. 79

Figura 5.24: Tempo paralelo (n=8, p=16) por camada [s]. 79

Figura 5.25: Tempo paralelo (n=8, p=16) acumulado [s]. 80
Figura 5.26: Tempo paralelo (n=16, p=32) por camada [s]. 80
Figura 5.27: Tempo paralelo (n=16, p=32) acumulado [s]. 81
Figura 5.28: Tempo paralelo (n=32, p=64) por camada [s]. 81
Figura 5.29: Tempo paralelo (n=32, p=64) acumulado [s]. 82
Figura 5.30: Speedups nos elementos, problema térmico, por camada. 82
Figura 5.31: Speedups no solver PCGD, problema térmico, por camada. 83
Figura 5.32: Speedups nos elementos, problema mecânico, por camada. 83
Figura 5.33: Speedups no solver PCGD, problema mecânico, por camada. 84
Figura 5.34: Speedups totais da análise por camada. 84
Figura 5.35: Speedups acumulados por nó. 85
Figura 5.36: Eficiência nos elementos, problema térmico, por camada. 85
Figura 5.37: Eficiência no solver PCGD, problema térmico, por camada. 86
Figura 5.38: Eficiência nos elementos, problema mecânico, por camada. 86
Figura 5.39: Eficiência no solver PCGD, problema mecânico, por camada. 87
Figura 5.40: Eficiência total da análise por camada. 87
Figura 5.41: Eficiência acumulada por nó. 88
Figura 5.42: Taxa de comunicação por processo para 32 nós. 89
Figura 5.43: Taxa de comunicação por processo para 16 nós. 89
Figura 5.44: Taxa de comunicação por processo para 8 nós. 90
Figura 5.45: Taxa de comunicação por processo para 4 nós. 90
Figura 5.46: Taxa de comunicação por processo para 2 nós. 91
Figura 6.1: Geometria da cada de força: vista isométrica. 94
Figura 6.2: Malha da cada de força: vista lateral. 94
Figura 6.3: Malha da cada de força: vista superior. 95
Figura 6.4: Malha da cada de força: vista de montante. 95

Figura 6.5: Curva de elevação adiabática de temperatura. 96

Figura 6.6: Propriedades vs grau de hidratação, casa de força. 97

Figura 6.7: Temperatura em corte longitudinal da casa de força. 98

Figura 6.8: Índice de fissuração ($\sigma 1/ft(\xi)$). 99

Figura 6.9: Particionamentos com 8 processos, camada 20. 100

Figura 6.10: Particionamentos com 8 processos, camada 40. 101

Figura 6.11: Particionamentos com 8 processos, camada 60. 101

Figura 6.12: Particionamentos com 32 processos, camada 20. 102

Figura 6.13: Particionamentos com 32 processos, camadas 40. 102

Figura 6.14: Particionamentos com 32 processos, camada 60. 103

Figura 6.15: Número de coeficientes da matriz armazenados (nad) por camada. 103

Figura 6.16: Número de equações (neq) por camada. 104

Figura 6.17: Tempo seqüencial (n=1, p=1) por camada [s]. 104

Figura 6.18: Tempo seqüencial (n=1, p=1) acumulado [s]. 105

Figura 6.19: Tempo paralelo (n=1, p=2) por camada [s]. 105

Figura 6.20: Tempo paralelo (n=1, p=2) acumulado [s]. 106

Figura 6.21: Tempo paralelo (n=2, p=4) por camada [s]. 106

Figura 6.22: Tempo paralelo (n=2, p=4) acumulado [s]. 107

Figura 6.23: Tempo paralelo (n=4, p=8) por camada [s]. 107

Figura 6.24: Tempo paralelo (n=4, p=8) acumulado [s]. 108

Figura 6.25: Tempo paralelo (n=8, p=16) por camada [s]. 108

Figura 6.26: Tempo paralelo (n=8, p=16) acumulado [s]. 109

Figura 6.27: Tempo paralelo (n=16, p=32) por camada [s]. 109

Figura 6.28: Tempo paralelo (n=16, p=32) acumulado [s]. 110

Figura 6.29: Tempo paralelo (n=32, p=64) por camada [s]. 110

Figura 6.30: Tempo paralelo (n=32, p=64) acumulado [s]. 111

Figura 6.31: Speedups nos elementos, problema térmico, por camada. 111
Figura 6.32: Speedups no solver PCGD, problema térmico, por camada. 112
Figura 6.33: Speedups nos elementos, problema mecânico, por camada. 112
Figura 6.34: Speedups no solver PCGD, problema mecânico, por camada. 113
Figura 6.35: Speedups totais da análise por camada. 113
Figura 6.36: Speedups acumulados por nó. 114
Figura 6.37: Eficiência nos elementos, problema térmico, por camada. 114
Figura 6.38: Eficiência no solver PCGD, problema térmico, por camada. 115
Figura 6.39: Eficiência nos elementos, problema mecânico, por camada. 115
Figura 6.40: Eficiência no solver PCGD, problema mecânico, por camada. 116
Figura 6.41: Eficiência total da análise por camada. 116
Figura 6.42: Eficiência acumulada por nó. 117
Figura 6.43: Taxa de comunicação por processo para 32 nós. 117
Figura 6.44: Taxa de comunicação por processo para 16 nós. 118
Figura 6.45: Taxa de comunicação por processo para 8 nós. 118
Figura 6.46: Taxa de comunicação por processo para 4 nós. 119
Figura 6.47: Taxa de comunicação por processo para 2 nós. 119

Índice de Tabelas

Tabela 3.1: Equações das retas no plano $\boldsymbol{S}\ vs\ I1$. 28

Tabela 3.2: Parâmetros de ativação. 29

Tabela 3.3: Testes de ativação. 29

Tabela 3.4 Parâmetros do modelo de fluência. 33

Tabela 4.1: Contagem *a priori* de OPFs no solver PCGD. 50

Tabela 4.2: Dados do problema. 51

Tabela 4.3: Redundância no PCGD (%). 52

Tabela 5.1: Propriedades de materiais. 67

Tabela 5.2: Características de hardware e software. 72

Tabela 6.1: Propriedades de materiais: casa de força. 96

Índice de Algoritmos

Algoritmo 3.1: Integração da equação de calor. 23

Algoritmo 3.2: Integração da equação cinética do grau de hidratação. 23

Algoritmo 4.1: Produto matriz-vetor no CSRC simétrico. 37

Algoritmo 4.2: Produto matriz-vetor no CSRC não-simétrico. 37

Algoritmo 4.3: Método PCGD (ver (Saad 2003)). 38

Algoritmo 4.4: Rotina de comunicação “non-overlapping”. 43

Algoritmo 4.5: Produto matriz-vetor no CSRCR, parte retangular. 47

Algoritmo 4.6: Rotina de comunicação “overlapping”. 48

Algoritmo 4.7: Esquema de re-particionamento após alteração de geometria. 61

1 INTRODUÇÃO

1.1 RELEVÂNCIA E OBJETIVOS

A crise no setor elétrico ocorrida na década de 90 (Rosa e Lomardo 2004) no Brasil evidencia a necessidade de investimento em geração de energia elétrica. Entre as opções de forma de geração, a usina hidrelétrica é forte candidata, uma vez que temos ainda potencial inexplorado, além de ser uma forma não poluente de geração.

Uma estrutura de barragem envolve peças estruturais (seções da barragem, seções do vertedouro, caixas espirais, casas de força, etc.) constituídas de centenas de milhares ou até milhões de metros cúbicos de concreto. Os principais problemas experimentados na fase de construção dessas estruturas são as tensões e deformações com origem na evolução da reação de hidratação do cimento associadas a campos transientes de grau de hidratação, temperatura, módulo de elasticidade e resistência. Todos estes campos atuam sobre as camadas de

concretagem desde o momento de seu lançamento na estrutura, o que pode resultar em fissuração. A fissuração pode diminuir a utilidade e a durabilidade da barragem, mesmo quando não chega a afetar a estabilidade estrutural da mesma. Com relação ao custo da obra, pode-se considerar, de forma simplificada, que a opção por camadas de maior espessura, a produção e lançamento do concreto em temperatura o mais próximo possível da ambiente e o baixo consumo de aglomerante implicam em maior economia. Por outro lado, camadas mais esbeltas e uso de sistema de resfriamento (pré ou pós-lançamento), implicam em uma menor tendência a ocorrer fissuração de origem térmica, enquanto que um material com maior proporção de aglomerante significa, grosso modo, rede porosa mais refinada e menos conectada, e conseqüentemente, maior durabilidade, impermeabilidade e resistência (Hewlett 2004),(Skalny 1997),(Taylor 1997),(Waller 1999),(Van Mier 1996). Entretanto, um maior consumo de material cimentante está geralmente relacionado a um aumento das temperaturas originadas na hidratação.

Portanto, para lidar com este problema, é necessário capturar adequadamente o comportamento intrínseco do material, bem como fazê-lo dentro de uma simulação numérica do processo construtivo da estrutura. Isto significa considerar efeitos locais (na escala de centímetros) em uma estrutura que alcança dezenas e até centenas de metros. Além disso, estes efeitos (campos térmicos, químicos e mecânicos) são transientes, o que significa ter que resolver um problema tri-dimensional com elevado número de equações (da ordem de 10^6) um grande número de vezes (ordem de 10^4).

O objetivo deste trabalho é dar uma contribuição significativa na modelagem computacional deste tipo de problema, tanto na concepção do modelo que envolve acoplamentos termo-químico-mecânicos como no desenvolvimento de ferramentas numéricas de alto desempenho. Dentro deste contexto, apresenta-se uma implementação em paralelo do método dos elementos finitos para a solução de um modelo termo-químico-mecânico para concreto jovem.

1.2 METODOLOGIA E REVISÃO BIBLIOGRÁFICA

O modelo termo-químico-mecânico proposto baseia-se num modelo constitutivo de material para concreto desenvolvido por Ulm e colaboradores (Ulm e Coussy 1996), (Coussy 1995), (Bernard, Ulm e Lemarchand 2003), (Hellmich e Ulm 1999), (Lemarchand, Ulm e Dormieux 2002). Este modelo permite lidar com a interdependência entre os comportamentos térmico, químico e mecânico do concreto variando no tempo. O modelo acopla, por meio da termodinâmica, os diversos fenômenos que intervêm na reação de hidratação, quais sejam: exotermia; termo-ativação e dependência das propriedades térmicas e mecânicas no avanço da reação de hidratação. Além deste quadro teórico, que inclui plasticidade e fluência, foram adotadas as leis de variação das propriedades com o grau de hidratação propostas em (Krauss e Hariri 2006), com adaptações aos materiais usados nas estruturas simuladas neste trabalho. A implementação realizada permite simular a construção por etapas, com condições de contorno adequadas, como atualização das superfícies de convecção e restrições.

Análises numéricas tri-dimensionais pelo método dos elementos finitos do problema acima descrito aplicado a barragens requerem a solução de dezenas de

milhares de sistemas de equações envolvendo milhões de graus de liberdade. Para que o problema possa ser efetivamente resolvido é necessário portanto o desenvolvimento de ferramentas numéricas de alto desempenho computacional.

Códigos paralelos de elementos finitos podem ser implementados em plataformas de memória compartilhada, distribuída ou compartilhada/distribuída. Sistemas de memória compartilhada trabalham em códigos por "threads", enquanto em sistemas de memória distribuída cada processador executa seu próprio código, e as comunicações entre nós são realizadas por alguma interface de troca de mensagens, tal como o padrão MPI. Em sistemas distribuídos, métodos tipo decomposição de domínio (DD) têm sido amplamente usados e estão ainda sob intensa pesquisa (Rao 2005), (Rebollo e Vera 2004). Técnicas DD usam uma estratégia "dividir e conquistar", na qual um sistema complemento de Schur provê a solução na interface enquanto um sistema no subdomínio é resolvido em cada processador. Muitas variantes destes métodos, empregando métodos diretos e/ou iterativos, podem ser encontrados na literatura (ver por exemplo (Rao 2005), (Dostal, Horak e Kucera 2005), (Charmpis e Papadrakis 2002), (Papadrakis e Bitzarakis 1996), (Soulaimani, Saad e Rebaine 2001), (Amestoy, Duff e L'Excellent 2000)). No caso particular de métodos iterativos, estratégias subdomínio-por-subdomínio (SBS) parecem ser uma boa alternativa aos métodos DD. Na abordagem SBS, os códigos serial e paralelo são exatamente o mesmo, com cada processador realizando tarefas locais sobre sua partição correspondente. Portanto a convergência não é afetada pela paralelização e uma melhor escalabilidade pode ser esperada. Em (Papadrakis e Bitzarakis 1996) e (Bitzarakis, Papadrakakis e Kotsopulos 1997) a abordagem SBS é usada como um procedimento

complementar a um método DD, enquanto em (Liu, Leung e Woo 2003) um esquema totalmente SBS é desenvolvido para a solução das equações de Navier-Stokes para fluido incompressível. Outro trabalho que usa abordagem semelhante é (Bova e Carey 2000).

O que se propõe neste trabalho é a utilização de um código do método dos elementos finitos desenvolvido em paralelo para arquiteturas de memória distribuída. A paralelização é feita dentro de um esquema SBS, e estruturas de dados comprimidas são utilizadas para o armazenamento das matrizes de coeficientes. Os sistemas de equações são resolvidos iterativamente através do método de gradientes conjugados. A estrutura de dados utilizada é o CSRC (compressed storage row/columns), que é uma variante da estrutura CSR (compressed storage row) e foi apresentada em (Ribeiro e Ferreira 2007), para matrizes simétricas e não simétricas. Para esta estrutura de dados utilizada (CSRC) desenvolveu-se um algoritmo especial para efetuar produtos matriz-vetor. A estrutura de dados CSRC juntamente com o algoritmo desenvolvido para o produto matriz-vetor permitem uma redução significativa tanto em termos de memória como em termos de tempo de execução, sendo ainda adequada à paralelização. A comunicação entre processos é feita através de rotinas do padrão MPI. Na implementação paralela do modelo para concreto a poucas idades o particionamento da malha é dinâmico, sendo a malha re-particionada a cada alteração na estrutura (lançamento de camada de concreto).

2 ACOPLAMENTO TERMO-QUÍMICO-MECÂNICO: QUADRO TEÓRICO

A construção de estruturas maciças de concreto requer que seja levada em consideração a integridade mecânica do material desde o momento do seu lançamento, pois desde então já atuam esforços que podem comprometê-la. Estes esforços são oriundos tanto das ações mecânicas devidas a peso próprio e forças externas quanto de campos térmicos resultantes da exotermia da reação de hidratação e do clima. Para tal consideração é necessário conhecer as propriedades mecânicas, i.e. a relação constitutiva entre tensões, deslocamentos e alguma medida de dano interno, como função do grau de hidratação, bem como conhecer a própria cinética da reação de hidratação. Em termos simples, saber a cada instante: (i) quanta hidratação já ocorreu; (ii) qual o efeito disto na rigidez e capacidade de carga do material, e, (iii) dadas as forças e restrições atuantes na estrutura, qual o comportamento desta; e (iv) qual o efeito deste estado na própria reação de hidratação.

A primeira questão se enquadra na química do cimento. A segunda questão evidencia o acoplamento existente entre a micro-estrutura do material cimentício, que é função do grau de hidratação, e seu comportamento mecânico macroscopicamente observável.

A terceira questão pode ser abordada a partir da combinação de diversos ramos[1] da mecânica, como: teoria da elasticidade, teoria de plasticidade, teoria de dano, mecânica da fratura, micromecânica e teoria de percolação.

A mecânica de materiais tem evoluído de modelos de material homogêneo (elasticidade, plasticidade, visco-plasticidade:(Coussy 1995),(Hill 1998),(Lemaitre e Chaboche 1998)) para modelos que consideram com cada vez mais profundidade as heterogeneidades do material (fratura:(Bazant e Planas 1998),(Van Mier 1996), micromecânica: (Nemat-Nasser e Hori 1999), ,(Pensée, Kondo e Dormieux 2002), percolação (Grimmett 1999),(Stauffer e Aharony 1994), física do estado sólido (Komarovsky 2003)). A par disso, a investigação experimental do material a nível micrométrico e até nanométrico tem também evoluído (Lemarchand, Ulm e Dormieux 2002),(Velez, et al. 2001).

Neste trabalho adotaremos uma abordagem baseada naquela desenvolvida ao longo de diversos trabalhos como (Ulm e Coussy 1996), (Coussy 1995), (Bernard, Ulm e Lemarchand 2003), (Hellmich e Ulm 1999), (Lemarchand, Ulm e Dormieux 2002) por ser esta linha de modelagem constitutiva bastante rigorosa na

[1] Esta lista não exaure as possibilidades.

fundamentação em termodinâmica. Segue uma breve descrição dos principais pontos do modelo.

2.1 MODELAGEM DO MATERIAL

O concreto é um meio poroso quimicamente reativo (Coussy 1995). Inicialmente fluido (tipo Bingham ou Herschel Buckley), ele torna-se um sólido poroso após atingir o patamar de percolação (Krauss e Hariri 2006), (Ulm e Coussy 1996). Admite-se a existência de um volume elementar de referência (VER), tal que seja grande o suficiente para conter todas as fases constituintes do material, mas pequeno o suficiente para que suas propriedades médias não variem de um VER para outro, dadas as mesmas condições de hidratação, temperatura e tensão. Isto equivale a dizer que o VER é representativo. O VER é considerado fechado quanto a troca de massa. A deformação e a tensão observáveis (macroscópicas) são as do esqueleto sólido, sendo a poro-pressão negligenciada.

A fase sólida é constituída de cimento anidro e hidratos, e a fase fluida de água livre e ar. À medida que a água livre reage com o cimento anidro formando hidratos, a fração volumétrica e a conexão entre grãos da matéria sólida aumentam, ao mesmo tempo em que a fração volumétrica e a conexão entre volumes das fases fluidas diminuem. Estes processos acarretam maior resistência da massa à deformação. Enquanto o material está fluido isto representa maior viscosidade, quando sólido, maior rigidez. O processo de solidificação descrito resulta também em maior resistência mecânica, dado que há mais matéria para suportar as cargas.

2.2 QUADRO TERMODINÂMICO - ENERGIA

Partindo das leis da termodinâmica, as relações constitutivas são derivadas (Hellmich 1999) entre as variáveis de estado utilizadas para descrever o problema. Sejam a primeira e segunda leis da termodinâmica dadas por:

$$\dot{e}_i = \boldsymbol{\sigma}: \dot{\boldsymbol{\varepsilon}} + r - div\boldsymbol{q} \quad e \quad \dot{S} \geq -div\frac{\boldsymbol{q}}{T} + \frac{r}{T} \qquad (2.1)\text{a,b}$$

Sendo e_i a energia interna do VER, $\boldsymbol{\sigma}$ o tensor de tensões, $\boldsymbol{\varepsilon}$ o tensor de deformações totais, r a fonte volumétrica de calor, $\boldsymbol{q}$ o vetor de fluxo de calor, S a entropia, e T a temperatura absoluta. Assumindo a hipótese de pequenas deformações e pequenos deslocamentos, seja a energia livre de Helmholtz Ψ:

$$\Psi = e - TS = \psi(T, \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p - \boldsymbol{\varepsilon}^f, m_s, w) + U(\boldsymbol{\chi}, \gamma, m_s, w) \qquad (2.2)$$

onde ψ é a energia elástica de deformação, $\boldsymbol{\varepsilon}^p$ é o tensor de deformações plásticas, $\boldsymbol{\varepsilon}^f$ é o tensor de deformações de fluência de longo prazo, m_s é a massa de água ligada quimicamente ao cimento, $\boldsymbol{\chi}$ é um vetor de variáveis internas de plasticidade (hardening), U é a "energia congelada" em processos internos de dissipação, γ é o deslocamento viscoso da fluência de longo prazo e w é a massa de água envolvida na deformação de fluência de curto prazo. Usando (2.1)a,b e (2.2) chega-se à seguinte forma para a desigualdade de Clausius-Duhem:

$$\varphi = \boldsymbol{\sigma}: \dot{\boldsymbol{\varepsilon}}^p + \boldsymbol{\sigma}: \dot{\boldsymbol{\varepsilon}}^f + A_m\dot{m}_s + A_w\dot{w} + \boldsymbol{\zeta}\dot{\boldsymbol{\chi}} + \Gamma\dot{\gamma} - \frac{\boldsymbol{q}}{T}gradT \geq 0 \qquad (2.3)$$

onde φ é a dissipação, A_m é a afinidade química da reação de hidratação, A_w é a força termodinâmica associada à difusão poro-capilar de água, $\boldsymbol{\zeta}$ é a força de

"hardening", e Γ é a força de micro pré-tensão. Em (2.3), foram usadas as equações de estado, discutidas adiante.

2.2.1 Equações de estado

As derivadas parciais de Ψ em relação às variáveis de estado são identificadas respectivamente pelas equações de estado como:

$$S = -\frac{\partial \psi}{\partial T};\ \boldsymbol{\sigma} = \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}};\ A_m = -\frac{\partial \Psi}{\partial m_s};\ A_w = -\frac{\partial \Psi}{\partial w};\ \boldsymbol{\zeta} = -\frac{\partial U}{\partial \boldsymbol{\chi}};\ \Gamma = -\frac{\partial U}{\partial \gamma} \quad (2.4)\text{a-f}$$

De acordo com a simetria de Maxwell $\left(\frac{\partial^2 \Psi}{\partial \alpha\, \partial \beta} = \frac{\partial^2 \Psi}{\partial \beta\, \partial \alpha}\right)$ as derivadas parciais de segunda ordem da energia de Helmholtz são simétricas com relação às variáveis de estado. Usando esta relação e reescrevendo as equações de estado na forma diferencial, temos, por exemplo:

$$dS = -\frac{\partial^2 \psi}{\partial T^2} dT - \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial T} : (d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p - d\boldsymbol{\varepsilon}^f) - \frac{\partial^2 \psi}{\partial T \partial m_s} dm_s \quad (2.5)$$

$$d\boldsymbol{\sigma} = \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon}^2} : (d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p - d\boldsymbol{\varepsilon}^f) + \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial T} dT + \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial m_s} dm_s + \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial w} dw \quad (2.6)$$

$$d\boldsymbol{\zeta} = -\frac{\partial^2 U}{\partial \boldsymbol{\chi} \partial m_s} dm_s - m_s \frac{\partial^2 U}{\partial \boldsymbol{\chi}^2} d\boldsymbol{\chi} \quad (2.7)$$

Os termos omitidos nas equações acima representam acoplamentos fracos (Hellmich 1999),(Hellmich e Ulm 1999),(Ulm e Coussy 1996), i.e., têm valor desprezível comparados aos termos mantidos nas equações. Estas hipóteses são ditas de desacoplamento parcial.

2.3 ACOPLAMENTO TERMO-QUÍMICO

Substituindo (2.5) em (2.3) obtemos, após as hipóteses de desacoplamento parcial mencionadas acima, uma equação na forma da equação de condução de calor.

$$c_e\dot{T} + div\boldsymbol{q} = L_m\dot{m}_s \tag{2.8}$$

Aqui, $c_e = -T\frac{\partial^2\psi}{\partial T^2}$ é o calor específico e $L_m = T\frac{\partial^2\psi}{\partial T\partial m_s}$ é o calor latente de hidratação, que é positivo dado que a reação é exotérmica. Considerando c_e e L_m independentes de $\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^p$ e $\boldsymbol{\chi}$, temos o problema termo-químico (2.8) independente do químico-mecânico (2.16). Desta forma podemos integrar (2.8) e então usar o resultado $\{T(t), m_s(t)\}$ como dados em (2.16).

2.3.1 Cinética da reação de hidratação

O montante de reação ocorrido até um determinado instante t tem sido referido como $m_s(t)$, ou massa de esqueleto sólido formada. O cálculo do termo $\dot{m}_s$ requer conhecimento da cinética da reação de hidratação. Os processos fundamentais associados à cinética de hidratação são (Hewlett 2004),(Skalny 1997),(Taylor 1997): difusão de água através das camadas de hidratos formadas em torno dos grãos de cimento anidro, dissolução do cimento anidro, reação propriamente e nucleação, ou precipitação de hidratos recém formados em volta do grão.

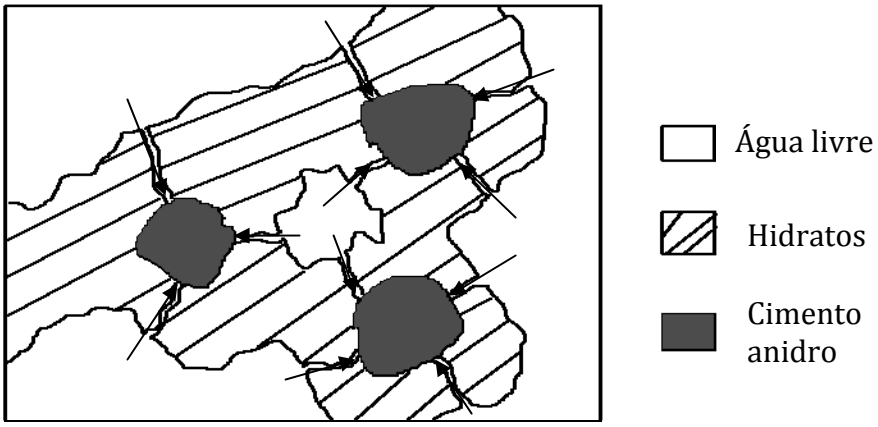


Figura 2.1: Processo de micro-difusão de água pelas camadas de hidtratos.

Dentre estes processos, o dominante no tocante à cinética é a difusão (Ulm e Coussy 1996), uma vez que os outros ocorrem em escalas de tempo bem inferiores a este.

Consideremos a cinética de uma forma integrada através do modelo básico usado para materiais cimentícios, que é a lei de Arrhenius (2.9).

$$\dot{m}_s = \frac{1}{\eta} A exp\left(-\frac{E_a}{RT}\right) \tag{2.9}$$

O lado esquerdo da equação representa uma medida de taxa temporal de avanço da reação. A variável η representa uma medida de viscosidade. O termo A é a afinidade química da reação, que é função das entalpias de formação dos compostos envolvidos. Os termos na exponencial são: a energia de ativação E_a, a constante universal dos gases R e a temperatura absoluta T. Além de exotérmica, a hidratação do cimento é termicamente ativada. Esta termo-ativação é considerada através do termo exponencial na temperatura, que implica em maior taxa de reação para maiores temperaturas.

Na equação (2.4)c considera-se que $A_m = A(m_s)$, i.e., depende apenas da reação química. Normalizando a variável m_s passamos a referenciar o grau de hidratação (2.10) em vez da massa de sólidos. O grau de hidratação é a medida da quantidade de reação química decorrida. Por ser normalizado, tem seu valor no intervalo [0,1], sendo que o zero corresponde a nenhuma hidratação ocorrida e 1 corresponde a reação completa (assintótica, pois dificilmente ocorre a hidratação de toda a massa do cimento usado na mistura).

$$\xi(t) = \frac{m_{cr}(t)}{m_{c\infty}} \tag{2.10}$$

onde $m_{cr}(t)$ é a massa de cimento já consumida na reação, e $m_{c\infty}$ a massa assintótica de cimento hidratado para $t = \infty$. Fazendo a transformação de m_s em ξ e embutindo η em A, temos:

$$\dot{\xi} = \tilde{A}(\xi) exp\left(-\frac{E_a}{RT}\right) \tag{2.11}$$

onde $\tilde{A}(\xi)$ é a afinidade normalizada. Tanto $\tilde{A}(\xi)$ quanto a energia de ativação E_a, são propriedades intrínsecas de uma dada dosagem. Por dosagem entenda-se não só o "traço" da mistura, mas todas as especificidades dos materiais usados, desde a composição química do cimento ao tipo litológico dos agregados. Se qualquer um dos materiais for alterado, ou se a proporção entre eles for alterada, tem-se um novo concreto, e por conseguinte um novo par $\{\tilde{A}(\xi), E_a\}$.

2.3.2 Determinação da cinética de hidratação por experimento macroscópico

A análise da equação (2.8) em condição adiabática ($div\mathbf{q} = 0$) revela a seguinte relação entre as taxas temporais de temperatura adiabática e grau de hidratação.

$$c_e \dot{T}^{ad} = L\dot{\xi} \tag{2.12}$$

onde L é uma normalização de L_m. A rigor a hidratação é um conjunto de reações químicas simultâneas, uma vez que o aglomerante é composto de várias substâncias, ou fases, distintas. As principais fases minerais são $(CaO)_3(SiO_2)$, $(CaO)_2(SiO_2)$, $(CaO)_3(Al_2O_3)$, $(CaO)_4(Al_2O_3)(Fe_2O_3)$ abreviadas pelos químicos de cimento como C_3S, C_2S, C_3A e C_4AF. Considera-se que uma dada fase x, a quantidade de calor liberada pela hidratação de uma determinada massa Δm_x independe de quanta hidratação já ocorreu, i.e. L_x independe de ξ.

Admitindo por simplificação que a proporção em massa entre as fases do aglomerante se mantém constante durante a hidratação, podemos estender a independência dos L_x em relação a ξ para o L global do cimento, que por fim resulta constante. O calor latente de hidratação L_x é invariável, i.e. independe da procedência do cimento, e seu valor para as diversas fases componentes do cimento está estabelecido na literatura (Waller 1999). Para obter um valor global de L [J/m^3] para um determinado cimento (Bernard, Ulm e Lemarchand 2003),(Ferreira, Fairbairn, et al. 2004), basta compor os valores por fase em uma média ponderada.

$$L = m_0 \sum_{x=1}^{n} m_x L_x \tag{2.13}$$

onde m_x $[Kg(x)/Kg(c)]$ é a proporção em massa da fase x no cimento, L_x $[J/Kg]$ é seu calor de hidratação e m_0 $[Kg]$ é a massa inicial de cimento em um volume unitário de concreto. Pelas considerações acerca de L, obtemos uma relação linear entre ξ e $\dot{T}^{ad}$. Integrando a cinética adiabática (2.12) temos:

$$\xi(t) = \left(T^{ad}(t) - T_0^{ad}\right)\frac{c_e}{L} \tag{2.14}$$

Note-se que $\xi(t=0) = 0$, e T_∞^{ad} corresponde a $\xi = 1$. Logo,

$$\xi(t) = \frac{\left(T^{ad}(t) - T_0^{ad}\right)}{\left(T_\infty^{ad} - T_0^{ad}\right)} \tag{2.15}$$

Conhecendo $T^{ad}(t)$ temos $\xi(t)$ e conseqüentemente $\tilde{A}(\xi)$ por meio de (2.11). Assim sendo, tem-se $\tilde{A}(\xi)$ de forma parametrizada por T^{ad}. Evidências experimentais (Ulm e Coussy 1996) comprovam o caráter intrínseco da afinidade normalizada encontrada via ensaio adiabático.

2.4 ACOPLAMENTOS QUÍMICO-MECÂNICOS

A equação (2.6), após algumas mudanças de variáveis e desenvolvimentos relativos à fluência (Hellmich e Ulm 1999), revela o acoplamento no sentido termo-químico → mecânico da equação constitutiva.

$$d\boldsymbol{\sigma} = \boldsymbol{C}(\xi) : (d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^p - d\boldsymbol{\varepsilon}^f - \boldsymbol{a}dT - \boldsymbol{b}d\xi - d\boldsymbol{\varepsilon}^v) \tag{2.16}$$

Em (2.16), $\boldsymbol{C} = \frac{\partial^2 \psi}{\partial(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^p)^2}$ é o tensor de quarta ordem de rigidez elástica. Como ψ é função de ξ, $\boldsymbol{C}$ também o é. Considerando isotropia, $\boldsymbol{C}(\xi)$ depende apenas de dois

parâmetros: Módulo de Young $E(\xi)$ e coeficiente de Poisson ν. O termo $\mathbf{a} = \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial T}$ é o tensor de dilatação térmica e representa o acoplamento termo-mecânico. Considerando dilatação volumétrica (hidrostática) tem-se $\mathbf{a} = \alpha \mathbf{1}$, onde $\mathbf{1}$ é o tensor diagonal unitário de segunda ordem, e α é o coeficiente linear de dilatação térmica. O termo $\mathbf{b} = \frac{\partial^2 \psi}{\partial \boldsymbol{\varepsilon} \partial \xi}$ é o tensor de retração autógena, i.e. retração devida à secagem dos poros por efeito da água consumida pela reação química. Considerando retração isotrópica, tem-se $\mathbf{b} = \beta \mathbf{1}$.

2.4.1 Plasticidade

A abordagem padrão da elastoplasticidade preconiza um critério de escoamento $F(\boldsymbol{\sigma}, \zeta)$. Este critério corresponde a uma superfície no espaço das tensões principais, que define o domínio elástico de tensões admissíveis.

$$F(\boldsymbol{\sigma}, \zeta) = f(\boldsymbol{\sigma}) - K(\zeta) \leq 0 \tag{2.17}$$

O limite de escoamento ζ é função das variáveis internas de plastificação $\boldsymbol{\chi}$**.** No modelo acoplado ζ é também função do avanço da reação de hidratação $\zeta = \zeta(\boldsymbol{\chi}, \xi)$ (2.7).

Dois critérios de plastificação são considerados: Drucker-Prager e limite de tração (Tension Cut-Off) (Ulm e Coussy 1996), (Hellmich e Ulm 1999). As funções que os definem são respectivamente (2.18) e (2.19).

$$F_1 = \alpha_{DP} I_1 + |\boldsymbol{S}| - K_{DP}(\chi_1, \xi) \tag{2.18}$$

$$F_2 = I_1 - T_C(\xi) \tag{2.19}$$

Nas equações acima, α_{DP} e $K_{DP}(\chi_1, \xi)$ são os coeficientes do critério Drucker-Prager, I_1 e $|\boldsymbol{S}|$ são invariantes de tensão e $T_C(\xi) = 3 f_t(\xi)$ é o limite de tração,

sendo f_t a resistência uniaxial à tração. As variáveis plásticas internas χ_1 e χ_2 representam uma medida da quantidade de deformação plástica ocorrida no material devido aos critérios Drucker-Prager e limite de tração respectivamente. A Figura 2.2 representa esquematicamente as superfícies de carregamento plásticas.

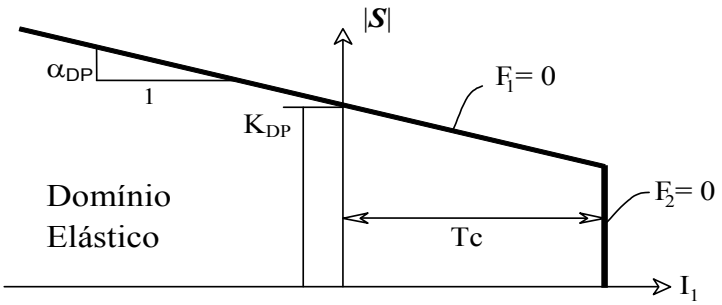


Figura 2.2: Superfícies de carregamento.

O aumento do domínio elástico, ou "hardening" ocorre de duas formas: puramente plástica, devido a deformações (χ_1); e química, devido ao avanço da reação de hidratação (ξ). Os parâmetros α_{DP} e K_{DP} são dados por:

$$\alpha_{DP} = \frac{f_{y2}/f_y - 1}{\sqrt{3}\left(2\, f_{y2}/f_y - 1\right)} \tag{2.20}$$

$$K_{DP} = \frac{\alpha_{DP} R(\xi)\left(f_{y2}/f_y\right)}{f_{y2}/f_y - 1} \tag{2.21}$$

onde

$$R(\xi) = f_y(\xi) + \Big(f_c(\xi) - f_y(\xi)\Big)\,(2 - \chi_1/\chi_{1u})(\chi_1/\chi_{1u}) \tag{2.22}$$

$$\chi_{1u} = \frac{9\alpha_{DP}(f_{c\infty} - E_\infty \varepsilon_{1u})}{E_\infty\left(3\alpha_{DP} - \sqrt{3}\right)} \tag{2.23}$$

Em (2.20) e (2.21) f_{y2} representa a tensão de escoamento biaxial à compressão e f_y, a tensão de escoamento uniaxial à compressão. As razões f_{y2}/f_y e f_y/f_c são consideradas constantes. Em (2.23) $f_{c\infty}$ e E_∞ são os valores assintóticos da resistência uniaxial a compressão módulo de Young respectivamente, enquanto ε_{1u}

representa a deformação correspondente a $f_{c\infty}$ e χ_{1u}, o limite de "hardening", após o qual a tem-se plasticidade perfeita também no critério Drucker-Prager. As equações contemplam as duas formas de "hardening". As variáveis plásticas χ_1 e χ_2 são calculadas a partir das deformações plásticas. Estas por sua vez são calculadas através da regra de fluxo e regra de "hardening". Os multiplicadores plásticos λ_j devem satisfazer a condição de consistência $df = 0$.

$$d\boldsymbol{\varepsilon}^p = \sum_j d\lambda_j \frac{\partial f_j}{\partial \boldsymbol{\sigma}} \tag{2.24}$$

$$\Delta\chi_1 + \Delta\chi_2 = tr(\Delta\boldsymbol{\varepsilon}^p) \tag{2.25}$$

As variações de resistências $f_c(\xi)$ e $f_t(\xi)$ e rigidez $E(\xi)$ com o grau de hidratação são calculadas de acordo com (Krauss e Hariri 2006), utilizando as funções:

$$\frac{\Pi(\xi)}{\Pi_\infty} = \begin{cases} 0, & \xi < \xi_0 \\ \gamma(\xi - \xi_0), & \xi_0 \le \xi < \alpha_1 \\ \dfrac{(\xi - \alpha_0)^n}{(1-\alpha_0)^n}, & \xi \ge \alpha_1 \end{cases} \qquad \gamma = \frac{(\alpha_1 - \alpha_0)^n}{(\alpha_1 - \xi_0)(1-\alpha_0)^n} \tag{2.26}$$

São considerados três trechos para a variação das propriedades: o primeiro constante e nulo para o grau de hidratação menor que ξ_0; o segundo linear, entre ξ_0 e o ξonde há perda de linearidade da relação (chamado aqui de α_1); e o terceiro que obedece a expressão $\frac{(\xi-\alpha_0)^n}{(1-\alpha_0)^n}$, para ξ maior que α_1. Um gráfico típico da variação das propriedades f_c, f_t e E com o grau de hidratação ξ está exemplificado na Figura 2.3.

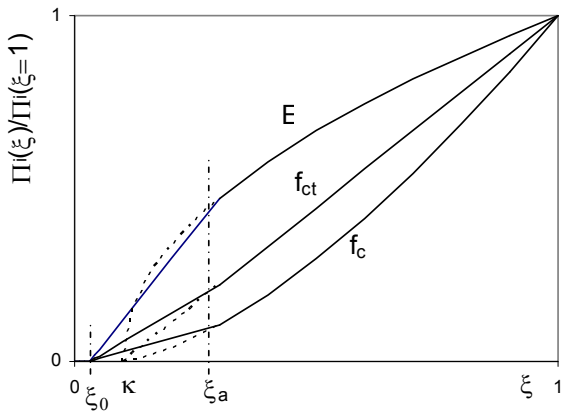


Figura 2.3: Variação de propriedades com o grau de hidratação.

O parâmetro n é dependente da propriedade analisada, e assume os seguintes valores: $n = 1$ para f_t, $n = 1{,}5$ para f_c e $n = 0{,}5$ para o módulo de elasticidade E (Krauss e Hariri 2006).

2.4.2 Fluência

Em (Sercombe, et al. 2000), a extensão do modelo de acoplamento para incluir fluência é apresentada. A fluência é separada em duas partes: curto prazo e longo prazo. A primeira é considerada como sendo causada pelo efeito da difusão de água através dos microporos do concreto. A segunda é considerada função do processo de dislocação entre as camadas de CSH, na escala nanométrica. A deformação viscosa de curto prazo é:

$$d\boldsymbol{\varepsilon}^{v}_{n+1} = \frac{1}{1 + \frac{\tau_w(\xi)}{dt}} [J^{v}_{\infty}\mathbf{G}\colon(\boldsymbol{\sigma_{n+1}}) - \boldsymbol{\varepsilon}^{v}_{n}] \tag{2.27}$$

Em (2.27) $\tau_w(\xi) = \tau_{w,\infty}\xi$ é o tempo característico, J^{v}_{∞} é a fluência específica assintótica e $\boldsymbol{G} = E\boldsymbol{C}^{-1}$. O fluxo de deformação de longo prazo é:

$$d\boldsymbol{\varepsilon}^{f}_{n+1} = \frac{\Delta t}{\eta_f(\Gamma_{n+1})}\mathbf{G}\colon\boldsymbol{\sigma_{n+1}} \tag{2.28}$$

Em (2.28) η_f é a viscosidade e Γ é a força de micro pré-tensão interna, que vem a ser a força governante da fluência de longo prazo. As equações a seguir completam o quadro incremental com o cálculo da viscosidade.

$$\frac{1}{\eta_f} = c'\Gamma_{n+1} exp\left[-\frac{2U}{R}\left(\frac{1}{T_{n+1}} - \frac{1}{T_0}\right)\right] \tag{2.29}$$

$$\Gamma_{n+1} = \Gamma_0(\xi_0) - H\gamma_{n+1} \tag{2.30}$$

$$d\gamma_{n+1} = dt c\Gamma_{n+1}^2 \exp\left[-\frac{U}{R}\left(\frac{1}{T_{n+1}} - \frac{1}{T_0}\right)\right] \tag{2.31}$$

Os parâmetros c e c' são constantes do material. T_{n+1} é a temperatura atual na simulação e T_0 é a temperatura de referência, à qual os parâmetros foram determinados, e γ é o deslizamento viscoso.

3 MODELO NUMÉRICO

A implementação em elementos finitos do modelo descrito foi feita em dois módulos independentes. O módulo termo-químico calcula campos de temperatura e o grau de hidratação. O módulo mecânico calcula os campos de deslocamentos, tensões e deformações com consideração de plasticidade e fluência.

3.1 MÓDULO TERMO-QUÍMICO

Os campos de temperatura são calculados segundo as equações transientes de difusão de calor (2.8),(3.1), onde k é a condutividade térmica, c é o calor específico e H representa uma fonte de calor, Ω representa o domínio espacial, Γ_f é a parte do contorno de Ω onde há prescrição de fluxo, e Γ_T é a parte do domínio onde há prescrição de temperatura.

$$\begin{cases} c\dot{T} + k\nabla^2 T = H \quad em\ \Omega \\ T = T_0 \quad em\ \Omega \\ \nabla T.\boldsymbol{n} = \boldsymbol{f} \quad em\ \Gamma_f \\ T = \bar{T} \quad em\ \Gamma_{\mathrm{T}} \end{cases} \tag{3.1}$$

A discretização espacial por elementos finitos e no tempo por diferenças finitas (regra trapezoidal) transformam o problema (3.1) em um problema incremental na forma (3.2), onde $\boldsymbol{K}$ é a matriz de condutividade térmica, $\boldsymbol{C}$ é a matriz de capacidade térmica e $\boldsymbol{R}$ é o vetor de carga, que compreende os efeitos das condições de contorno e de fonte de calor.

$$\left(\beta\boldsymbol{K} + \frac{1}{\Delta t}\boldsymbol{C}\right)\boldsymbol{T}_{n+1} = (1-\beta)\boldsymbol{R}_n + \beta\boldsymbol{R}_{n+1} - \left[(1-\beta)\boldsymbol{K} - \frac{1}{\Delta t}\boldsymbol{C}\right]\boldsymbol{T}_n \tag{3.2}$$

Os efeitos exotérmico e de termo-ativação da reação de hidratação são integrados no problema através de uma parcela de fonte no vetor $\boldsymbol{R}$. Esta parcela representa o calor latente de hidratação, calculado por $H = L\dot{\xi}$ em cada ponto material (ver seção 2.3), onde ξ é o grau de hidratação. A taxa de reação $\dot{\xi}$ depende da temperatura (termo-ativação, ver equação (2.11)). Por outro lado, a temperatura depende da taxa de reação através da taxa de geração de calor $\boldsymbol{R}$ (equação (3.2)). Esta dependência circular gera um problema não-linear entre temperatura e grau de hidratação, que é resolvido de forma iterativa pelo Algoritmo 3.1.

$\boldsymbol{T}_{n+1}^{(1)} = \boldsymbol{T}_n,\ \ \boldsymbol{\xi}_{n+1}^{(1)} = \boldsymbol{\xi}_n$
$k = 1$
$\boldsymbol{\xi}_{n+1}^{(k+1)} \leftarrow \left\{\boldsymbol{\xi}_n, \boldsymbol{T}_n, \boldsymbol{T}_{n+1}^{(k)}\right\}$ Cinética
$\boldsymbol{R}_{n+1}^{(k+1)} \leftarrow \boldsymbol{\xi}_{n+1}^{(1)}$
$\boldsymbol{T}_{n+1}^{(k+1)} \leftarrow \left\{\boldsymbol{T}_n, \boldsymbol{R}_n, \boldsymbol{R}_{n+1}^{(k+1)}\right\}$ Sistema linear
Se $\left\|\boldsymbol{T}_{n+1}^{(k+1)} - \boldsymbol{T}_{n+1}^{(k)}\right\| < \text{tol}$ fim
Senão $k = k + 1$

Algoritmo 3.1: Integração da equação de calor.

No Algoritmo 3.1 a quinta linha corresponde à solução do sistema linear representado na equação (3.2). No mesmo algoritmo terceira linha corresponde à atualização do grau de hidratação de acordo com a equação (2.11). A discretização da equação cinética (2.11) gera uma equação não-linear em ξ_{n+1}. A solução desta equação pelo método da posição falsa está resumida no Algoritmo 3.2.

$a = \xi_n + 0.001, \quad b = \xi_n + 0.999$
$c1 = \beta exp\left(-\frac{E_a}{RT_{n+1}}\right), \ \ c2 = (1-\beta)\tilde{A}(\xi_n)$
$f(a) = c2 + c1\tilde{A}(a) - a, \qquad f(b) = c2 + c1\tilde{A}(b) - b$
Posição falsa:

$x = \frac{af(b) - bf(a)}{f(b) - f(a)}$
$f(x) = c2 + c1\tilde{A}(x) - x$
Se $f(x) < 0$ fim
Senão
  Se $f(a)f(x) < 0$
    $b \leftarrow x, \quad f(b) \leftarrow f(x)$
  Senão
    $a \leftarrow x, \quad f(a) \leftarrow f(a)$
  Próxima iteração

Algoritmo 3.2: Integração da equação cinética do grau de hidratação.

3.1.1 Validação: teste adiabático

Para testar a acuidade do modelo cinético foi simulada a hidratação de um corpo de prova (CP), sob condição adiabática. A Figura 3.1 mostra a curva de elevação

adiabática experimental e também a resposta obtida com uso do código implementado.

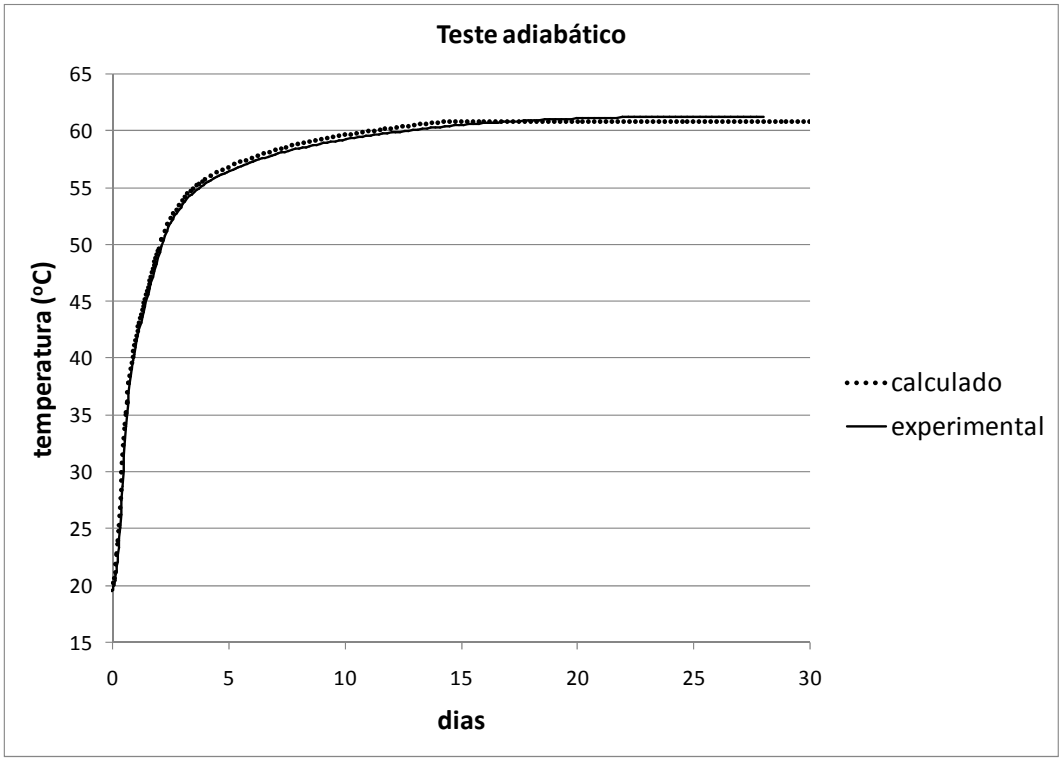


Figura 3.1: Teste adiabático para validação do modelo cinético.

A simulação foi feita com incrementos de tempo de tamanho variável (dt), de forma semelhante à que será adotada nas simulações de estruturas de grande porte nos capítulos de exemplos/resultados.

3.2 MÓDULO MECÂNICO

Os campos de deslocamento são calculados segundo a equação constitutiva (3.3), onde $\boldsymbol{b}$ representa forças de corpo, $\boldsymbol{u}$ os deslocamentos, $\boldsymbol{f}_0$ representa forças aplicadas, Ω representa o domínio espacial, Γ_u é a parte de Ω onde há prescrição de deslocamentos, e Γ_f é a parte do domínio onde há prescrição de forças.

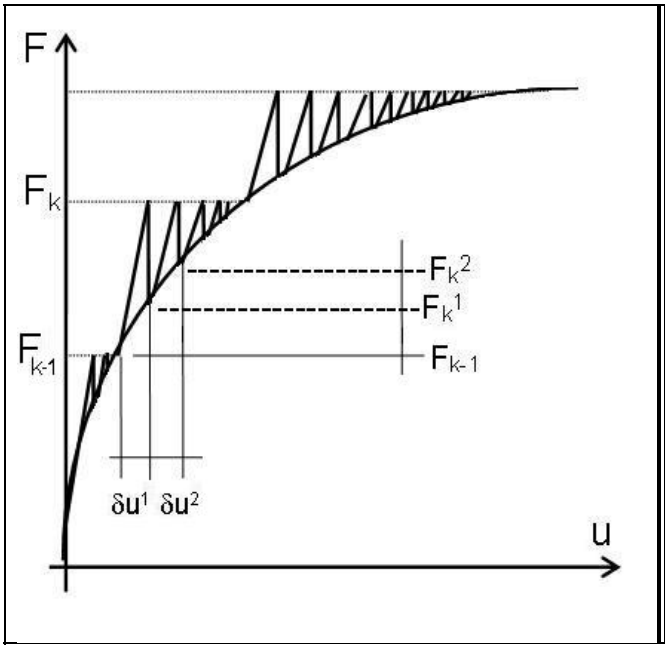
$$\begin{cases} div\boldsymbol{\sigma} - \boldsymbol{b} = \mathbf{0} \quad em\ \Omega \\ \boldsymbol{u} = \boldsymbol{u_0} \quad em\ \Gamma_u \\ \boldsymbol{f} = \boldsymbol{f_0} \quad em\ \Gamma_f \end{cases} \tag{3.3}$$

A discretização espacial por elementos finitos do problema (3.3) toma a forma incremental (3.4), onde $\boldsymbol{K}$ é a matriz de rigidez e $\boldsymbol{F}$ é o vetor de forças.

$$\boldsymbol{K}\Delta\boldsymbol{u}_{n+1} = \Delta\boldsymbol{F}_{n+1} \tag{3.4}$$

A força externa $\boldsymbol{F}_k$ corresponde a um incremento k no tempo. Cada incremento $\mathbf{F}_k$é resolvido de forma não-linear pelo esquema Newton-Raphson modificado (NRM), usando rigidez inicial, i.e. a matriz de rigidez $\boldsymbol{K}$ associada à configuração indeformada inicial é usada em todos os cálculos de deslocamento tipo $\delta\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{F}$.

Apesar de apresentar convergência inferior (mais lenta) que o Newton-Raphson puro, o NRM não necessita do cálculo do operador tangente (Simo e Hughes 1997), (Hellmich 1999), que não foi implementado neste trabalho.



A cada passo de carga (k):
$\delta\boldsymbol{u}^1 = \boldsymbol{K}^{-1}(\boldsymbol{F}_K - \boldsymbol{F}_{K-1})$
$\boldsymbol{u} = \boldsymbol{u} + \delta\boldsymbol{u}^1$
$\boldsymbol{u}$ é integrado, gerando $\boldsymbol{F}_K^1$
resíduo: $\boldsymbol{\Lambda}^1 = \boldsymbol{F}_K - \boldsymbol{F}_K^1$
$\delta\boldsymbol{u}^2 = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^1$
$\boldsymbol{u} = \boldsymbol{u} + \delta\boldsymbol{u}^2$
$\boldsymbol{u}$ é integrado, gerando $\boldsymbol{F}_K^2$
resíduo: $\boldsymbol{\Lambda}^2 = \boldsymbol{F}_K - \boldsymbol{F}_K^2$
$\delta\boldsymbol{u}^3 = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^2$
E assim até que $\|\boldsymbol{\Lambda}^n\| < tol_{\boldsymbol{\Lambda}}$

Figura 3.2: Método iterativo não-linear (NRM) de integração de forças.

A integração da equação constitutiva (2.16) segue um modelo de elastoplasticidade (Simo e Hughes 1997) acoplada (Hellmich e Ulm 1999) com fluência (Sercombe, et al. 2000). Esta integração fornece o atual valor de forças internas/tensões para um dado incremento $\delta\boldsymbol{u}$ no campo de deslocamentos $\boldsymbol{u}$.

3.2.1 Cálculo de tensões

O incremento de deformação total $\Delta\boldsymbol{\varepsilon}$, calculado a partir do incremento de deslocamentos $\delta\boldsymbol{u}$, define o chamado estado tentativa.

$$\boldsymbol{\sigma}_{n+1}^{tent} = \boldsymbol{\sigma}_n + \frac{1}{K_v}\boldsymbol{C}(\xi_{n+1}):\left[\Delta\boldsymbol{\varepsilon}_{n+1} - \mathbf{1}\Delta\boldsymbol{\varepsilon}_{n+1}^{shr} - \frac{1}{1+\frac{\tau_w}{\Delta t}}(J_\infty^v\boldsymbol{G}:\boldsymbol{\sigma}_n - \boldsymbol{\varepsilon}_n^v)\right.$$
$$\left. - \alpha\mathbf{1}\Delta T - \frac{\Delta t}{\eta_f(\Gamma_{n+1})}\boldsymbol{G}:\boldsymbol{\sigma}_n\right] \tag{3.5}$$

onde o fator K_v é relativo à relaxação das tensões pelo efeito da fluência.

$$K_v = 1 + \frac{1}{1+\frac{\tau_w}{\Delta t}}E(\xi_{n+1})J_\infty^v + \Delta t\frac{E(\xi_{n+1})}{\eta_f(\Gamma_{n+1})} \tag{3.6}$$

A tensão tentativa $\boldsymbol{\sigma}_{n+1}^{tent}$ deve ser testada contra as funções de carregamento. Caso o estado de tensões tentativa seja admissível ($F_1 \leq 0$ e $F_2 \leq 0$), o incremento de deslocamentos – deformações - tensões é elástico e está resolvido ($\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{tent}$). Caso seja inadmissível, deve-se prosseguir com o mapeamento de retorno. Este último trata do cálculo do conjunto tensão – deformação plástica – endurecimento que satisfaz às equações de Kuhn-Tucker e de consistência da plasticidade.

Calculado o incremento de deformações plásticas, o estado final de tensões é dado por:

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{tent} - \frac{1}{K_v}\boldsymbol{C}(\xi_{n+1}) : \Delta\boldsymbol{\varepsilon}_{n+1}^{p} \tag{3.7}$$

3.2.2 Fluência

Em (3.5) a viscosidade $\eta_f(\Gamma_{n+1})$ é calculada através das equações (2.29) e (2.30), onde o deslizamento viscoso γ_{n+1} é calculado pelas equações a seguir.

$$\Delta\gamma_{n+1} = \frac{1}{2}\left\{\frac{1 + 2\Gamma_0 H\Delta t\Theta}{\Delta t H^2\Theta} - \sqrt{\Xi}\right\} - \gamma_n \tag{3.8}$$

$$\Xi = \left\{\frac{1 + 2\Gamma_0 H\Delta t\Theta}{\Delta t H^2\Theta}\right\}^2 - 4\frac{\gamma_n + \Delta t\Gamma_0^2\Theta}{\Delta t H^2\Theta} \tag{3.9}$$

Nas equações (3.8) e (3.9), $\Theta = exp\left[-\frac{\mathrm{U}}{\mathrm{R}}\left(\frac{1}{T_{n+1}} - \frac{1}{T_0}\right)\right]$. Após calculado o estado final de tensões $\boldsymbol{\sigma}_{n+1}$, o incremento de deformações de fluência de curto prazo deve ser calculado (equação (2.27)) e armazenado.

3.2.3 Ativação das superfícies de plastificação

Dividiremos o mapeamento de retorno em três situações, de acordo com as regiões definidas no seguinte plano de invariantes de tensão (Figura 3.3).

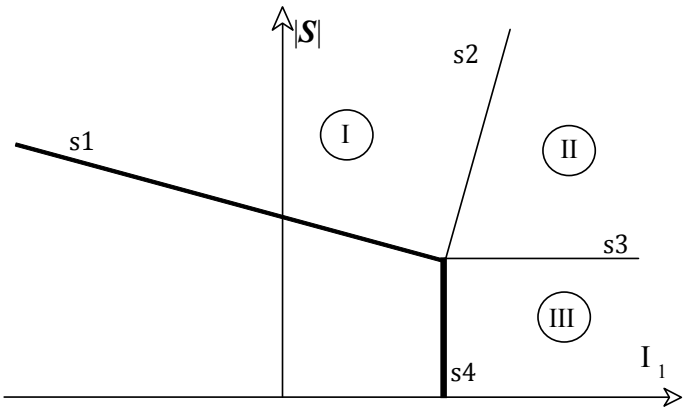


Figura 3.3: Regiões usadas no mapeamento de retorno.

A superfície apontada como s1 corresponde a uma superfície de revolução (tronco de cone) associada ao critério Drucker-Prager (DP). A superfície s4 corresponde ao critério "Tension Cut-Off" (formato de disco). As superfícies s2 e s3 são normais respectivamente a s1 e s4 na linha de interseção entre estas. O espaço fora de s1 e à esquerda de s2 define a região I, na qual apenas o critério DP é considerado ativo. O espaço à direita de s4 e dentro de s3 define a região III onde apenas o critério Tension Cut-Off é considerado ativo. O espaço à direita de s2 e fora de s3 define a região II onde ambas superfícies são ativadas. A Tabela 3.1 mostra as equações das retas correspondentes à interseção das respectivas superfícies com o plano $|\boldsymbol{S}|\ vs\ I_1$,.

Tabela 3.1: Equações das retas no plano $|\boldsymbol{S}|\ vs\ I_1$.

r1 (s1)	$\|\boldsymbol{S}\| = K - \alpha I_1$
r2 (s2)	$\|\boldsymbol{S}\| = I_1/\alpha + K - 3f_t(\alpha + 1/\alpha)$
r3 (s3)	$\|\boldsymbol{S}\| = K - 3\alpha f_t$
r4 (s4)	$I_1 = 3f_t$

Separando tensão de propriedades temos um conjunto de parâmetros (Tabela 3.2) e testes (Tabela 3.3) para determinação da ativação das superfícies de plastificação.

Tabela 3.2: Parâmetros de ativação.

$\mu_1 = \lvert\boldsymbol{S}\rvert + \alpha I_1$	$\phi_1 = K$
$\mu_{12} = \lvert\boldsymbol{S}\rvert - I_1/\alpha$	$\phi_{12} = K - 3f_t(\alpha + 1/\alpha)$
$\mu_{23} = \lvert\boldsymbol{S}\rvert$	$\phi_{23} = K - 3\alpha f_t$
$\mu_3 = I_1$	$\phi_3 = 3f_t$

Tabela 3.3: Testes de ativação.

$\mu_1 > \phi_1$ e $\mu_{12} \geq \phi_{12}$	região I ativada
$\mu_{12} < \phi_{12}$ e $\mu_{23} > \phi_{23}$	região II ativada
$\mu_{23} \leq \phi_{23}$ e $\mu_3 > \phi_3$	região III ativada

3.2.4 Mapeamento de retorno às superfícies

3.2.4.1Região I

Para resolver o problema de mapeamento de retorno à superfície DP com endurecimento, um algoritmo do tipo "cutting plane" (Simo e Hughes 1997) é adaptado ao problema acoplado (Ferreira 1998), gerando as etapas de cálculo seguintes.

$$d\lambda_1 = \frac{F_{1,n+1}^{(k)}}{\bar{A} + \partial_{\boldsymbol{\sigma}} f_1^{(k)} : \boldsymbol{C} : \partial_{\boldsymbol{\sigma}} f_1^{(k)}} \qquad d\boldsymbol{\sigma} = -d\lambda_1 \boldsymbol{C} : \partial_{\boldsymbol{\sigma}} f_1^{(k)} \qquad (3.10)\text{a,b}$$

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + d\boldsymbol{\sigma} \qquad \Delta\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k+1)} - \boldsymbol{\sigma}_n \qquad (3.11)\text{a,b}$$

$$\Delta\boldsymbol{\varepsilon}_{n+1}^{e,(k+1)} = \boldsymbol{C}^{-1} : \boldsymbol{\sigma}_{n+1}^{(k+1)} \qquad \Delta\boldsymbol{\varepsilon}_{n+1}^{p,(k+1)} = \Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}_{n+1}^{e,(k+1)} \qquad (3.12)\text{a,b}$$

$$\Delta\lambda_1^{(k+1)} = \frac{\partial_{\boldsymbol{\sigma}} f_1^{(k+1)} : \Delta\boldsymbol{\varepsilon}_{n+1}^{p,(k+1)}}{\partial_{\boldsymbol{\sigma}} f_1^{(k)} : \partial_{\boldsymbol{\sigma}} f_1^{(k)}} \qquad \Delta\chi_1^{(k+1)} = 3\alpha\Delta\lambda_1^{(k+1)} \qquad (3.13)\text{a,b}$$

$$\chi_{1,n+1} = \chi_{1,n} + \Delta\chi_1^{(k+1)} \qquad F_{1,n+1}^{(k+1)} = f_1\left(\boldsymbol{\sigma}_{n+1}^{(k+1)}\right) - K\left(\xi_{n+1}, \chi_{1,n+1}\right) \qquad (3.14)\text{a,b}$$

As equações acima descrevem uma iteração do método de mapeamento adotado. No passo final o critério é testado contra uma tolerância, o que desvia a execução para nova iteração, considera resolvido o mapeamento ou acusa divergência no processo numérico. No caso de divergência a execução é finalizada com uma mensagem de erro.

3.2.4.2Região II

Apesar da região II envolver o critério DP, não consideramos o endurecimento na solução do mapeamento nesta região, o que resulta em um processo direto (em oposição ao processo iterativo da região I). Realizamos então duas projeções: uma paralela a r2 até atingir r3 e então uma projeção sobre r3 até atingir o vértice.

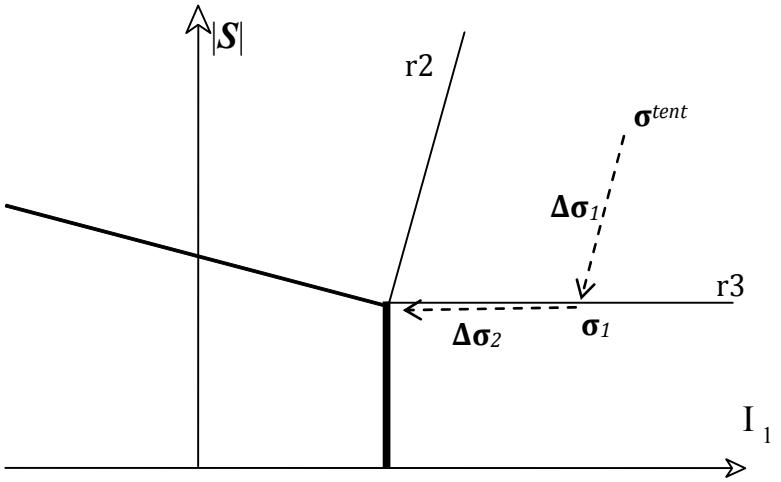


Figura 3.4: Mapeamento de retorno da região II.

A primeira projeção tem a direção e sentido inverso do gradiente da superfície DP, enquanto a segunda tem a direção do eixo hidrostático, sentido de compressão. Logo a primeira projeção é dada por:

$$\Delta\boldsymbol{\sigma}_1 = \mu\partial_{\boldsymbol{\sigma}} f_1 = \mu\left(\alpha\mathbf{1} + \frac{1}{2|\boldsymbol{S}^{tent}|}\boldsymbol{S}^{tent}\right), \mu = 2(K - 3\alpha f_t - |\boldsymbol{S}^{tent}|) \qquad (3.15)\text{a,b}$$

A segunda projeção é calculada por:

$$\Delta\boldsymbol{\sigma}_2 = -\frac{1}{3}(tr(\boldsymbol{\sigma}_1) - 3f_t)\mathbf{1} \tag{3.16}$$

As variáveis internas de plasticidade são calculadas em função das deformações plásticas como:

$$\Delta\boldsymbol{\varepsilon}_i^p = -\boldsymbol{C}^{-1}{:}\,\Delta\boldsymbol{\sigma}_i \qquad \Delta\chi_i = tr\left(\Delta\boldsymbol{\varepsilon}_i^p\right) \tag{3.17}$$a,b

3.2.4.3Região III

A projeção nesta região segue o esquema da segunda projeção apresentada para a região II.

$$\Delta\boldsymbol{\sigma} = -\frac{1}{3}(tr(\boldsymbol{\sigma}^{tent}) - 3f_t)\mathbf{1} \quad \Delta\boldsymbol{\varepsilon}^p = -\boldsymbol{C}^{-1}{:}\,\Delta\boldsymbol{\sigma} \quad \Delta\chi_2 = tr(\Delta\boldsymbol{\varepsilon}^p) \tag{3.18}$$a,b,c

3.2.5 Validação: plasticidade

A simulação de um teste de tensão uniaxial com controle por deslocamento mostra a resposta da implementação das duas superfícies de plastificação. A Figura 3.5 mostra o gráfico tensão *versus* deformação para um corpo de prova considerando-se apenas plasticidade, sem o efeito de envelhecimento das superfícies causado pela reação de hidratação.

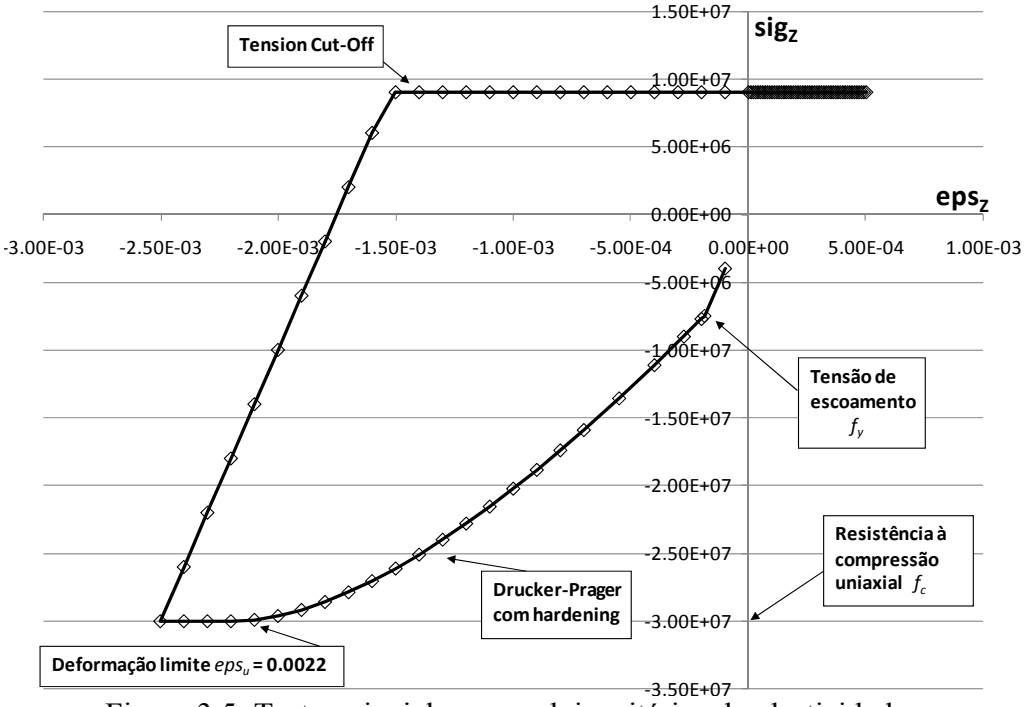


Figura 3.5: Teste uniaxial com os dois critérios de plasticidade.

3.2.6 Validação: fluência

Mostramos a seguir a determinação dos parâmetros de material para dois concretos que foram objeto de estudos em (Fairbairn, et al. 2008). De acordo com o modelo (Sercombe, et al. 2000), a curva da taxa de fluência específica $\frac{dJ}{dt}$ no tempo tem uma assíntota na escala *loglog*. O módulo H é o inverso da inclinação desta assíntota. A fluência específica de longo prazo é obtida pela integração de $\frac{dJ^f}{dt} = \frac{1}{Ht}$. Ao separar os efeitos de curto e longo prazo em J obtém-se o parâmetro J_∞^v. O tempo característico $\tau_{w,\infty}$ corresponde a $\tau_{w,\infty} = t(J^v = 0.63J_\infty^v) - t_0$, onde t_0 é instante do carregamento. Os parâmetros restantes foram calculados pelo retro-ajuste da simulação do teste de compressão uniaxial que gera a curva de fluência específica com os respectivos resultados experimentais. Os valores de parâmetros obtidos estão na tabela abaixo. A Figura 3.6 e a Figura 3.7 mostram as curvas de fluência específica experimental e simulada para dois materiais: um concreto de

referência e uma blenda com substituição de cimento por cinzas de produtos agrícolas (Fairbairn, et al. 2008), referenciado como SCBA/RHA (sugar cane bagasse ash/ rice husk ash). Nas Figuras 3.6-7 o primeiro material é referenciado como ref. e o segundo como cinzas, respectivamente.

Tabela 3.4 Parâmetros do modelo de fluência.

Misturas	$\tau_{w,\infty}$ (dd)	J_∞^v (Pa-1)	Γ_0 (1/Pa.dd)	H (Pa)	U/R (K)	T_0 (oC)
Referência	0.5	13.013E^{-12}	4	0.1	2700	20
SCBA/RHA	0.8	12.288E^{-12}	7	0.12	2700	20

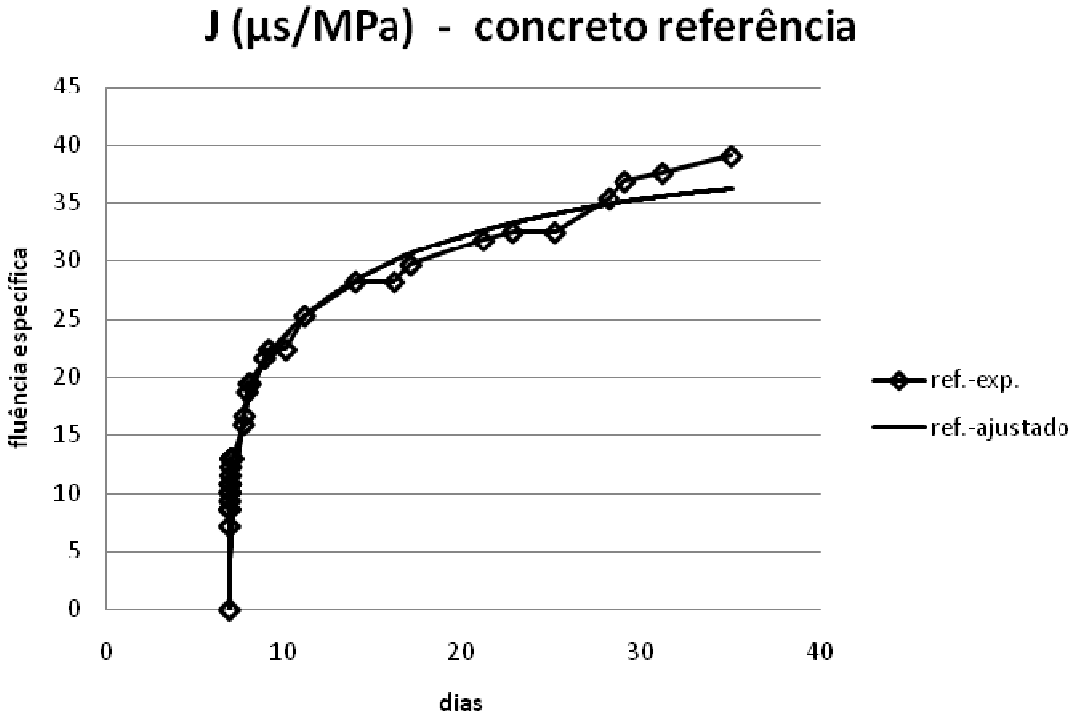


Figura 3.6: Fluência específica para concreto de referência.

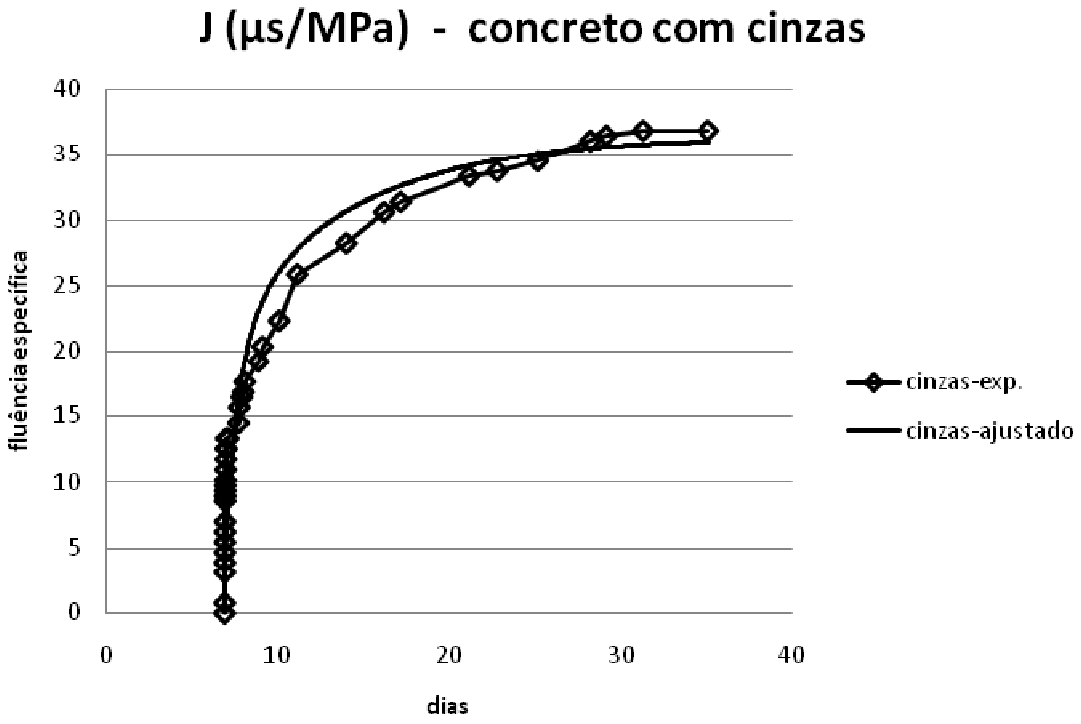


Figura 3.7: Fluência específica para concreto blenda.

3.3 CONSTRUÇÃO POR ETAPAS DE CONCRETAGEM

É fundamental na simulação da construção de usinas hidrelétricas de concreto, como em qualquer outra estrutura de concreto massa, a capacidade de reproduzir o processo de construção por etapas de concretagem. A diferença de idades entre camadas é fonte de vários problemas, pois ela acarreta diferenças de rigidez, resistência e temperaturas. Aliando a isto o efeito de restrição de deslocamento que as camadas exercem umas sobre as outras, tem-se uma potencial configuração geradora de tensões que podem levar à fissuração.

A forma adotada para simular o processo de lançamento das camadas de concretagem é adotar um método de ativação de elementos a partir de uma malha completa. Inicialmente estão ativados apenas os elementos da primeira camada. Ao

tempo certo, determinado por um plano de concretagem, ativam-se os elementos de cada nova camada, uma camada por vez.

A cada camada lançada as matrizes oriundas dos problemas de elementos finitos (termo-químico e mecânico) são remontadas, incluindo os novos elementos (recém-ativados). Todos os campos já calculados para camadas pré-existentes (deslocamentos, tensões, temperaturas , graus de hidratação, etc) são mantidos na nova estrutura de dados para montagem dos sistemas lineares de equações.

Isto pode criar uma inconsistência na interface camada pré-existente/camada recém lançada: os nós desta interface pertencem agora ao mesmo tempo às duas camadas, porém a mais antiga já avançou na hidratação, enquanto a nova começa do zero. Então qual será o grau de hidratação destes nós é uma questão que afetará as propriedades mecânicas nesta região. Entretanto as propriedades mecânicas são normalmente associadas ao elemento e não ao nó.

A solução adotada é considerar o grau de hidratação constante no elemento, enquanto que a temperatura tem seu valor dado por nó da malha. Desta forma se mantém a independência entre os materiais antigo e novo (associados aos elementos), quanto ao estado da reação e conseqüentemente, quanto às propriedades mecânicas. No cálculo do grau de hidratação de um elemento, a temperatura considerada é a média das temperaturas nodais do elemento.

Os valores iniciais de deslocamento e tensão nos elementos recém-lançados são zero. A temperatura inicial é dada também pelo plano de concretagem.

4 IMPLEMENTAÇÃO EM PARALELO

4.1 ESTRUTURA DE DADOS

Na estrutura CSRC a matriz esparsa $A(neqxneq)$ (neq = número de equações) é armazenada em três arranjos: $\{al\}$, $\{ia\}$ e $\{ja\}$, quando simétrica. Caso seja não simétrica, há ainda o arranjo $\{au\}$. Os arranjos reais $\{al\}$ e $\{au\}$ contêm os nad coeficientes não-nulos de A, tomados respectivamente por linha acima da diagonal e por coluna abaixo da diagonal. O arranjo inteiro $\{ia\}$ contém neq+1 ponteiros para o primeiro coeficiente de cada linha em $\{al\}$, que corresponde ao primeiro coeficiente de cada coluna em $\{au\}$. O arranjo inteiro $\{ja\}$ contém nad valores que indicam a coluna dos coeficientes de $\{al\}$ e nad valores que indicam a linha dos coeficientes de $\{au\}$. Por questão de eficiência armazenamos a diagonal de A em um vetor separado $\{ad\}$, isto faz com que a parcela da diagonal saia do laço interno no produto matriz vetor. Com esta estrutura, o produto matriz-vetor é realizado como mostra o quadro do Algoritmo 4.1, quando a matriz A é simétrica, o que é o caso nos dois problemas em questão: elasticidade e equação do calor.

do i = 1, neq	percorre as equações
t = ad(i)*v(i)	parcela da diagonal
vi = v(i)	
do j = ia(i), ia(i+1)-1	percorre linha
k = ja(j)	
s = al(j)	
t = t + s*v(k)	parte inferior da matriz
p(k) = p(k) + s*vi	parte superior da matriz
enddo	
p(i) = t	
enddo	

Algoritmo 4.1: Produto matriz-vetor no CSRC simétrico.

do i = 1, neq	percorre as equações
t = ad(i)*v(i)	parcela da diagonal
vi = v(i)	
do j = ia(i), ia(i+1)-1	percorre linha
k = ja(j)	
t = t + al(j)*v(k)	parte inferior da matriz
p(k) = p(k) + au(j)*vi	parte superior da matriz
enddo	
p(i) = t	
enddo	

Algoritmo 4.2: Produto matriz-vetor no CSRC não-simétrico.

O produto matriz-vetor é a operação de maior custo computacional, que é de $(neq + 4nad)$ operações de ponto flutuante (OPF) em uma iteração do método de gradientes conjugados com pré-condicionador diagonal (sigla em inglês: PCGD), que tem um total de: $4(3neq + nad) + 3$ OPF. É importante notar que nad tem ordem de grandeza maior que neq. O algoritmo do PCGD utilizado é mostrado no Algoritmo 4.3, no qual M é o pré-condicionador diagonal e x_0 é a estimativa inicial.

compute $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$, $p_0 = z_0$
for j = 0, 1, ... until convergence do:
$\alpha_j = (r_j, z_j)/(Ap_j, p_j)$
$x_{j+1} = x_j + \alpha_j p_j$
$r_{j+1} = r_j - \alpha_j Ap_j$
$z_{j+1} = M^{-1}r_{j+1}$
$\beta_j = (r_{j+1}, z_{j+1})/(r_j, z_j)$
$p_{j+1} = z_{j+1} + \beta_j p_j$
enddo

Algoritmo 4.3: Método PCGD (ver (Saad 2003)).

4.2 PARTICIONAMENTO DA MALHA

4.2.1 Particionamento "non-overlapping"

A Figura 4.1 ilustra o método "non-overlapping" de particionamento, que se baseia em uma divisão do domínio original $\boldsymbol{\Omega}$ em p sub-domínios $\boldsymbol{\Omega}^i$ $(i = 0,1, \dots, n-1)$, sendo p o número de processadores, tal que,

$$\boldsymbol{\Omega} = \bigcup_{i=0}^{p-1} \boldsymbol{\Omega}^i \qquad \boldsymbol{\Omega}^i \bigcap \boldsymbol{\Omega}^j = \emptyset, \qquad i \neq j \tag{4.1}$$

A cada subdomínio $\boldsymbol{\Omega}^i$ corresponde uma partição $\boldsymbol{P}^i$ e uma malha $\boldsymbol{M}^i$ com nel^i elementos Ω_k^i e um conjunto $\boldsymbol{V}^i$ contendo os nós V_k^i da partição.

$$\boldsymbol{\Omega}^i = \bigcup_{k=1}^{nel^i} \Omega_k^i \quad \boldsymbol{V}^i = \bigcup_{k=1}^{nnode^i} V_k^i \tag{4.2}$$

Cada nó da malha adotada $\boldsymbol{M}$ em $\boldsymbol{\Omega}$ pertence exclusivamente a uma única partição $\boldsymbol{P}^i$. Portanto, um conjunto $\boldsymbol{V}^i$ dos nós que constituem a partição $\boldsymbol{P}^i$ associada a $\boldsymbol{\Omega}^i$ é formado por três conjuntos de nós: o conjunto $\boldsymbol{V}^{i,1}$, com nós pertencentes ao interior do subdomínio $\boldsymbol{\Omega}^i$ ou ao contorno de $\boldsymbol{\Omega}$, o conjunto $\boldsymbol{V}^{i,2}$ de nós

pertencentes à partição $\boldsymbol{P}^i$ e à interface entre $\boldsymbol{P}^i$ e as demais partições, e o conjunto $\boldsymbol{V}^{i,3}$ de nós na interface entre $\boldsymbol{P}^i$ e as demais partições, mas que não pertencem a $\boldsymbol{P}^i$.

$$\boldsymbol{V}^i = \left\{\boldsymbol{V}^{i,1}, \boldsymbol{V}^{i,2}, \boldsymbol{V}^{i,3}\right\} \tag{4.3}$$

Note-se que, com a definição acima, o conjunto de nós $\boldsymbol{V}$ do domínio inteiro é dado por

$$\boldsymbol{V} = \bigcup_{i=0}^{p-1} \left\{\boldsymbol{V}^{i,1}, \boldsymbol{V}^{i,2}\right\} \tag{4.4}$$

Além disso, o conjunto $\boldsymbol{V}^f$ da interface entre partições é

$$\boldsymbol{V}^f = \bigcup_{i=0}^{p-1} \boldsymbol{V}^{i,2} \tag{4.5}$$

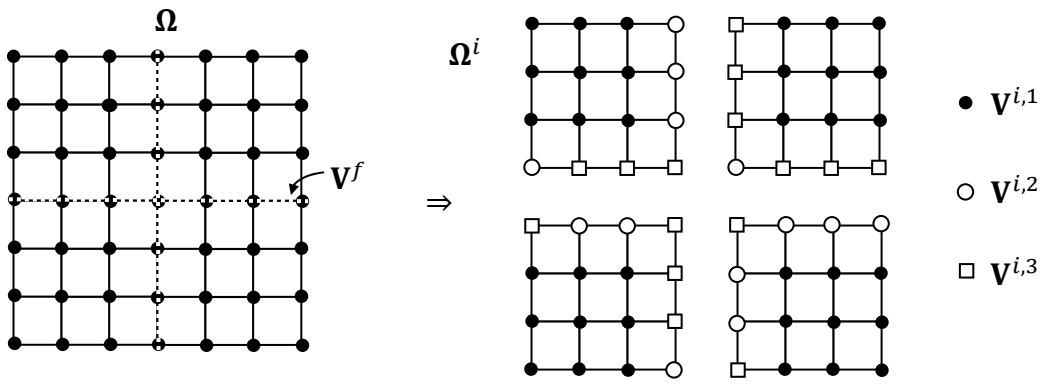


Figura 4.1: Particionamento “Non-overlapping”.

O conjunto $\boldsymbol{V}^f$ tem importância na eficiência da paralelização por está associado à razão entre comunicação e processamento.

4.2.2 Particionamento "overlapping"

Para descrever o método de particionamento overlapping, partimos do particionamento "non-overlapping" e definimos um quarto conjunto de nós $\boldsymbol{V}^{i,4}$ que contém nós de partições vizinhas $\boldsymbol{P}^j$ $(j \neq i)$ e são conectados aos nós $\boldsymbol{V}^{i,2}$ em $\boldsymbol{P}^i$. As partições "overlapping" $\overline{\boldsymbol{P}}^i$ são obtidas pela adição a cada partição "non-overlapping" dos elementos $\overline{\Omega}_k^i = \Omega_l^j$ $(j \neq i)$ contendo nós $V_k^i \in \boldsymbol{V}^{i,2}$, como mostra a Figura 4.2. Desta forma, o conjunto $\overline{\boldsymbol{V}}^i$ de uma partição $\overline{\boldsymbol{P}}^i$ é dado por:

$$\overline{\boldsymbol{V}}^i = \left\{\boldsymbol{V}^{i,1}, \boldsymbol{V}^{i,2}, \boldsymbol{V}^{i,3}, \boldsymbol{V}^{i,4}\right\} \tag{4.6}$$

onde os índices 1, 2, 3 têm o mesmo significado que têm no método "non-overlapping". O conjunto de nós $V_k^i \in \boldsymbol{V}^{i,1}$ no interior de $\overline{\boldsymbol{P}}^i$ que também pertencem a $\boldsymbol{V}^{j,4}$ em qualquer partição vizinha $\overline{\boldsymbol{P}}^j$ é denotado por $\boldsymbol{V}^{i,1a}$, quando referenciado a partir de $\overline{\boldsymbol{P}}^i$.

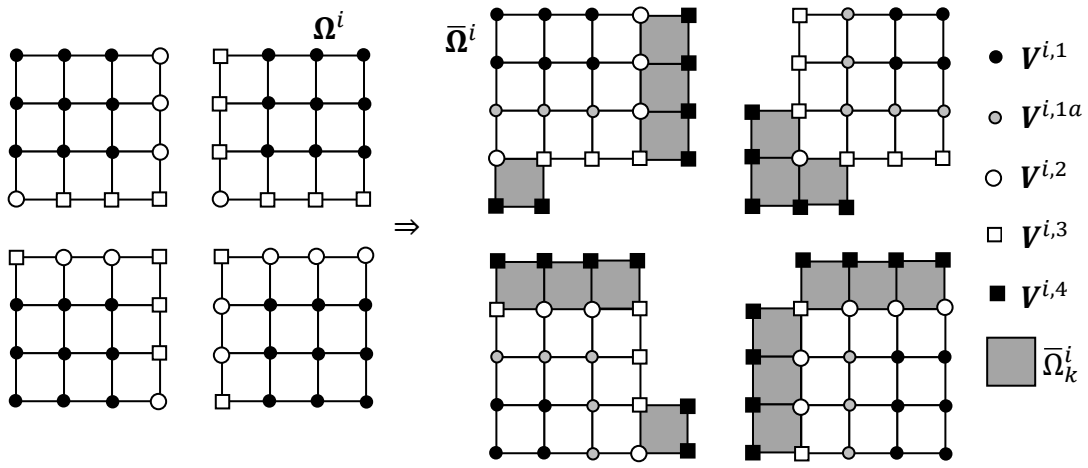


Figura 4.2: Particionamento "Overlapping".

O conjunto $\boldsymbol{V}^f$ de nós na interface entre partições neste método é

$$\boldsymbol{V}^f = \bigcup_{i=0}^{p-1} \left(\boldsymbol{V}^{i,1a} + \boldsymbol{V}^{i,2} \right) \tag{4.7}$$

A cada conjunto de nós corresponde um conjunto de equações e conseqüentemente um número de equações. Denotaremos por neq^f o número de equações correspondente a $\boldsymbol{V}^f$.

4.3 SOLUÇÃO EM PARALELO

A solução de um problema de elementos finitos pode ser separada em duas grandes operações: 1 - montagem de matrizes e vetores, e 2 – solução do sistema linear. A cada processador é atribuída uma partição, e cada processador trabalha apenas com os elementos que constituem esta partição.

Na fase de montagem, as matrizes e vetores são calculados por elemento e então espalhados (montados) na matriz e no vetor do problema. Nesta fase, o método "overlapping" implica em redundância, pois os elementos em contato com $\boldsymbol{V}^f$ são montados em pelo menos dois processadores.

Na fase de solução, cada processo opera sobre sua porção local $A^i x^i = b^i$ do sistema linear global $Ax = b$. Apresentamos a seguir a forma deste sistema local em cada método de sub-divisão de domínio e as implicações desta forma nas operações típicas de um método iterativo de solução de sistema linear.

4.3.1 Método "non-overlapping"

No método "non-overlapping", o sistema local de equações $A^i x^i = b^i$ obtido em cada partição $\boldsymbol{\Omega}^i$ é formado das seguintes partes:

$$A^i = \left[A^i_{jk}\right] \quad x^i = \left[x^{i,k}\right] \quad b^i = \left[b^{i,j}\right] \quad (j,k = 1..3) \tag{4.8}$$

Cada partição $\boldsymbol{\Omega}^i$ possui sua própria estrutura CSR ($\{ia^i\}, \{ja^i\}, \{a^i\}, \{ad^i\}$), baseada no grafo local, cuja numeração de equações segue a classificação dos tipos de nós $\{\boldsymbol{V}^{i,1}, \boldsymbol{V}^{i,2}, \boldsymbol{V}^{i,3}\}$, nesta ordem. A ausência dos elementos de superposição faz com que alguns coeficientes sejam montados parcialmente (A^i_{jk} $(j,k = 2,3)$). Similarmente, os coeficientes de vetores são montados globalmente apenas para os graus de liberdade dos nós em $\boldsymbol{V}^{i,1}$. As três operações vetoriais básicas em PCGD são o produto matriz-vetor ($matvec$), a soma de vetores e o produto interno de vetores (dot). A operação sobre vetores de maior peso computacional entre estas três é o produto matriz-vetor, que localmente tem a forma $p^i = A^i v^i$.

4.3.1.1 Produto matriz-vetor non-overlapping

No caso non-overlapping, o produto matriz-vetor local $p^i = A^i v^i$ tem a forma:

$$p^{i,j} = A^i_{jk} v^{i,k} \quad (j,k = 1..3) \tag{4.9}$$

No vetor produto p^i os valores $p^{i,2}$ e $p^{i,3}$ resultam parciais (após a realização do produto). É então necessário completar estes valores para poder seguir com o processo iterativo. Os resultados $p^{i,2}$ e $p^{i,3}$ de partições vizinhas se complementam.

4.3.1.2 Comunicação non-overlapping

A comunicação (troca) destes valores entre os processos é realizada pelas instruções de comunicação ponto a ponto imediatas mpi_isend e mpi_irecv, conforme Algoritmo 4.4.

copia de $\{p^{i,2}, p^{i,3}\}$ para f_1^i	mapa($p^i \rightarrow f_1^i$)
laço nos vizinhos $k = 1..nviz^i$	
$mpi_irecv(de: \text{viz}^i(k),\ para\ f_2^i)$	mapas($\text{viz}^i(k) \rightarrow f_2^i$)
$mpi_isend(para: \text{viz}^i(k),\ de\ f_1^i))$	mapas($f_1^i \rightarrow \text{viz}^i(k)$)
fim laço	
$mpi_waitall(recv)$	
copia de f_2^i para $\{p^{i,2}, p^{i,3}\}$, **somando**	mapa($f_2^i \rightarrow p^i$)
$mpi_waitall(send)$	

Algoritmo 4.4: Rotina de comunicação "non-overlapping".

Neste processo, cada partição envia uma parte do seu vetor local p^i, no caso $\{p^{i,2}, p^{i,3}\}$, apenas aos $nviz^i$ processos adjacentes a $\boldsymbol{\Omega}^i$ que compartilhem dos nós correspondentes $\{V^{i,2}, V^{i,3}\}$. No código, um mapa de equações local-fronteira por vizinho (mapa($p^i \rightarrow f_1^i$)) especifica quais coeficientes do vetor local p^i são enviadas a cada vizinho de Ω^i por meio de f_1^i. Este mapa é usado na cópia de valores do vetor local p^i para o vetor de comunicação f_1^i. Um mapa análogo de equações local-fronteira por vizinho (mapa($f_2^i \rightarrow p^i$)) especifica quais coeficientes do vetor local p^i são recebidos de cada vizinho de $\boldsymbol{\Omega}^i$ em f_2^i. Nestes mapas os coeficientes são ordenados por vizinho (ver figuras a seguir).

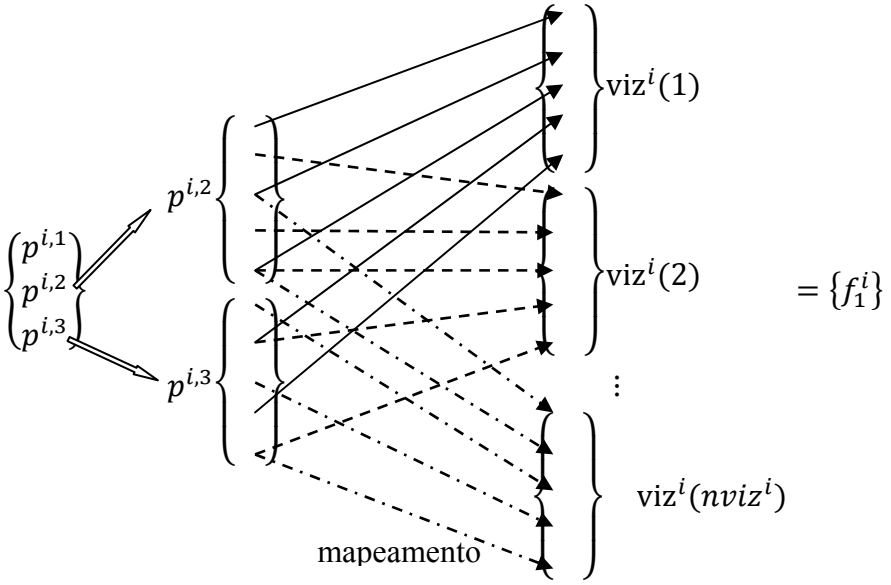


Figura 4.3: Mapeamento de equações local -> fronteira.

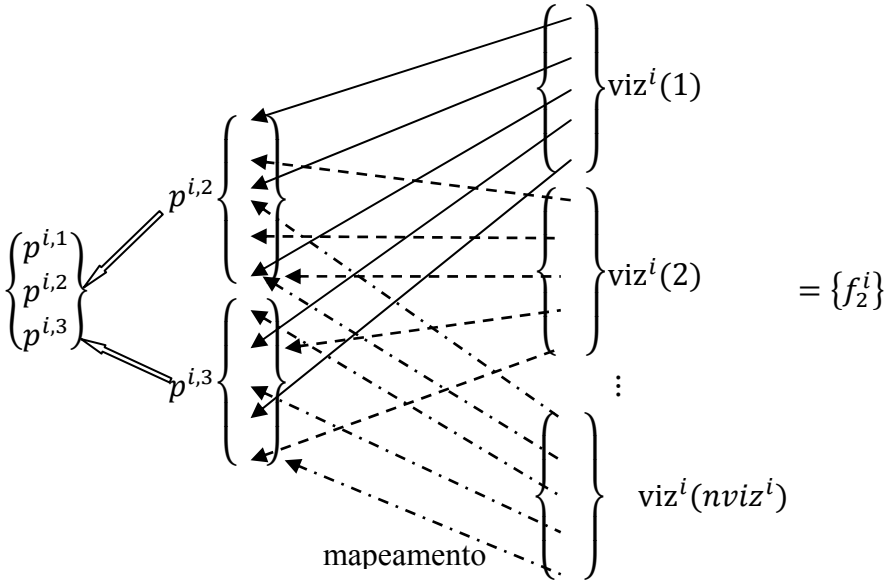


Figura 4.4: Mapeamento de equações fronteira -> local.

Outros arranjos auxiliares (mapas($f_1^i \rightarrow viz$) e mapas($viz \rightarrow f_2^i$)) controlam quais porções de f_1^i devem ser enviadas a cada vizinho e onde (em f_2^i) escrever os valores recebidos de cada vizinho. A filtragem de valores locais para um vetor de

comunicação e vice-versa permite que se comuniquem apenas os coeficientes necessários em cada direção, uma vez que estes coeficientes estão espalhados no vetor local.

A comunicação imediata (*mpi_isend*, *mpi_irecv*) não bloqueia a execução (Grama, et al. 2003), i.e. o controle de execução passa ao próximo comando independentemente da completude do comando de comunicação (envio-> recebimento-> confirmação de recebimento). Desta forma todas as instâncias de comunicação são disparadas de forma independente e realizadas de forma simultânea. Esta simultaneidade faz a comunicação imediata ser mais rápida que a comunicação padrão (*mpi_send*, *mpi_recv*) na qual se cria uma fila de comunicação onde um comando só pode ser iniciado após o término do anterior.

A comunicação ponto a ponto é a base sobre a qual as rotinas de comunicação coletiva (eg. *mpi_allreduce*, *mpi_allgatherv*, *mpi_gatherv*) são construídas. A comunicação coletiva envolve obrigatoriamente todos os processos em um comunicador. Esta característica é desvantajosa quando não há necessidade de troca de informações entre todos os processos. Dependendo do particionamento pode ocorrer ainda que a comunicação entre dois vizinhos seja unidirecional. A melhor alternativa em comunicação coletiva testada ao longo do desenvolvimento (Ferreira, Ribeiro, et al. 2004), (Ribeiro, Ferreira e Fairbairn 2005) e (Ferreira, Fairbairn, et al., A coupled numerical model for concrete parallelized with MPI 2006) é o *mpi_allgatherv*, em que cada processo envia uma cópia de um mesmo vetor a todos os outros e conseqüentemente recebe uma cópia dos vetores de todos os outros processos. Por envolver obrigatoriamente todos os processos e

não permitir comunicação mínima ou unidirecional entre processos a comunicação coletiva é, via de regra, mais lenta que a comunicação ponto a ponto imediata.

4.3.1.3 Soma de vetores non-overlapping

A soma de vetores deve ser feita sobre toda a extensão dos vetores locais, pois todas estas equações são resolvidas em cada processo.

$$v^{i,k} = p^{i,k} + \alpha u^{i,k} \quad (k = 1..3) \tag{4.10}$$

4.3.1.4 Produto escalar non-overlapping

No produto escalar, apenas os coeficientes associados a $\boldsymbol{V}^{i,1}$ e $\boldsymbol{V}^{i,2}$ devem ser computados no processo, evitando erro por redundância.

$$dot^i = u^i \cdot v^i = u^{i,1} \cdot v^{i,1} + u^{i,2} \cdot v^{i,2} \tag{4.11}$$

Assim sendo, os produtos locais devem ser somados, o que foi realizado com a instrução coletiva $mpi_allreduce$.

$$dot = \sum_{i=0}^{p-1} dot^i \tag{4.12}$$

Por fim, a comunicação entre processos realizada na fase de solução se resume ao conjunto de $mpi_irecv's$ - $mpi_isend's$ após cada $matvec$ e ao $mpi_allreduce$ após cada dot.

4.3.2 Método "overlapping"

No método com superposição, o sistema local de equações na partição $\overline{\boldsymbol{\Omega}}^i$ é:

$$A^i = \left[A^i_{jk}\right] \quad x^i = \left[x^{i,k}\right] \quad b^i = \left[b^{i,j}\right] \quad (j = 1..2), (k = 1..4) \tag{4.13}$$

A matriz retangular A^i é montada sem perda de informação a partir da malha local que inclui todos os nós em $\overline{\mathbf{V}}^i$. No entanto, apenas as equações relativas a $\mathbf{V}^{i,1}$ e $\mathbf{V}^{i,2}$ são montadas em $\overline{\boldsymbol{\Omega}}^i$. Observando a Figura 4.2 nota-se que a sub-matriz $A^i_{1,4} = 0$ pois não há arestas ligando diretamente $\mathbf{V}^{i,1}$ a $\mathbf{V}^{i,4}$. Como no método anterior a numeração de equações segue a classificação dos tipos de nós, neste caso: $\left\{\boldsymbol{V}^{i,1} - \boldsymbol{V}^{i,1a}, \boldsymbol{V}^{i,1a}, \boldsymbol{V}^{i,2}, \boldsymbol{V}^{i,3}, \boldsymbol{V}^{i,4}\right\}$. Pelo fato de o sistema local ter o formato retangular, o produto matriz-vetor CSRC assume neste caso a forma especial CSRCR (compressed sparse row/column+rectangular). o armazenamento e o produto da parte quadrada da matriz $\left[\mathrm{A}^{\mathrm{i}}_{\mathrm{jk}}\right] (j, k = 1..2)$ correm conforme já mencionado nos Algoritmos 4.1 e 4.2. A parte retangular da matriz $\left[\mathrm{A}^{\mathrm{i}}_{\mathrm{jk}}\right] (j = 1..2), (k = 3..4)$ é armazenada em três arranjos extras $\{ar\}$, $\{ja1\}$ e $\{ia1\}$, com respectivamente $nad1^i$, $nad1^i$ e neq^i posições, onde $nad1^i$ é o número de coeficientes não nulos em $\{ar\}$. O produto da parte retangular é feito em um segundo loop sobre as equações, de acordo com o Algoritmo 4.5.

do i = 1, neq	percorre as equações
do j = ia1(i), ia1(i+1)-1	percorre linha
k = ja1(j)	
p(i) = p(i) + ar(j)*v(k)	parte retangular da matriz
enddo	
enddo	

Algoritmo 4.5: Produto matriz-vetor no CSRCR, parte retangular.

4.3.2.1 Produto matriz-vetor overlapping

O produto matriz-vetor local $p^i = A^i v^i$ no método overlapping tem a forma:

$$p^{i,j} = A^i_{jk} v^{i,k} \quad (j = 1..2), (k = 1..4) \tag{4.14}$$

Para realizar apropriadamente este produto é necessário obter os coeficientes relativos a $\mathbf{V}^{i,3}$ e $\mathbf{V}^{i,4}$ em $v^{i,k}$ a partir de suas partições de origem antes da realização do produto.

4.3.2.2 Comunicação overlapping

Como no método anterior, esta comunicação é realizada pelo uso das instruções de comunicação ponto a ponto imediatas mpi_isend e mpi_irecv. Porém agora, cada partição envia um vetor contendo componentes de $\{v^{i,1a}, v^{i,2}\}$ aos seus vizinhos. Após recebido o vetor correspondente de cada vizinho, o processo copia a partir deles para suas posições em $\{v^{i,3}, v^{i,4}\}$, sobrescrevendo os valores antigos.

copia de $\{v^{i,1a}, v^{i,2}\}$ para f_1^i	mapa($v^i \rightarrow f_1^i$)
laço nos vizinhos $k = 1..nviz^i$	
$mpi_irecv(de: \text{viz}^i(k),\ para\ f_2^i)$	mapas($\text{viz}^i(k) \rightarrow f_2^i$)
$mpi_isend(para: \text{viz}^i(k),\ de\ f_1^i)$	mapas($f_1^i \rightarrow \text{viz}^i(k)$)
fim laço	
$mpi_waitall(recv)$	
copia de f_2^i para $\{v^{i,3}, v^{i,4}\}$, **sobrecrevendo**	mapa($f_2^i \rightarrow v^i$)
$mpi_waitall(send)$	

Algoritmo 4.6: Rotina de comunicação "overlapping".

Os mapeamentos de coeficientes seguem de forma similar aos do método anterior, trocando-se os $\{p^{i,2}, p^{i,3}\}$ por $\{v^{i,1a}, v^{i,2}\}$ no envio e por $\{v^{i,3}, v^{i,4}\}$ no recebimento. No método non-overlapping os mapas local - fronteira e fronteira – local são idênticos, i.e. o conjunto de coeficientes enviado a um vizinho é o mesmo recebido. No método overlapping os conjuntos de coeficientes enviado e recebido de um

vizinho são disjuntos, i.e. não têm qualquer interseção. Neste método a comunicação entre dois processos pode até ser unidirecional, dependendo do particionamento, o que não ocorre no método non-overlapping, no qual a comunicação entre dois processos é obrigatoriamente bidirecional.

4.3.2.3 Soma de vetores overlapping

A soma de vetores é realizada apenas sobre os coeficientes associados a $\boldsymbol{V}^{i,1}$ e $\boldsymbol{V}^{i,2}$.

$$v^{i,k} = p^{i,k} + \alpha u^{i,k} \quad (k = 1..2) \tag{4.15}$$

4.3.2.4 Produto escalar overlapping

O produto escalar tem a mesma forma do método anterior.

$$dot^i = u^i \cdot v^i = u^{i,1} \cdot v^{i,1} + u^{i,2} \cdot v^{i,2} \tag{4.16}$$

$$dot = \sum_{i=0}^{p-1} dot^i \tag{4.17}$$

A comunicação resultante na fase de solução é então restrita àquela antes de cada $matvec$ e ao $mpi_allreduce$ após cada dot.

4.4 COMPARAÇÃO ENTRE MÉTODOS DE PARTICIONAMENTO

Entre as formas de avaliar a qualidade de uma implementação paralela (Dongarra 2005),(Grama, et al. 2003) a principal é o "*speedup*". Esta medida de desempenho traduz aproximadamente quantas vezes o código ficou mais rápido com a paralelização. Sua definição é:

$$S = \frac{t_s}{t_p} \tag{4.18}$$

Em (4.18) t_s é o tempo de execução seqüencial e t_p é o tempo de execução em paralelo. O "*speedup*" depende da tomada de tempos durante a execução do programa. Outra medida importante é a de Redundância:

$$R = \frac{nopf_p - nopf_s}{nopf_s}\% \tag{4.19}$$

Em (4.19) $nopf_p$ é o número de OPFs da execução em paralelo e $nopf_s$ é o número de OPFs da execução em seqüencial. A Redundância é calculada em duas etapas. A primeira é independente da execução código no computador, pois consiste na análise do código fonte e contagem de suas operações de ponto flutuante. A segunda depende da execução do código na solução de um determinado problema. Dado um particionamento, tem-se os $neqs$ e $nads$ de cada partição. Da execução do código tem-se o número de iterações no PCGD. Com estes dados é possível calcular a redundância na solução por PCGD de um determinado problema.

Tabela 4.1: Contagem *a priori* de OPFs no solver PCGD.

sequencial	$8(\quad neq + nad \quad)+3+nit\{4(3neq+nad)+3\}$
non-overlapping	$8(neq^{i,1}+neq^{i,2}+nad^i)+3+nit\{4[3(neq^{i,1}+neq^{i,2})+ nad^i]+3+8neq^{i,3}+neq^f\}+4neq^{i,3}+2neq^f$
overlapping	$8(neq^{i,1}+neq^{i,2}+nad^i)+3+nit\{4[3(neq^{i,1}+neq^{i,2})+ nad^i]+3+2nad1^i\}+4nad1^i$

Na Tabela 4.1 nit indica o número de iterações necessárias a uma determinada solução do PCGD.

Mostramos a seguir resultados obtidos em um cluster de PCs com 32 máquinas (nós) bi-processadas (porcessador Intel core2 duo, 6MB cache). O exemplo escolhido é o problema de cinco poços, (ver (Ribeiro e Ferreira 2007)), que usa a equação de transporte para simular o processo de injeção e produção em um reservatório. Inicialmente calcula-se o campo de pressões e velocidades (a partir da derivada do campo de pressões), depois estes campos são usados na solução do problema de transporte propriamente. Nesta análise resolvemos apenas o problema da pressão (simétrico, logo usando PCGD). O domínio considerado é um quadrado, no plano, com uma profundidade na vertical. Na linha sob um dos vértices do quadrado existe uma fonte e na aresta oposta existe um sumidouro. A Tabela 4.2 mostra dados do problema. A Tabela 4.3 mostra valores de redundância no solver. As Figuras 4.6 a 4.14 mostram curvas de speedup.

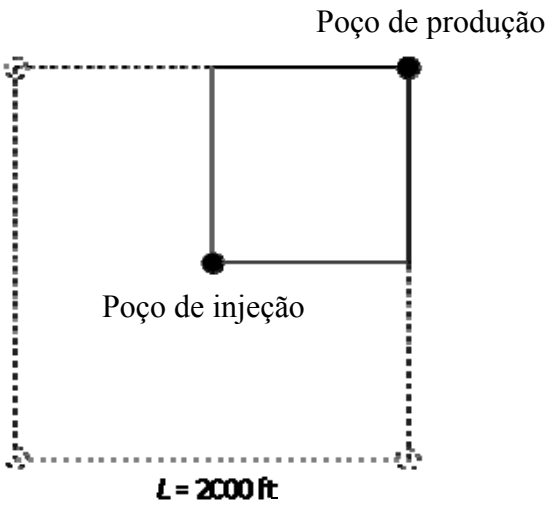


Figura 4.5: Problema dos poços de injeção e produção.

Tabela 4.2: Dados do problema.

	hexahedros	nós	Equações
malha 1	40.000	44.217	44.183
malha 2	200.000	214.221	214.179
malha 3	1.000.000	1.050.426	1.050.374

Tabela 4.3: Redundância no PCGD (%).

Redundância	numprocs	1	2	4	8
non-overlapp.	malha 1	0.00	0.94	1.98	4.24
	malha 2	0.00	0.47	1.02	2.21
	malha 3	0.00	0.21	0.62	1.11

A redundância no PCGD do método non-overlapping provém da divisão das operações $A_{jk}^{i}x^{i,k}$ (j,k=2,3) entre processos devido à montagem parcial dos coeficientes A_{jk}^{i} (j,k=2,3). Em contraponto, o método overlapping é redundante na montagem da matriz local $A_{2,4}^{i}$**.**

O comportamento do speedup varia com o tamanho da malha. A montagem de elementos é uma operação propícia à paralelização por prescindir de comunicação entre processos. A solução do sistema de equações (solver), como visto, requer comunicação e portanto tem ganhos menores com a paralelização. No total da análise o speedup é dominado pelo solver, por esta operação ter maior custo computacional que a montagem de elementos. Em um problema transiente, onde o solver é executado várias vezes após cada montagem de elementos, este domínio deve ser ainda mais acentuado.

Nas Figuras 4.6-14 a seguir a seguinte notação é usada: o radical ***elmt*** indica operação de montagem de elementos; o radical ***solver*** indica solução do sistema linear pelo PCGD; o radical ***total*** indica o conjunto de operações envolvendo montagem de elementos, solver PCGD e demais operações auxiliares ao controle da execução; ***ideal*** indica o speedup correspondente a eficiência de 100%; o sufixo ***c*1** indica core1 ou um processo por máquina; o sufixo ***c*2** indica core2 ou dois

processos por máquina; o sufixo ***n*** indica particionamento non-overlapping e o sufixo ***o*** indica particionamento overlapping.

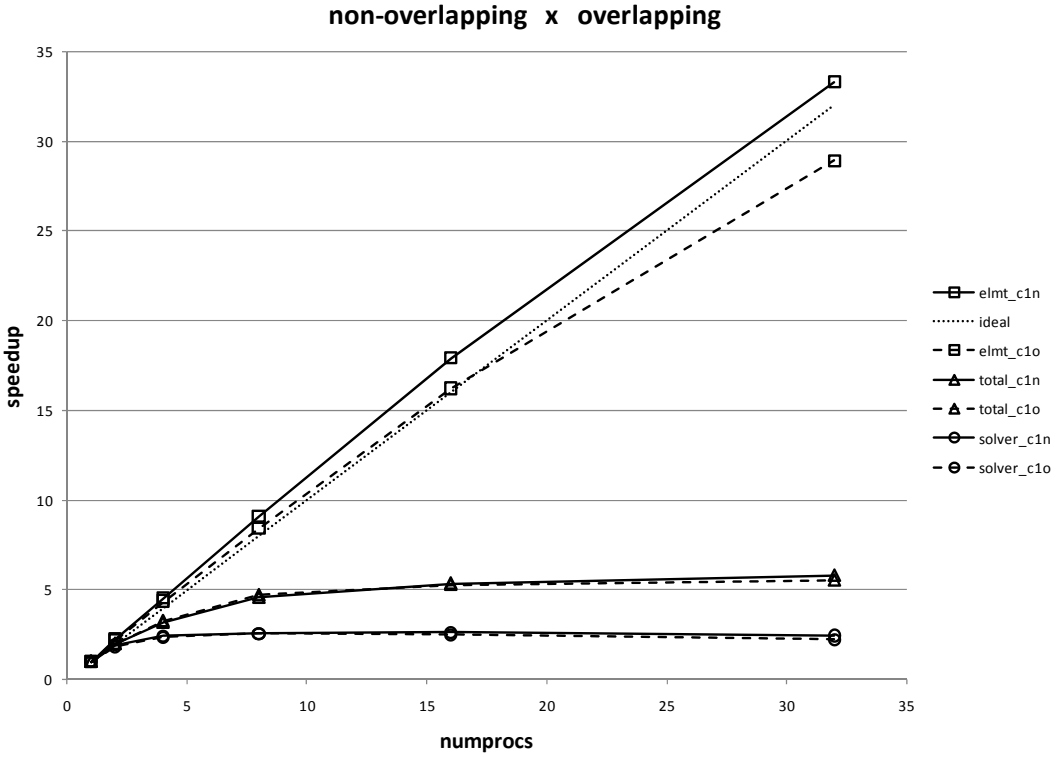


Figura 4.6: Speedups malha 1: non-overlapping x overlapping.

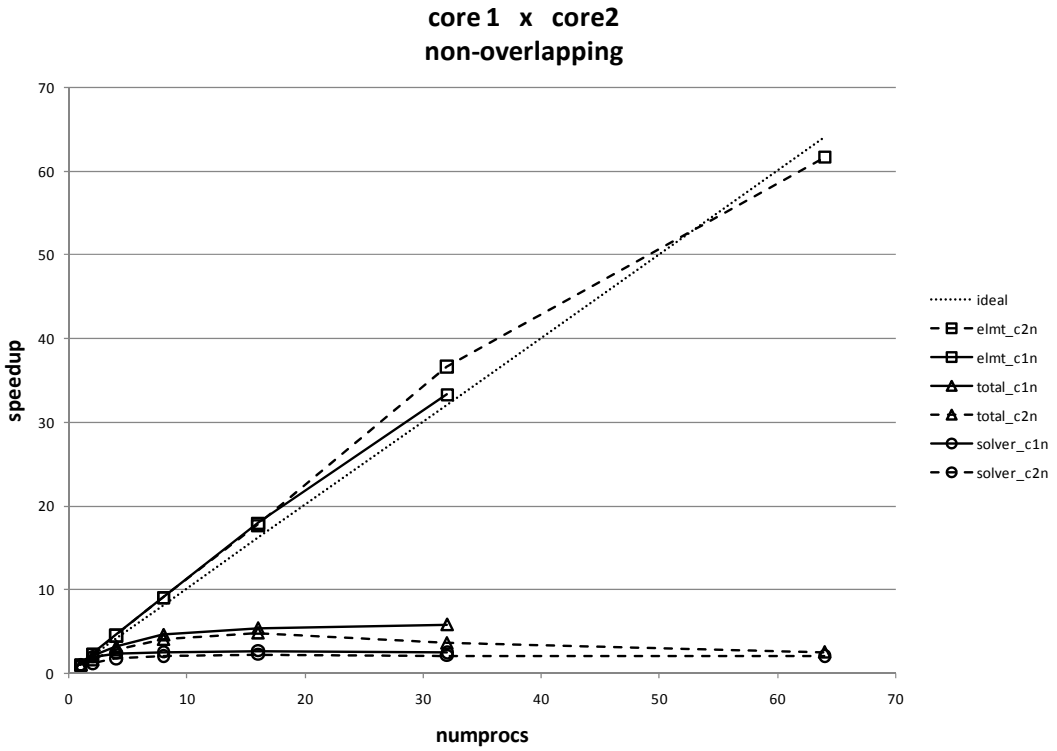


Figura 4.7: Speedups malha 1: core1 x core2, por processo.

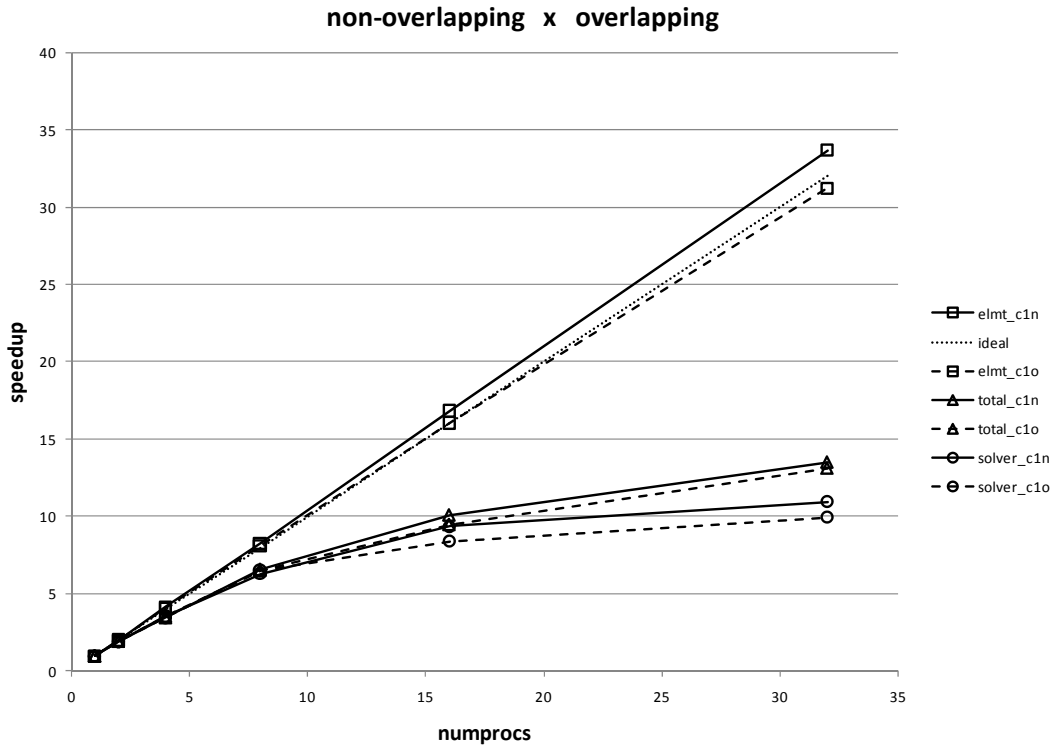


Figura 4.8: Speedups malha 2: non-overlapping x overlapping.

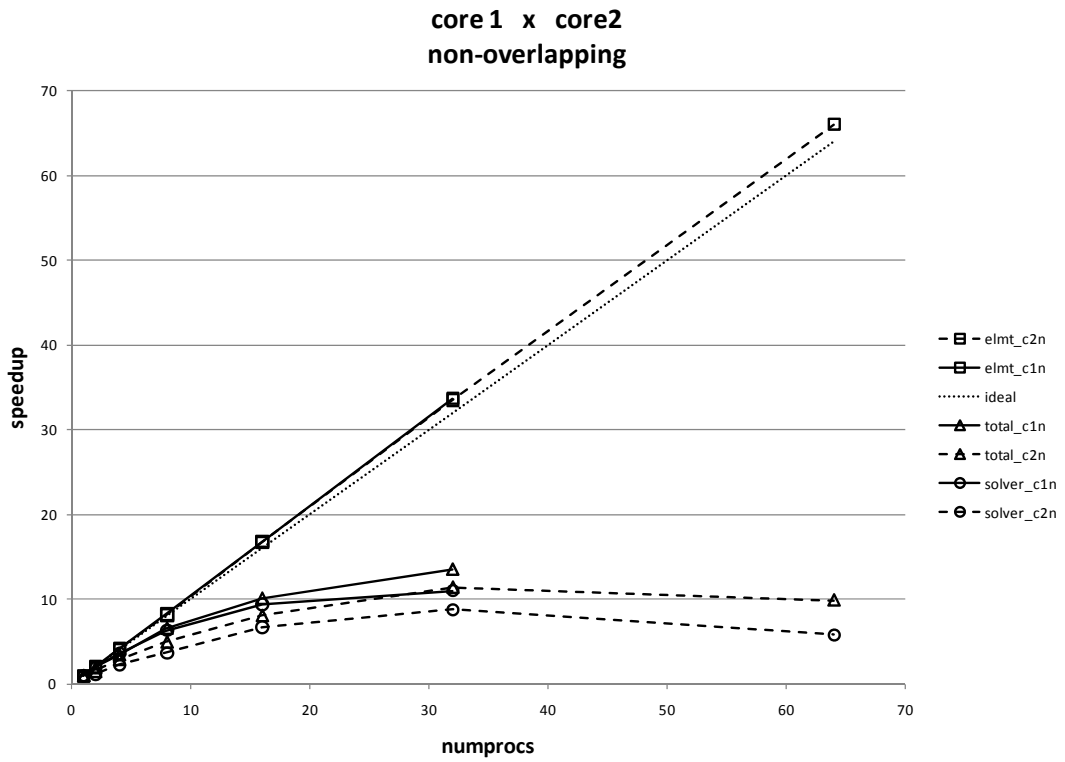


Figura 4.9: Speedups malha 2: core1 x core2, por processo.

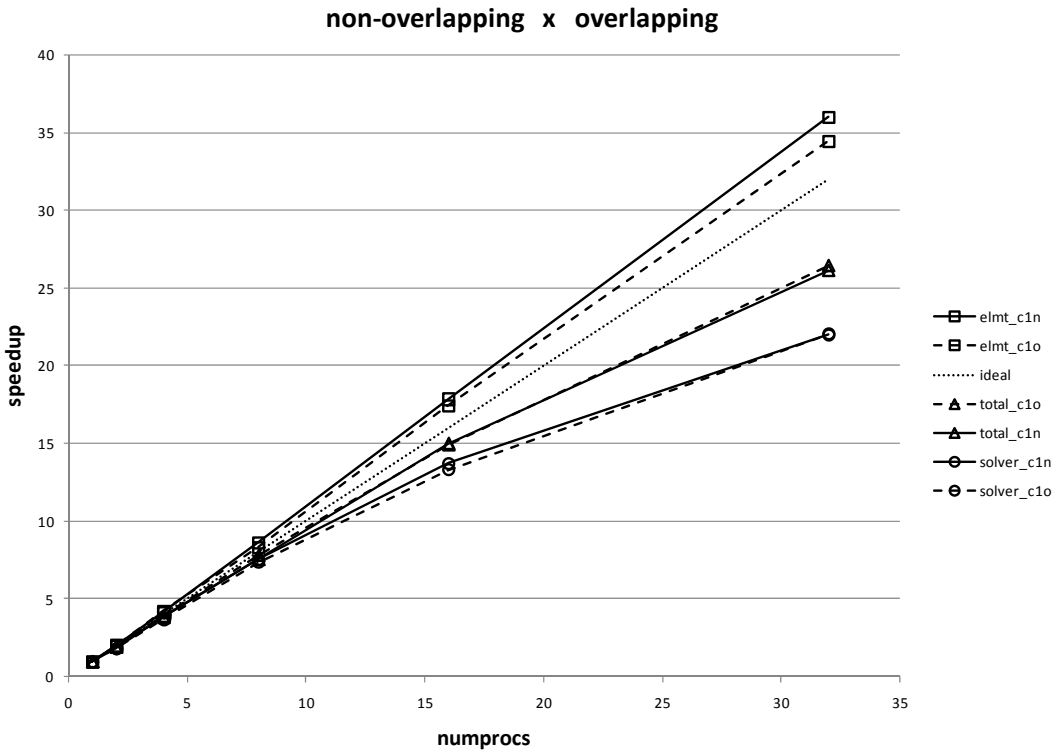


Figura 4.10: Speedups malha 3: non-overlapping x overlapping.

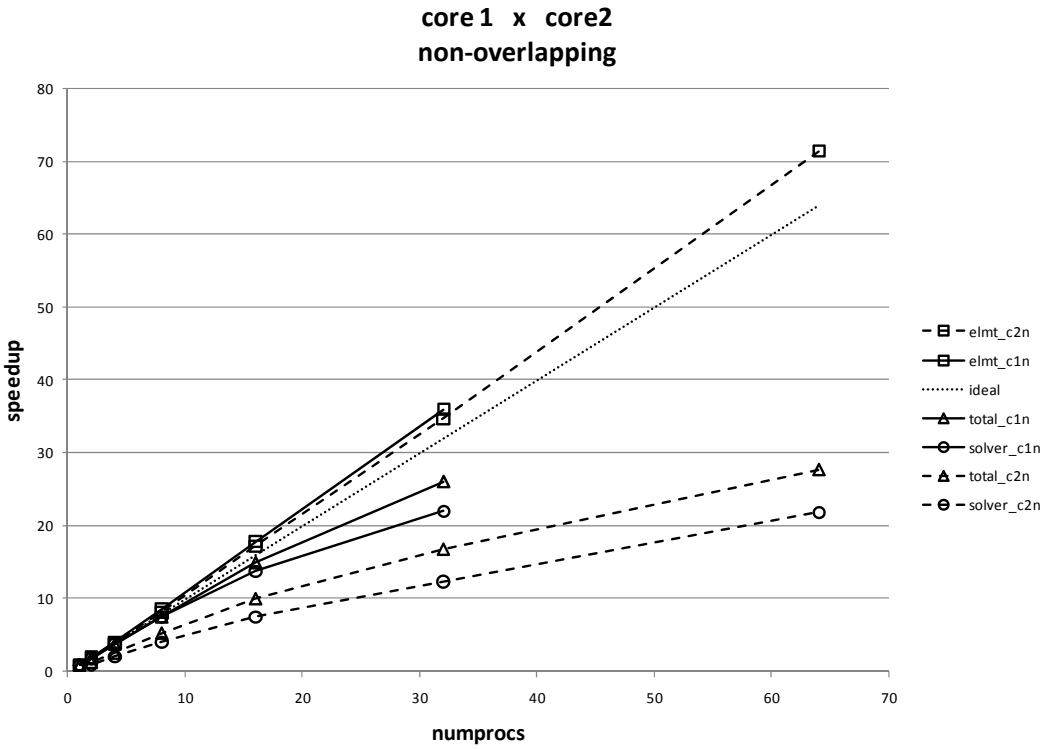


Figura 4.11: Speedups malha 3: core1 x core2, por processo.

A comparação entre os dois métodos mostra a desvantagem do overlapping na fase de montagem de matriz, devido à redundância nos elementos, onde ele é conseqüentemente mais lento. Uma análise geral dos gráficos overlapping x non-overlapping sugere que este fator (redundância nos elementos) pesa na solução como um todo, conferindo superioridade, embora pequena, ao método non-overlapping. Quanto à tecnologia dos dois núcleos (core2), os resultados obtidos apontam a princípio para uma desvantagem do seu uso na solução em paralelo de problemas de EF conforme implementado neste trabalho. O speedup obtido foi sempre pior com 2 núcleos para um mesmo número total de processos. Os processadores utilizados são processadores duais (2 cores), e o padrão MPI permite que cada máquina de fato execute dois processos de memória compartilhada como se fossem de memória distribuída. Este procedimento resultou em speedups iguais ou superiores a 2, na fase de montagem das matrizes

(loop nos elementos) e, em contrapartida, speedups muito pobres na fase de solução do sistema de equações. Deve-se isto ao fato de que durante o loop nos elementos, o número de operações de ponto flutuante prepondera sobre o acesso contínuo à memória, enquanto que na solução do sistema a situação se inverte, com o acesso a memória sendo realizado de forma continuada e em larga escala, criando assim um gargalo no fluxo da execução do código, uma vez que os dois cores passam a disputar o acesso à mesma memória física. Esta questão parece ser o ponto crucial no que diz respeito às modernas arquiteturas de hoje, ou seja, como acessar grandes quantidades de dados em sistemas de memória compartilhada.

Uma análise diferente pode ser feita se compararmos o desempenho máquina por máquina em vez de comparar processo por processo.

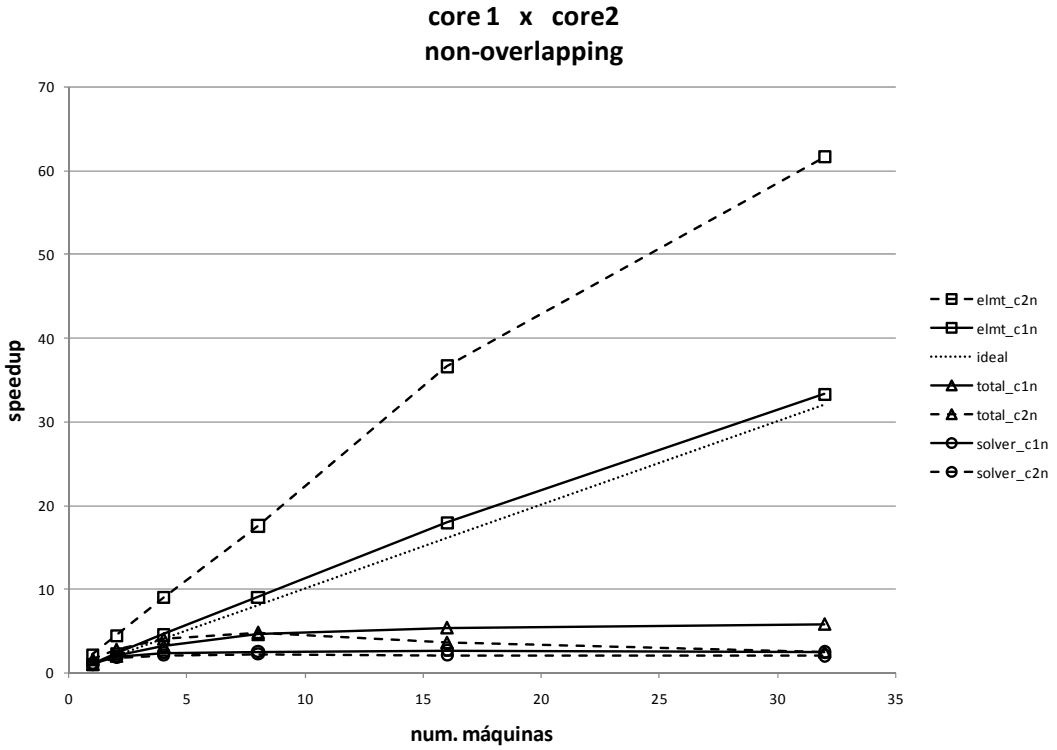


Figura 4.12: Speedups malha 1: core1 x core2, por máquina.

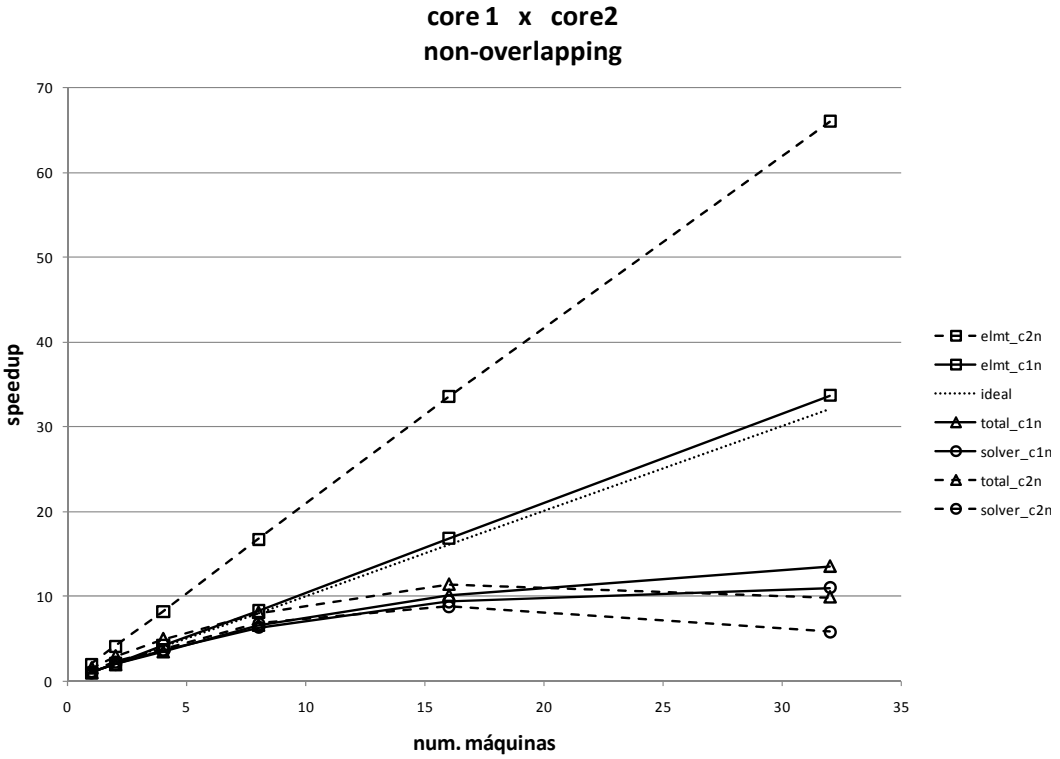


Figura 4.13: Speedups malha 2: core1 x core2, por máquina.

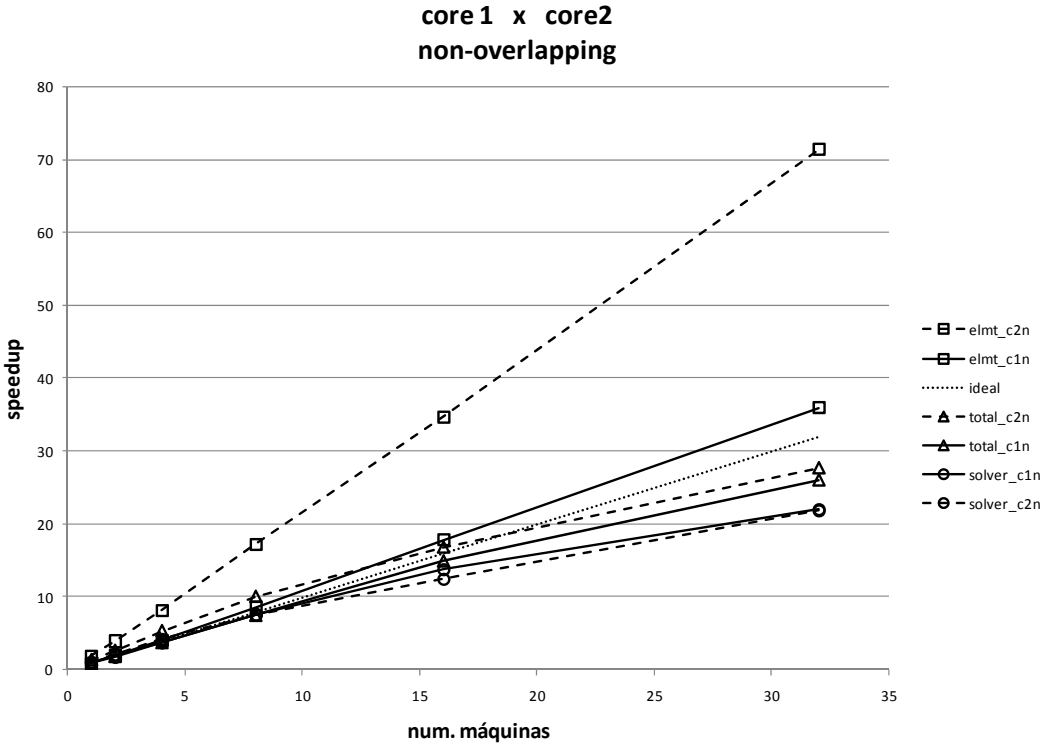


Figura 4.14: Speedups malha 3: core1 x core2, por máquina.

Na comparação por número de máquinas observamos um ganho no speedup consistente na montagem de elementos e uma degradação do speedup no solver.

No conjunto da análise (speedup total) o comportamento varia com o tamanho da malha, sendo a vantagem do uso do core2 crescente com o tamanho da malha.

Como a aplicação proposta do trabalho é a simulação de problemas de grande porte, adotaremos dois processos por máquina (um por núcleo) nas análises seguintes. Para manter a uniformidade nos gráficos passamos a comparar valores obtidos por máquina, em vez de valores por processo. Dessa forma, o valor de referência passa a ser o de uma máquina, ou seja 2 processos. O speedup em n nós (máquinas) passa a ser calculado como:

$$S_n = \frac{t_{n=1}}{t_{n\geq 2}} \tag{4.20}$$

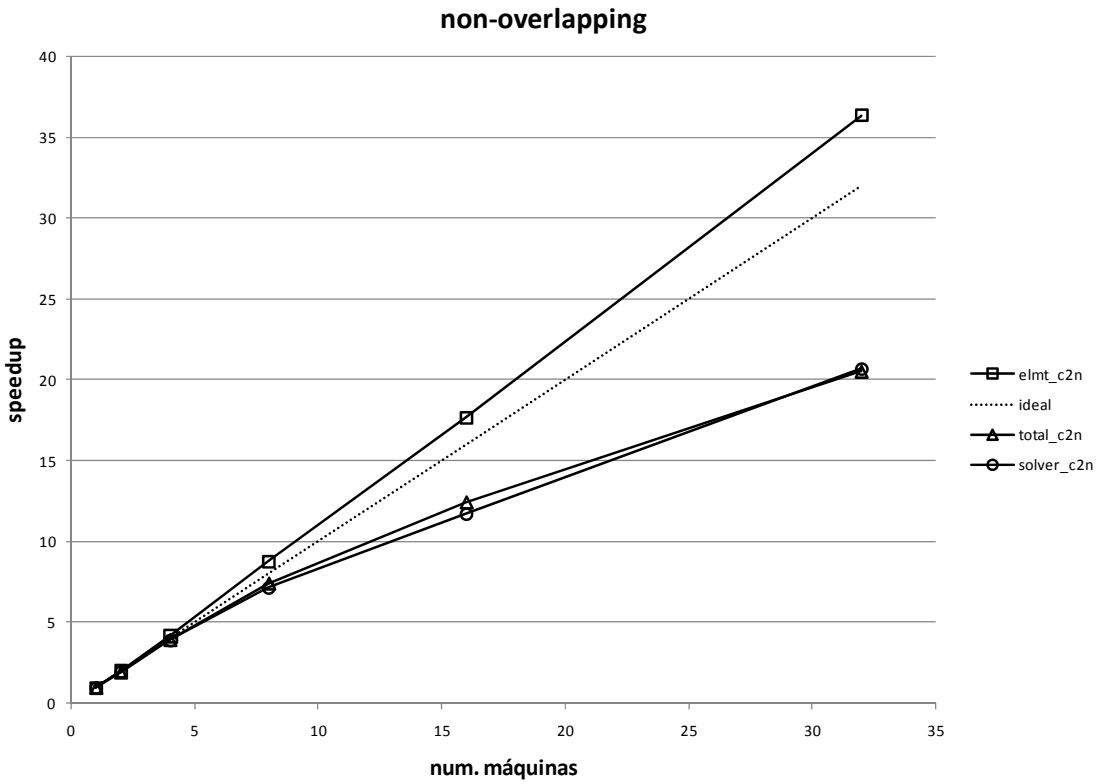


Figura 4.15: Speedups malha 3 non-overlapping por máquina.

Os speedups na Figura 4.15 são menores que os da Figura 4.14 pois o numerador em (4.20) é menor que o de (4.18). Seja a eficiência da paralelização definida por:

$$E_p = \frac{S_p}{p} \qquad ou \qquad E_n = \frac{S_n}{n} \tag{4.21}$$

A Figura 4.16 mostra as eficiências calculadas por processo p e por nó n. No solver, a eficiência E_n é maior que a E_p por conta da redução à metade no denominador.

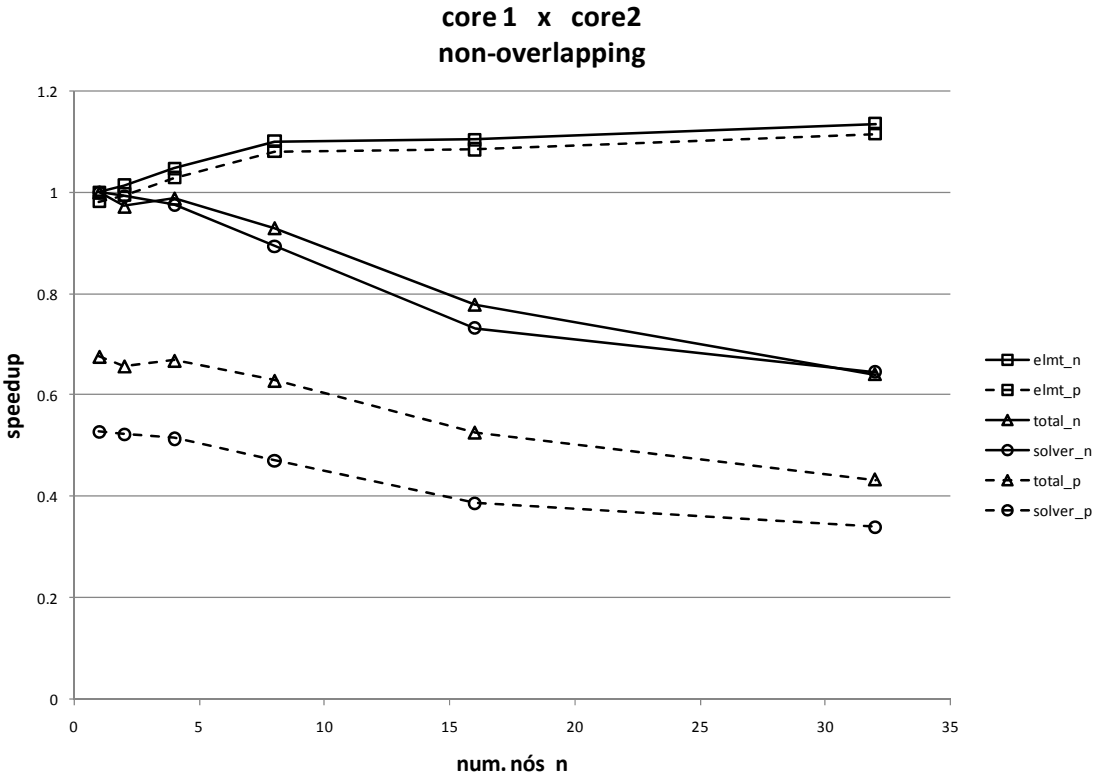


Figura 4.16: Eficiências malha 3 non-overlapping.

4.5 PARALELIZAÇÃO DO MODELO NUMÉRICO ACOPLADO PARA CONCRETO

O modelo numérico descrito no Capítulo 3 foi implementado na base paralelizada de acordo com os conceitos e as características descritas neste capítulo. Nesta implementação apenas o método non-overlapping foi utilizado.

4.5.1 Particionamento com construção em camadas

Considerando o sistema de construção por etapas, existe um problema a ser enfrentado na solução em paralelo: um particionamento feito sobre toda a malha pode criar partições em regiões que só serão ativadas ao final do cronograma de construção. Os processos associados a tais partições ficarão ociosos até a ativação destas. Tal ociosidade constitui um paralelização pobre. O melhor aproveitamento do processamento paralelo depende de uma distribuição equilibrada de carga de trabalho entre os processadores. Para evitar este problema é necessário fazer o re-particionamento da malha a cada nova camada ativada. Assim, quando uma nova camada é ativada, tem-se um novo domínio espacial a definir o problema, e deve haver portanto um novo particionamento correspondente, que distribua o melhor possível a mesma quantidade de trabalho (elementos, nós, equações) a cada processo. Foi utilizado o pacote Métis para particionar as malhas. A biblioteca do Métis, particularmente a subrotina `metis_partmeshdual` foi inserida no código.

1. se há lançamento de camada:
2. descarta arranjos locais do sistema
3. monta arranjos globais
4. define quais elementos estão ativos
5. define novas condições de contorno
6. loop nos elementos
7. se elemento está ativo
8. copia conectividades $\{ix\}_{Total}$ para $\{ix0\}_{Ativo}$
9. fim loop nos elementos
10. descarta arranjos globais
11. chama metis_partmeshdual($\{ix0\}_{Ativo}$)
12. cria estruturas de dados locais
13. numera equações localmente
14. reordena $\mathbf{V}^{i,1}$.
15. aloca arranjos para o sistema
16. prepara a comunicação:
17. identifica partições vizinhas
18. cria mapas de comunicação
19. retorna à solução

Algoritmo 4.7: Esquema de re-particionamento após alteração de geometria.

No Algoritmo 4.7, as linhas 4-9 são executadas sobre arranjos relativos à malha final do problema (nós, elementos, condições de contorno, etc.). Assim sendo, cada processo deve ter, neste ponto da execução, uma cópia global dos dados do problema. Para disponibilizar estes arranjos (por exemplo: temperaturas, deslocamentos, tensões, grau de hidratação) são realizadas operações (linha 3) de comunicação coletiva tipo mpi_allreduce, em que cada processo contribui com sua parte do arranjo que é então montado e redistribuído entre os processos. Variáveis relativas a elementos e nós ainda inativos recebem valor zero. Após o devido uso dos arranjos globais, os eles são descartados (linha 10) para liberar memória para os novos arranjos do sistema (linhas 12 e 15), cujas versões anteriores foram descartadas da memória na linha 2 para minimizar o consumo de memória. Alguns dados específicos como coordenadas nodais, conectividades de elementos e condições de contorno (prescrições de deslocamento) devem ter uma cópia global mantida e isto é feito no processo raiz (rank 0 no mpi). Isto ocorre porque se estas informações forem perdidas na operação da linha 10 não será possível recuperá-las posteriormente. Portanto o processo raiz utiliza mais memória que os demais. O re-ordenamento apontado na linha 14 é feito utilizando-se o algoritmo "reverse Cuthill-Mckee" (RCM), e atua somente sobre os nós $\mathbf{V}^{i,1}$. Esta restrição se dá para que não seja perdida a separação feita entre os tipos de nós $\mathbf{V}^{i,1}$, $\mathbf{V}^{i,2}$ e $\mathbf{V}^{i,3}$, que é necessária à eficiência dos algoritmos de preparação e execução da comunicação entre processos.

5 SIMULAÇÃO DE UM VERTEDOURO

Parte importante de uma barragem, o vertedouro é a estrutura que dá vazão ao excedente de água acumulada no reservatório. Por trabalhar sob fluxo de água turbulenta deve ter resistência à abrasão, por isso normalmente é construído em concreto com resistências maiores que as usadas em outras regiões da usina. Em comparação com o material usado no corpo da barragem de gravidade, a maior resistência requerida nesta estrutura normalmente é conseguida à custa de maior consumo de cimento. E maior consumo de cimento implica, via de regra, maior calor de hidratação, o que assevera os problemas de origem térmica.

5.1 ESTRUTURA

As figuras 5.1 e 5.2 mostram a estrutura criada para simular a construção de um vertedouro. A estrutura conta com três pilares com calhas e bloco munhão para suporte das comportas. Por um dos pilares desce uma galeria que atravessa a

fundação. A curva do vertedouro é seguida por uma rampa e um salto de esqui (estrutura com fim de dissipação da energia cinética da água).

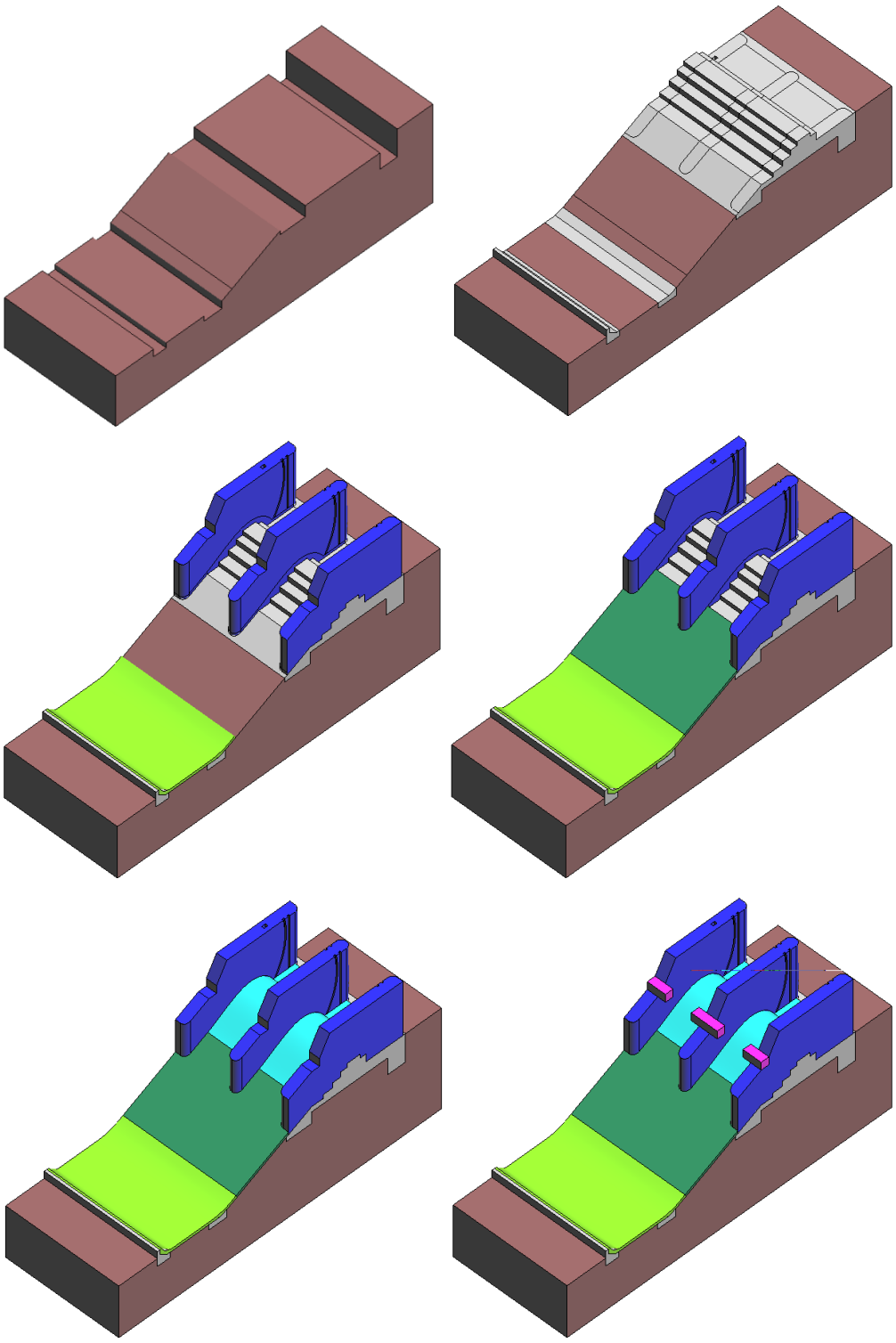


Figura 5.1: Partes da estrutura do vertedouro: solo, fundação, pilares e salto de esqui, rampa, curva do vertedouro, blocos munhão.

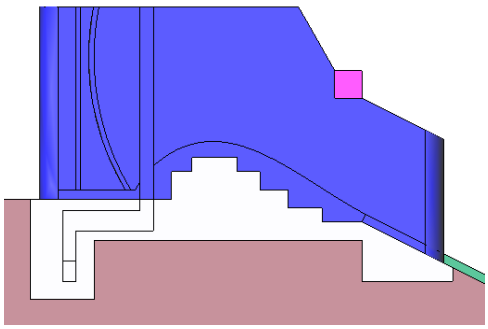


Figura 5.2: Detalhe da estrutura do vertedouro: galeria.

Figura 5.3: Malha da estrutura do vertedouro.

A malha usada tem 239.402 nós e 1.288.422 tetrahedros. Foram impostas restrições de deslocamento e condição adiabática nas laterais e fundo. Devido à baixa rigidez do concreto ao ser lançado, foram criadas restrições de deslocamento (apoio) variáveis para simular o efeito das formas. Isto evita o aparecimento de grandes deformações que podem até impossibilitar a análise por demora excessiva na convergência no NRM. Estas restrições são impostas sobre os nós da superfície externa da camada recém lançada. Após um determinado tempo na simulação (incrementos de tempo) estas restrições são retiradas, e os graus de liberdade correspondentes, antes inativos, passam a integrar o sistema de equações. A partir deste ponto o cálculo segue e estes nós começam a experimentar deslocamentos. Nesta análise foi considerado como tempo de desforma o período de dois dias.

5.2 MATERIAIS

Foram utilizados quatro materiais, com propriedades baseadas em (Fairbairn, et al. 2008). As propriedades adotadas estão listadas nas tabelas a seguir, juntamente com as do solo. Os quatro materiais distintos são destinados às quatro partes principais da estrutura em questão: fundação, pilares, blocos munhão e superfícies de escoamento. A Tabela 5.1 apresenta as propriedades dos materiais. A Figura 5.4 apresenta as curvas de elevação adiabática de temperatura dos diferentes concretos usados na simulação. As Figuras 5.5-6 mostram as curvas de propriedade em função do grau de hidratação obtidas pelo modelo da equação (2.26)(Krauss e Hariri 2006), porém com valores dos expoentes adaptados para melhor ajuste aos materiais em questão.

Tabela 5.1: Propriedades de materiais.

	fundação	pilares	munhão	superfícies	solo
E_∞ [GPa]	27,4662	32,837	27,4662	27,4662	40
$f_{c\infty}$ [MPa]	34,83	39,47	45,47	39,47	
ε_{1u} [1]	0,0022	0,0022	0,0022	0,0022	
f_t [MPa]	3,3252	3,772	4,672	3,772	
H [Pa]	0,1	0,12	0,12	0,12	
J_∞^v [Pa^{-1}]	13*10^{-12}	12.28*10^{-12}	12.28*10^{-12}	12.28*10^{-12}	
Γ_0 [Pa^{-1}dd^{-1}]	4	7	7	7	
$\tau_{w,\infty}$ [dd]	0,5	0,8	0,8	0,8	
c_e [J/KgK]	980	980	980	980	1074
k [W/mK]	3,1	3,1	3,1	3,1	4
ρ [Kg/m^3]	2280	2280	2280	2280	2760

Foram adotados $\alpha = 11*10^{-6}K^{-1}$ e $\upsilon = 0,2$ para todos os materiais. Para os concretos f_y=25MPa e f_{y2}=60MPa.

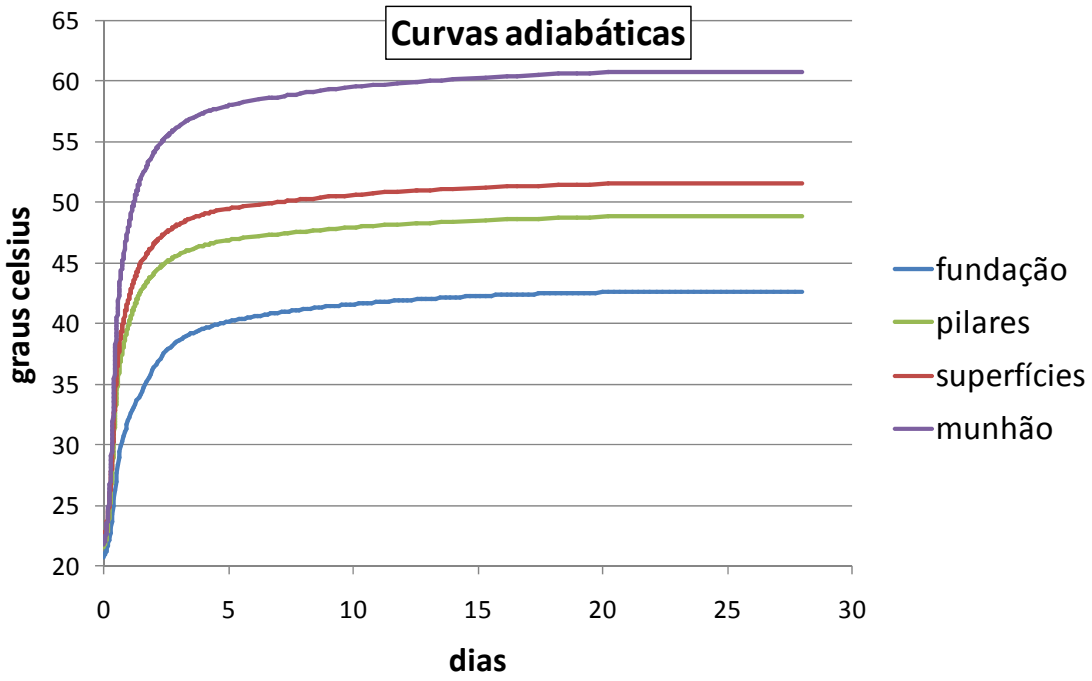


Figura 5.4: Curvas de elevação adiabática de temperatura.

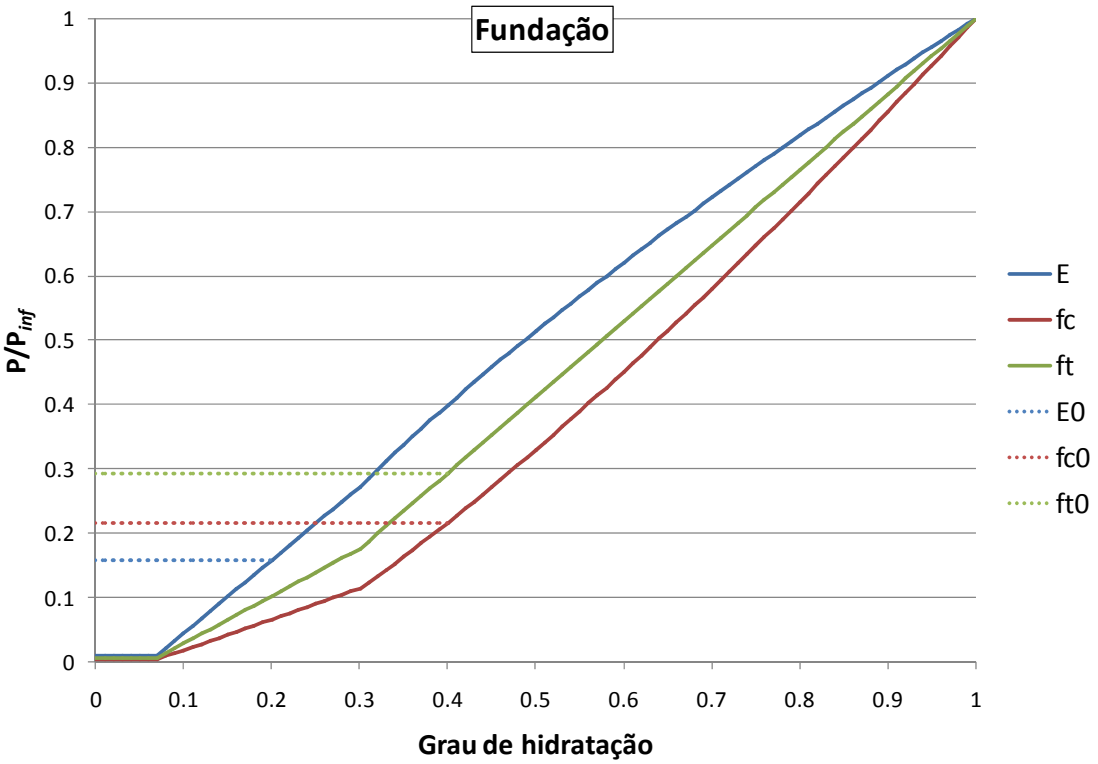


Figura 5.5: Propriedades x grau de hidratação, fundação.

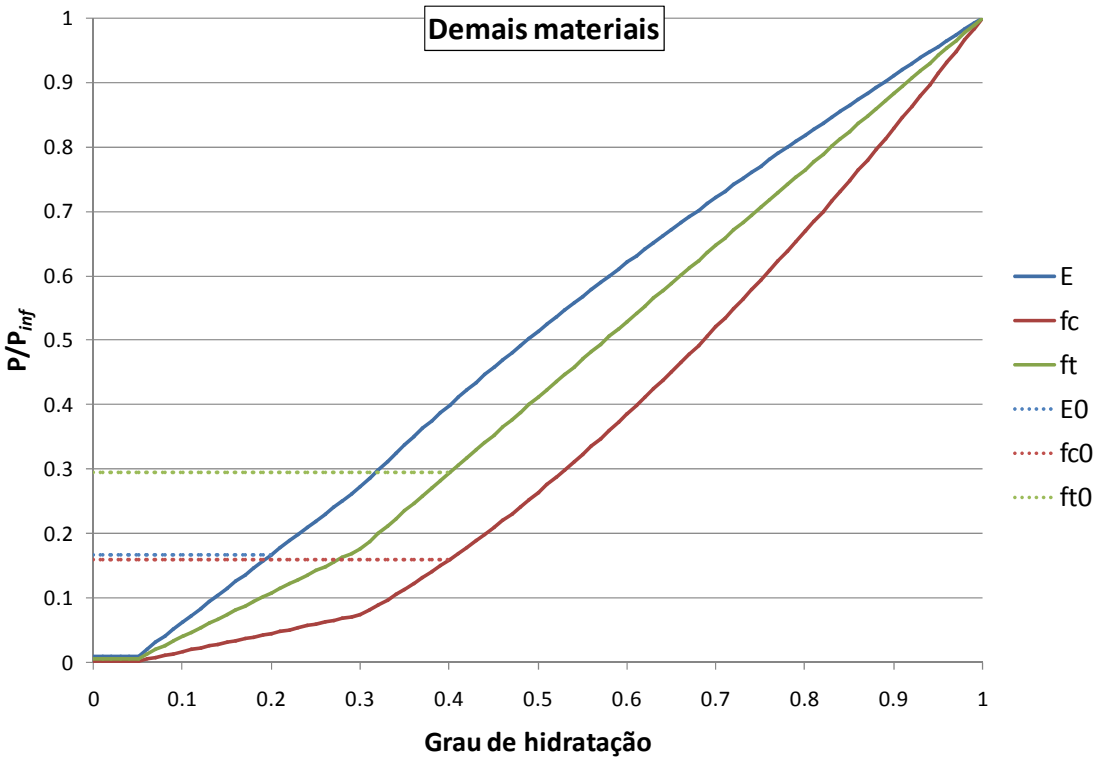


Figura 5.6: Propriedades x grau de hidratação, pilares, superfícies e munhão.

5.3 PLANO DE CONSTRUÇÃO

Inicialmente são concretadas as fundações. Em seguida são executados o salto de esqui e os pilares. Durante a construção dos pilares (parte mais demorada) são iniciadas a rampa e a curva do vertedouro. Por último são concretados os blocos munhão. A Figura 5.1 mostra esquematicamente as etapas de concretagem.

5.4 RESULTADOS

As figura 5.7-9 a seguir mostram, em cortes pela estrutura, os campos de temperatura obtidos em alguns instantes de tempo destacados na simulação. Os valores finais das variáveis internas de plasticidade são mostradas Figuras 5.10-11.

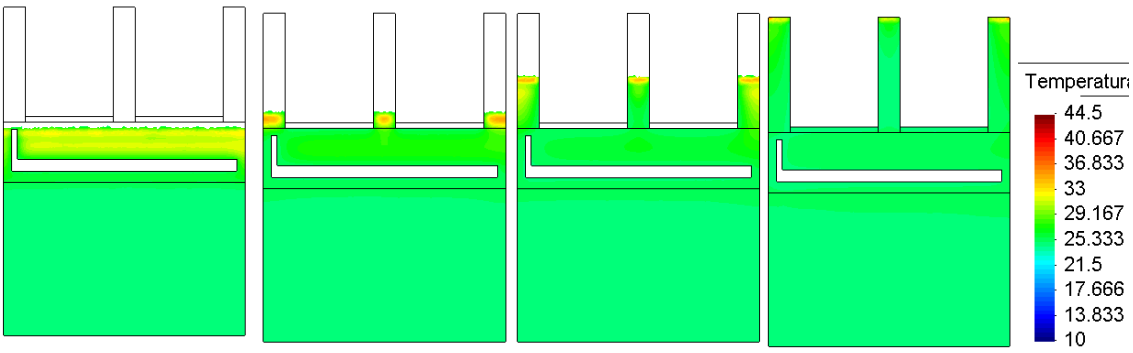


Figura 5.7: Temperatura, corte transversal passando pela galeria.

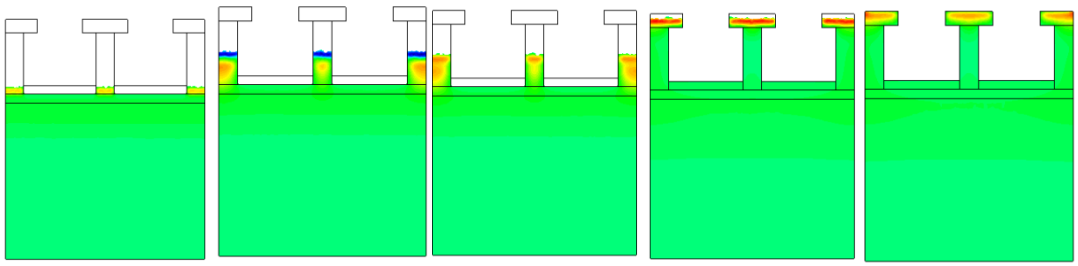


Figura 5.8: Temperatura, corte transversal passando pelo munhão.

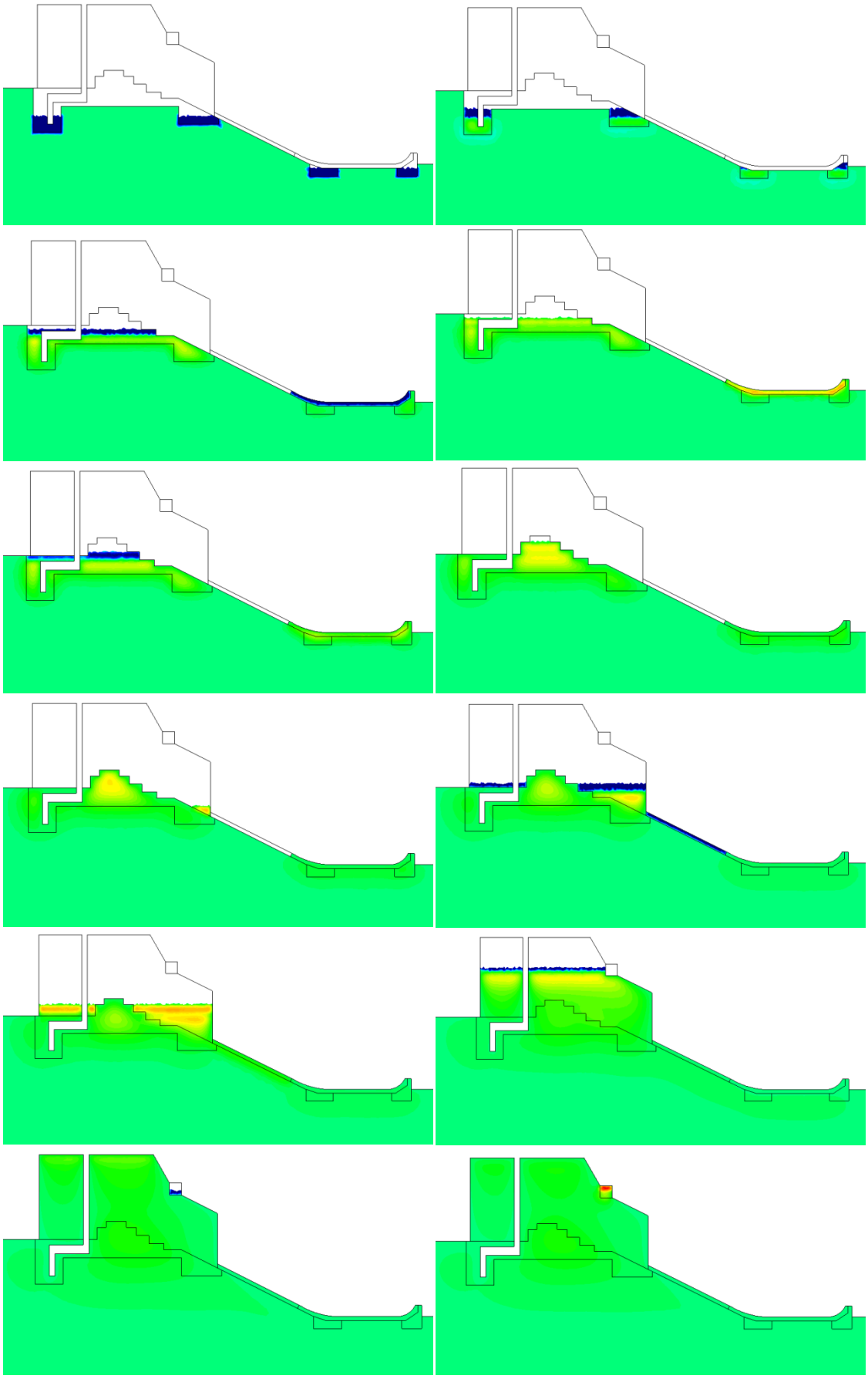


Figura 5.9: Temperatura, corte longitudinal passando pelo pilar da direita.

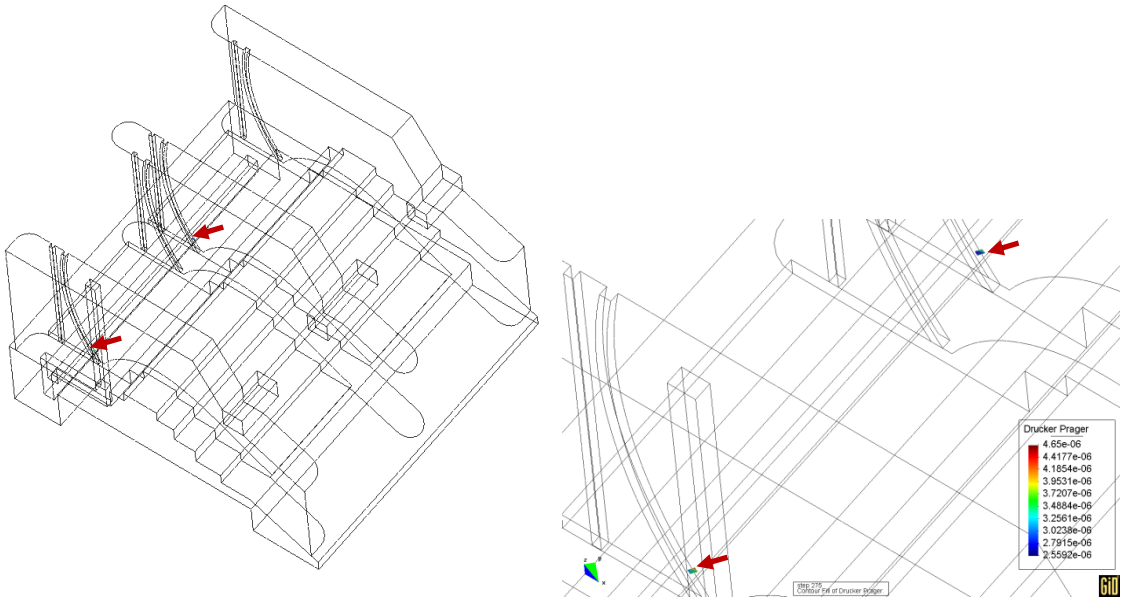


Figura 5.10: Regiões plastificadas pelo critério Drucker-Prager.

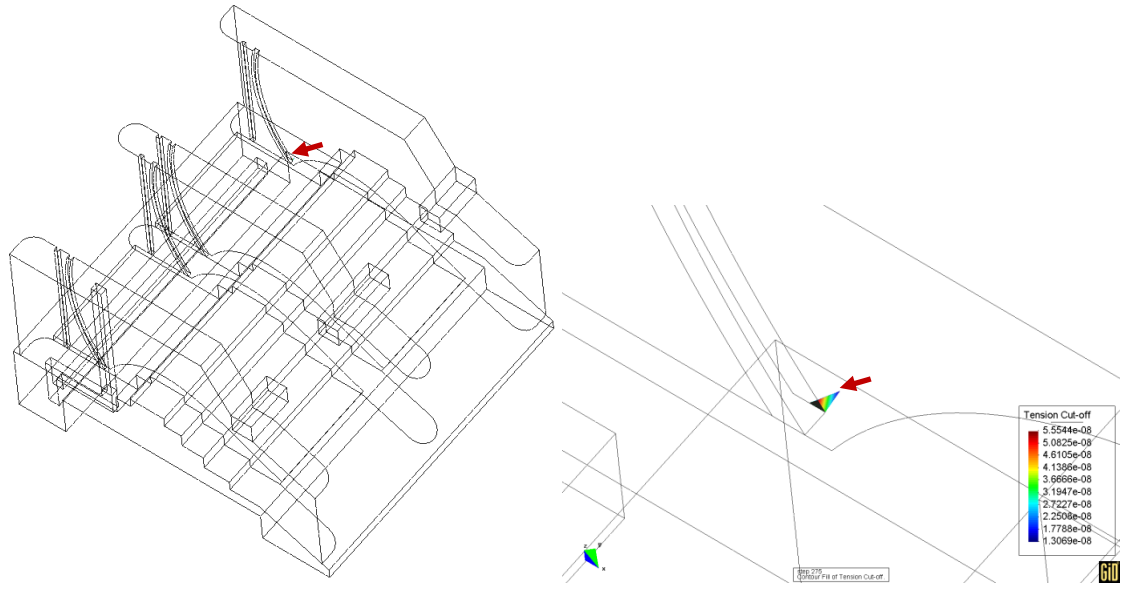


Figura 5.11: Região plastificada pelo critério tension cut-off.

5.5 DESEMPENHO

As figuras a seguir mostram curvas de tempo de execução (Wall clock time), Speedup (de acordo com (4.20)) e eficiência (de acordo com (4.21)b) variando ao longo da evolução (crescimento) do domínio. O problema foi resolvido em um cluster de PCs cujas características estão na Tabela 5.2. A simulação de 275 dias em

3680 passos de tempo de tamanho variável correspondeu a 5840 soluções PCGD do problema térmico e a 6964 soluções PCGD do problema mecânico, devido às não-linearidades nos dois problemas. As Figuras 5.12-13 são exemplos de resultados do particionamento dinâmico. Elas mostram diferentes particionamentos para etapas diferentes da simulação. As Figuras 5.14-15 mostram respectivamente a quantidade de coeficientes da matriz $\{al\}$ armazenados e o número de equações crescendo com o avanço das camadas de construção. As Figuras 5.16-29 mostram gráficos de tempo de execução individualizado por camada (Figuras pares) e também acumulado (Figuras ímpares) ao longo da execução. As Figuras 5.30-34 mostram os gráficos de speedup individualizado por camada (eixo x) e por parte do código (diferentes curvas). A Figura 5.35 mostra os speedups gerais da simulação, considerados os tempos acumulados da análise completa. As Figuras 5.36-40 mostram a evolução da eficiência da paralelização com o avanço das camadas. A Figura 5.41 mostra a eficiência em cada parte do código da paralelização na simulação como um todo. As Figuras 5.42-46 mostram a evolução da taxa de comunicação no módulo mecânico com o avanço das camadas.

Tabela 5.2: Características de hardware e software.

processador	clock	Cache	sist. oper.	compilador
Intel® Core™2 Duo E8200	2,66 GHz	6 MB L2	CentOs Linux (Kernel 2.6.18)	Intel Fortran 10.1
Placa mãe	FSB	RAM	Rede	Switch
Intel® DG33BU	1333MHz	8GB DDR2 667MHz	Intel® PRO 1Gbit	3com baseline 2498 plus

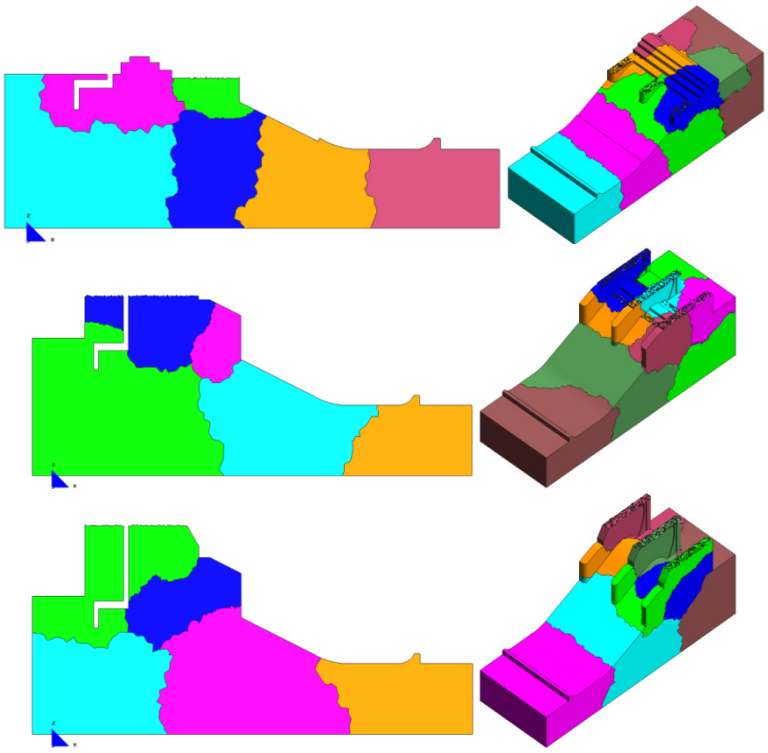


Figura 5.12: Particionamentos com 8 processos (camadas 10, 20 e 30).

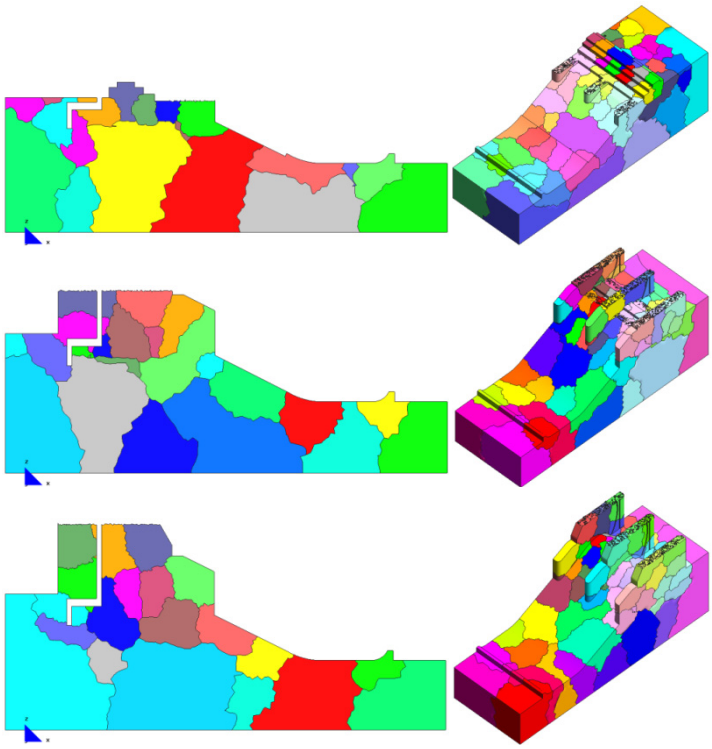


Figura 5.13: Particionamentos com 64 processos (camadas 10, 20 e 30).

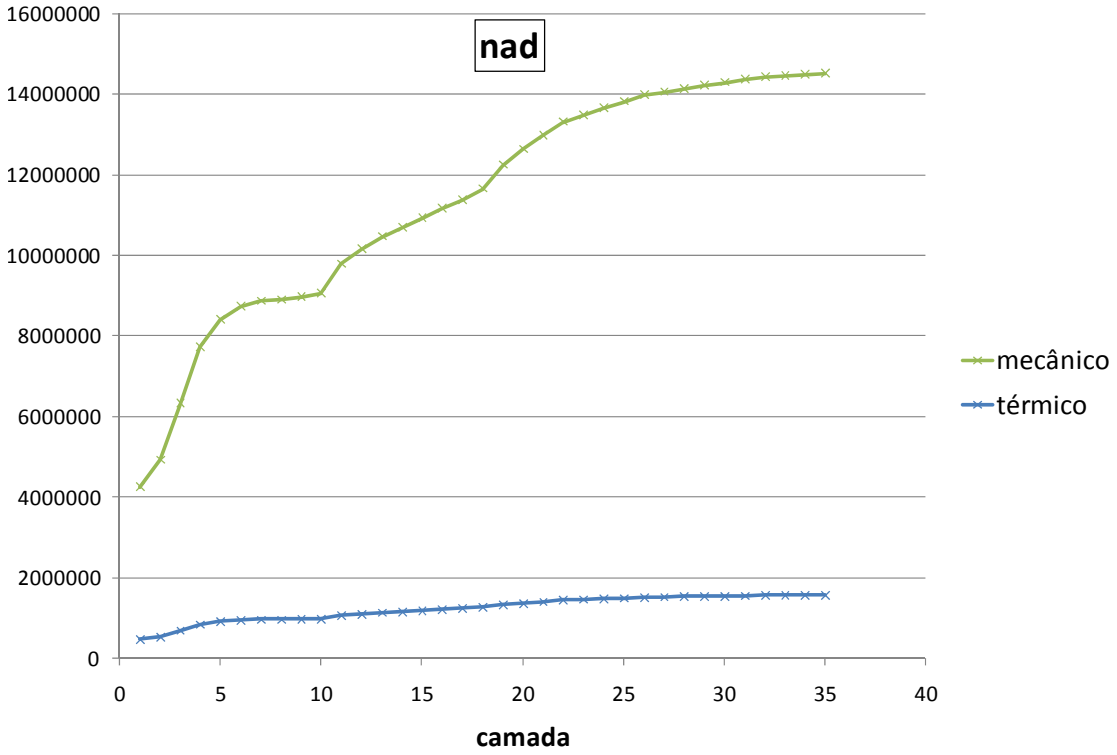


Figura 5.14: Número de coeficientes da matriz armazenados (nad) por camada.

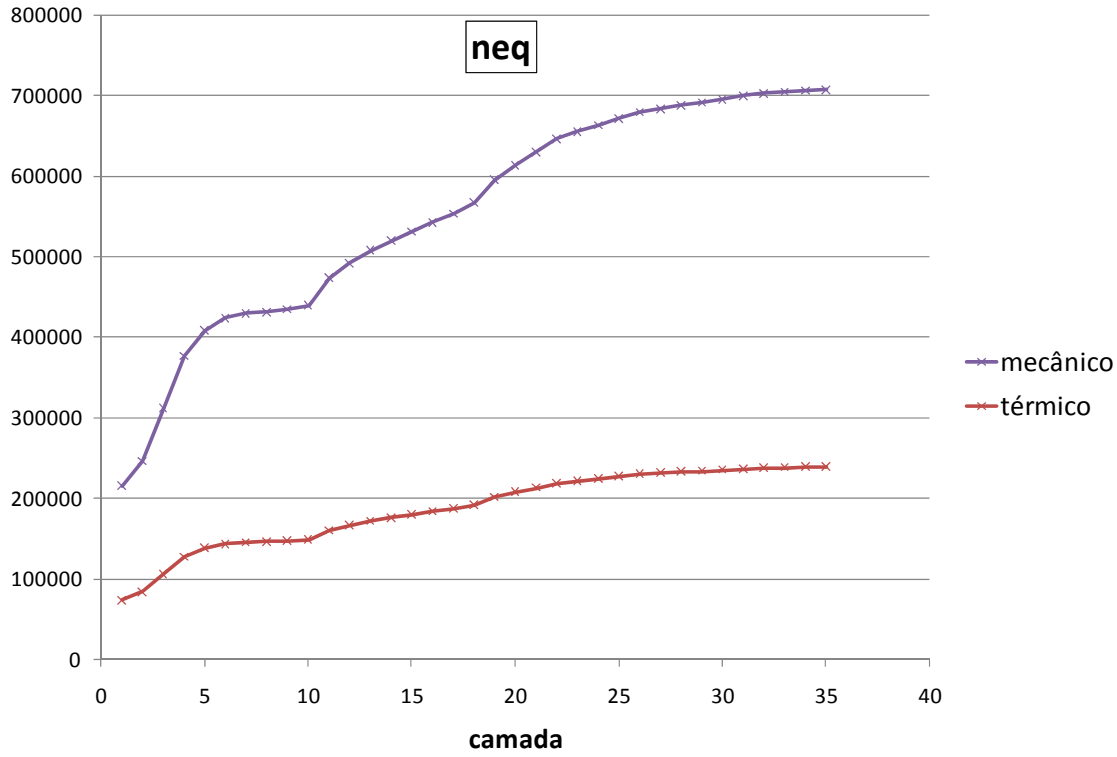


Figura 5.15: Número de equações (neq) por camada.

Nas Figuras a seguir a seguinte notação foi usada nas legendas das curvas: **T-elm** indica montagem de elementos no módulo térmico; **T-solv** indica solução do sistema linear pelo PCGD no módulo térmico; **M-elm** indica montagem de elementos no módulo mecânico; **M-solv** indica solução do sistema linear pelo PCGD no módulo mecânico; **Total** indica o conjunto de operações da análise, incluindo elementos, solver e demais operações de controle da execução. Ainda sobre as legendas, o número à direita do nome, quando houver, indica a quantidade de nós usada na solução em paralelo.

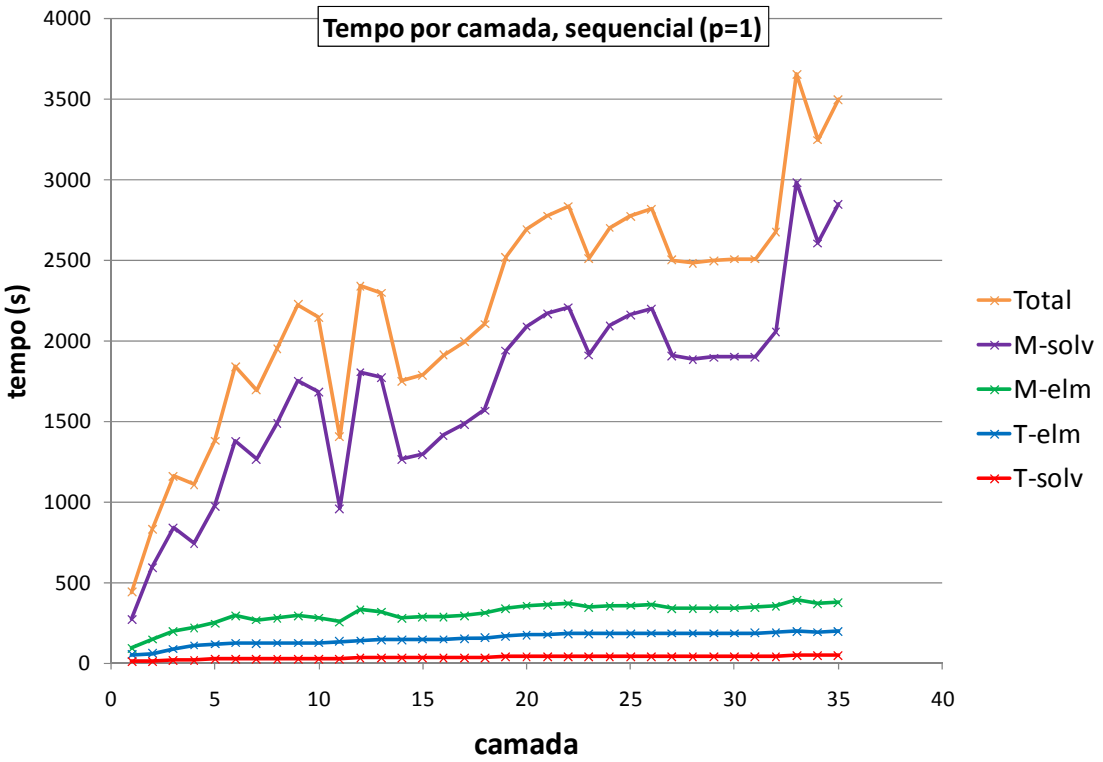


Figura 5.16: Tempo seqüencial (n=1, p=1) por camada [s].

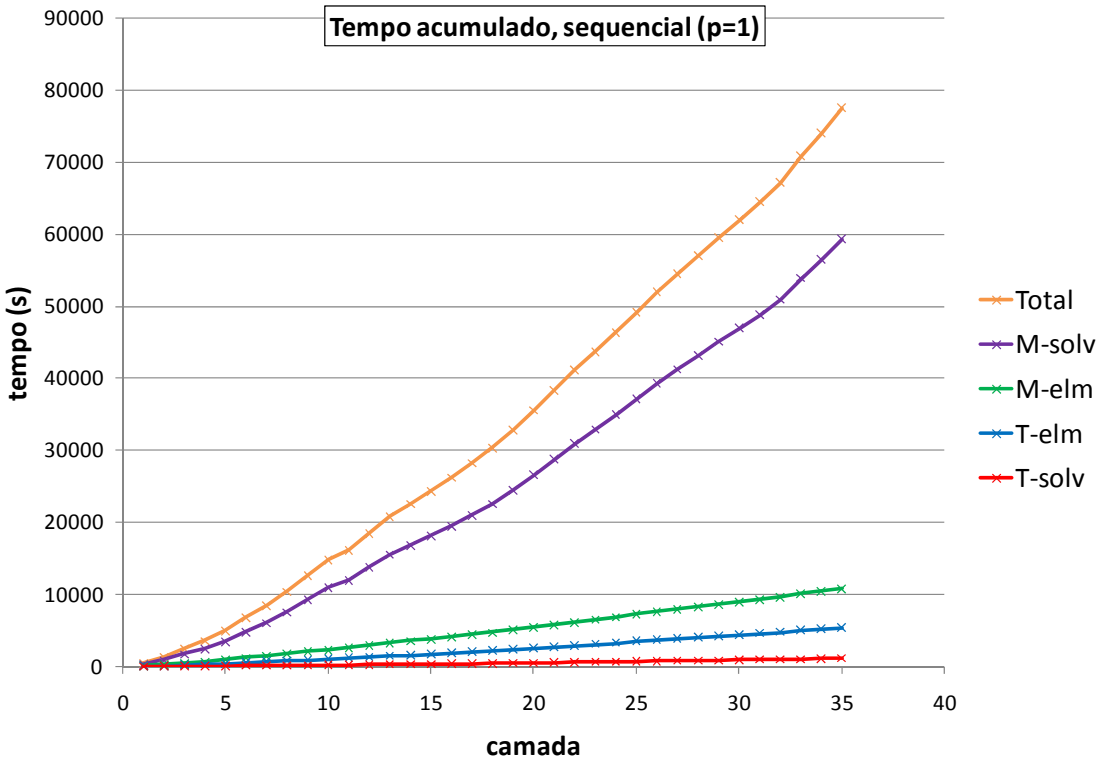


Figura 5.17: Tempo seqüencial (n=1, p=1) acumulado [s].

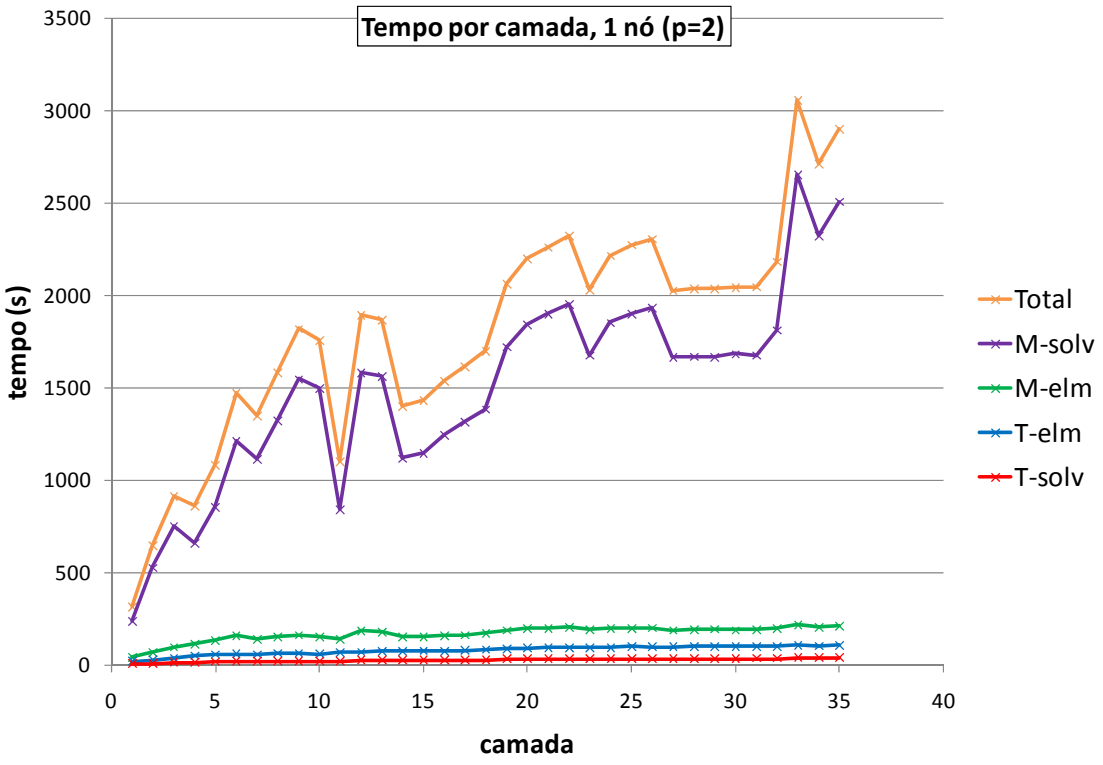


Figura 5.18: Tempo paralelo (n=1, p=2) por camada [s].

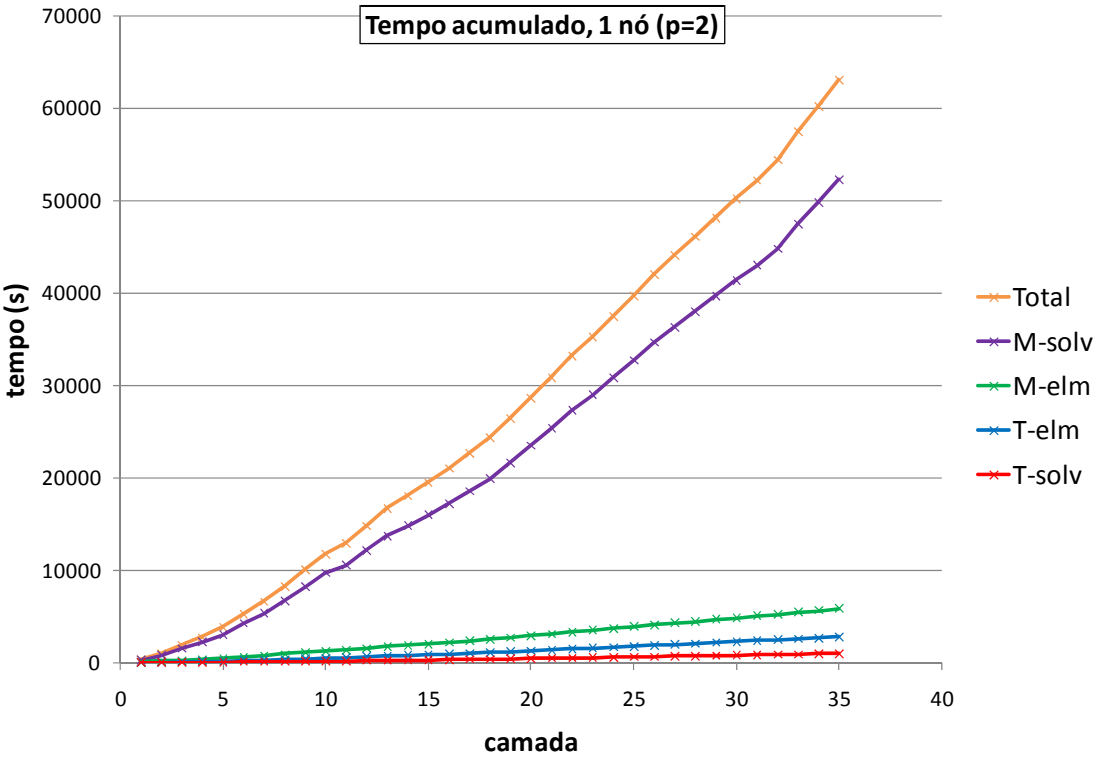


Figura 5.19: Tempo paralelo (n=1, p=2) acumulado [s].

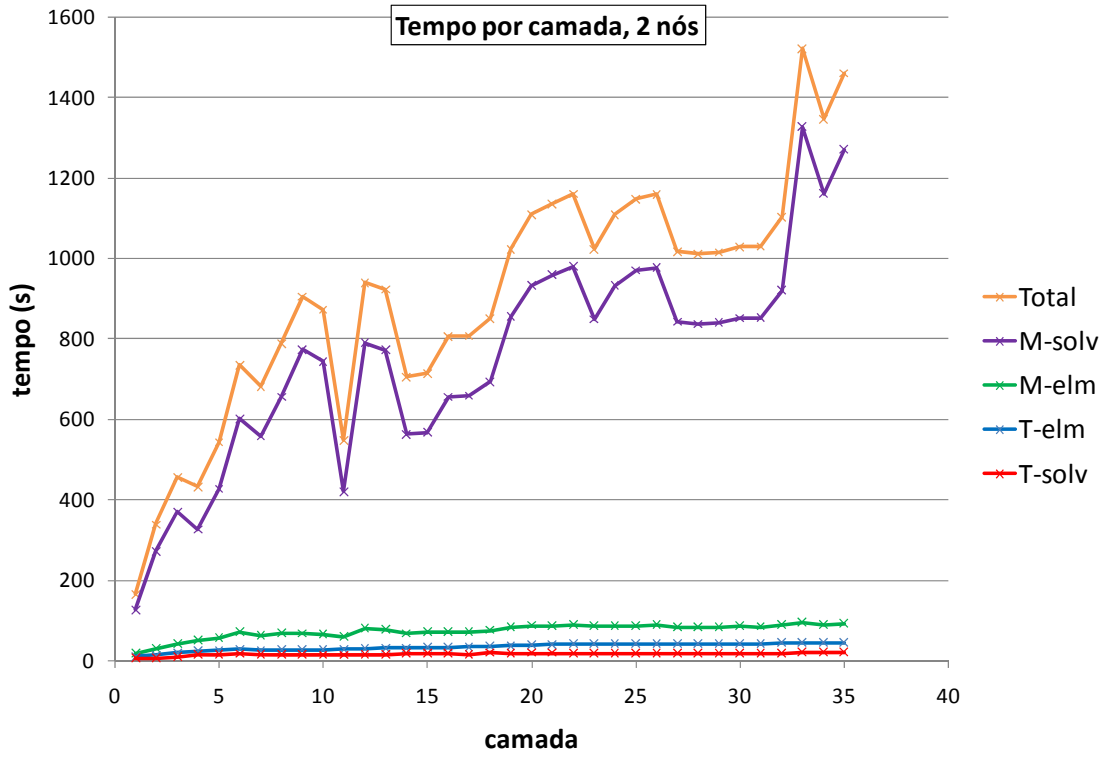


Figura 5.20: Tempo paralelo (n=2, p=4) por camada [s].

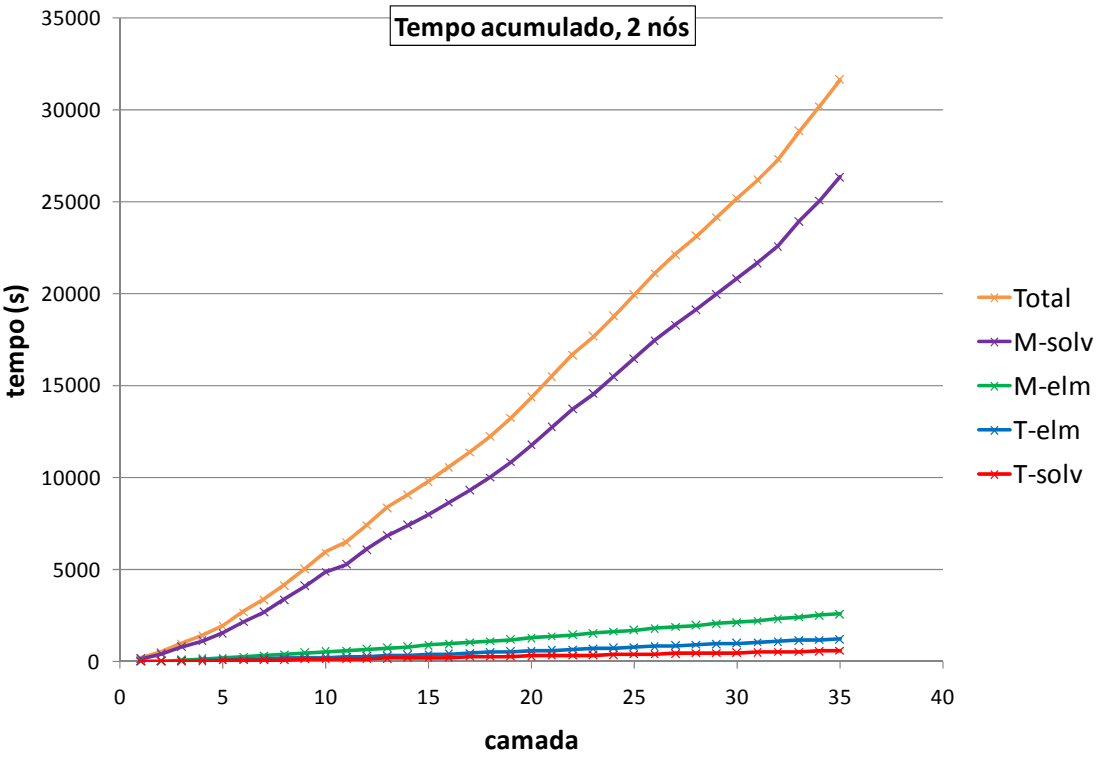


Figura 5.21: Tempo paralelo (n=2, p=4) acumulado [s].

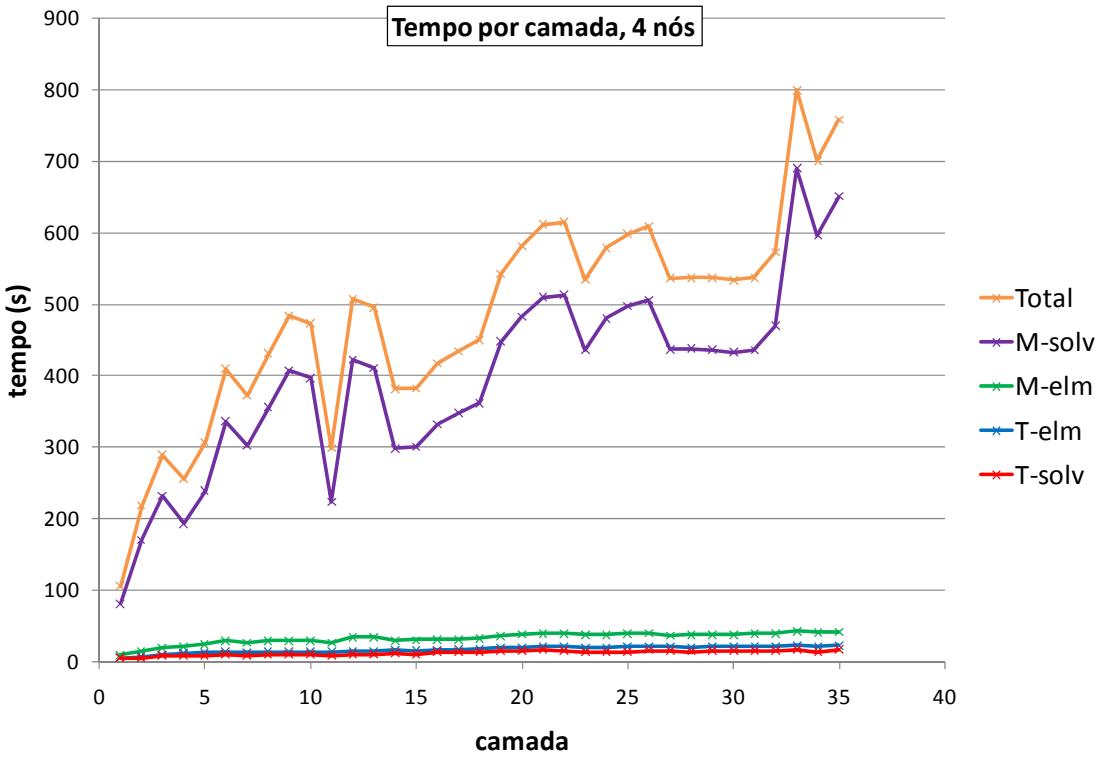


Figura 5.22: Tempo paralelo (n=4, p=8) por camada [s].

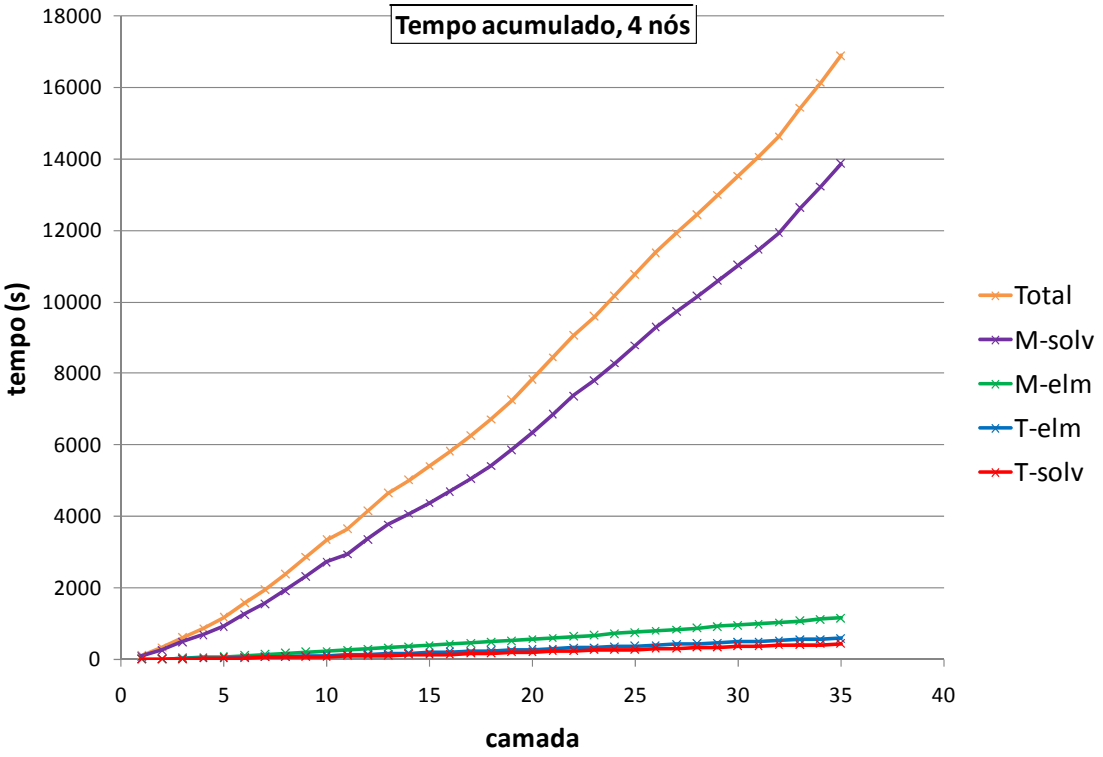


Figura 5.23: Tempo paralelo (n=4, p=8) acumulado [s].

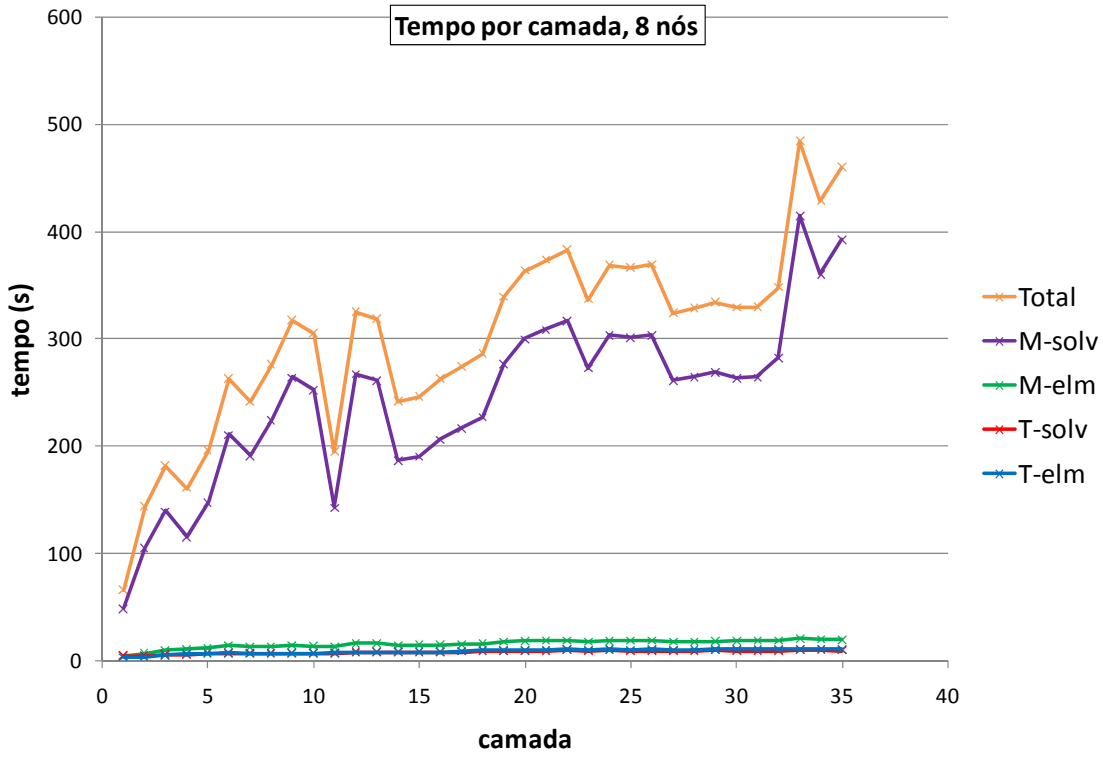


Figura 5.24: Tempo paralelo (n=8, p=16) por camada [s].

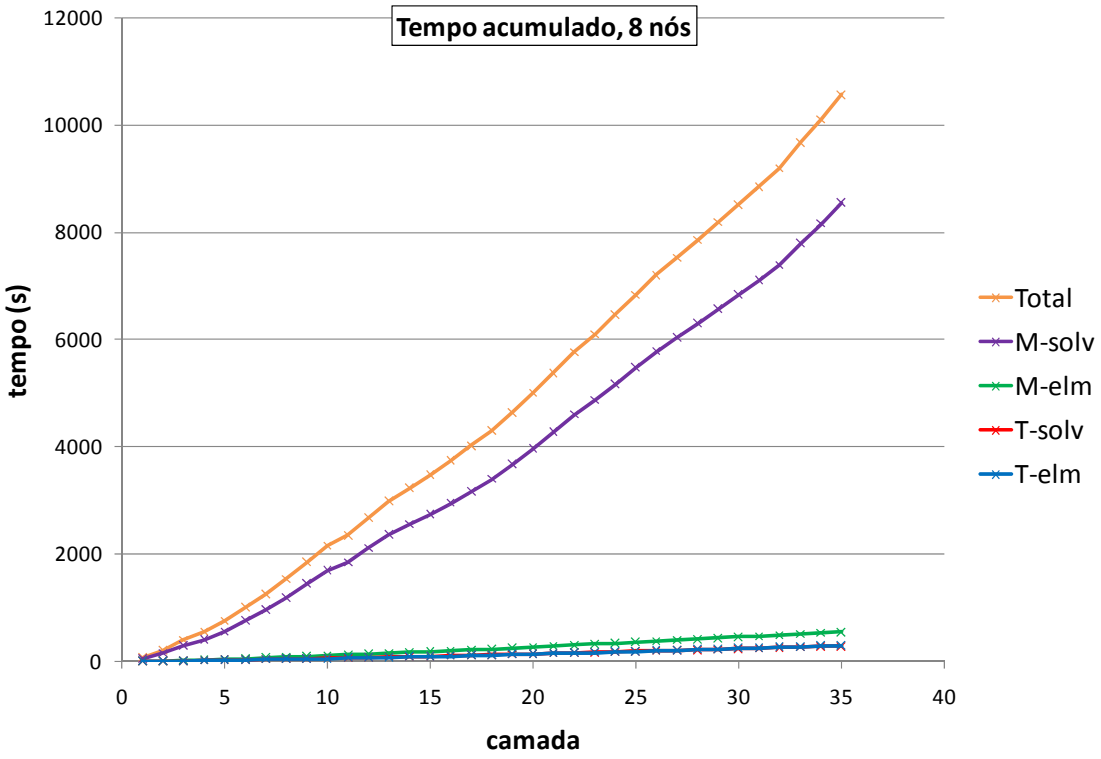


Figura 5.25: Tempo paralelo (n=8, p=16) acumulado [s].

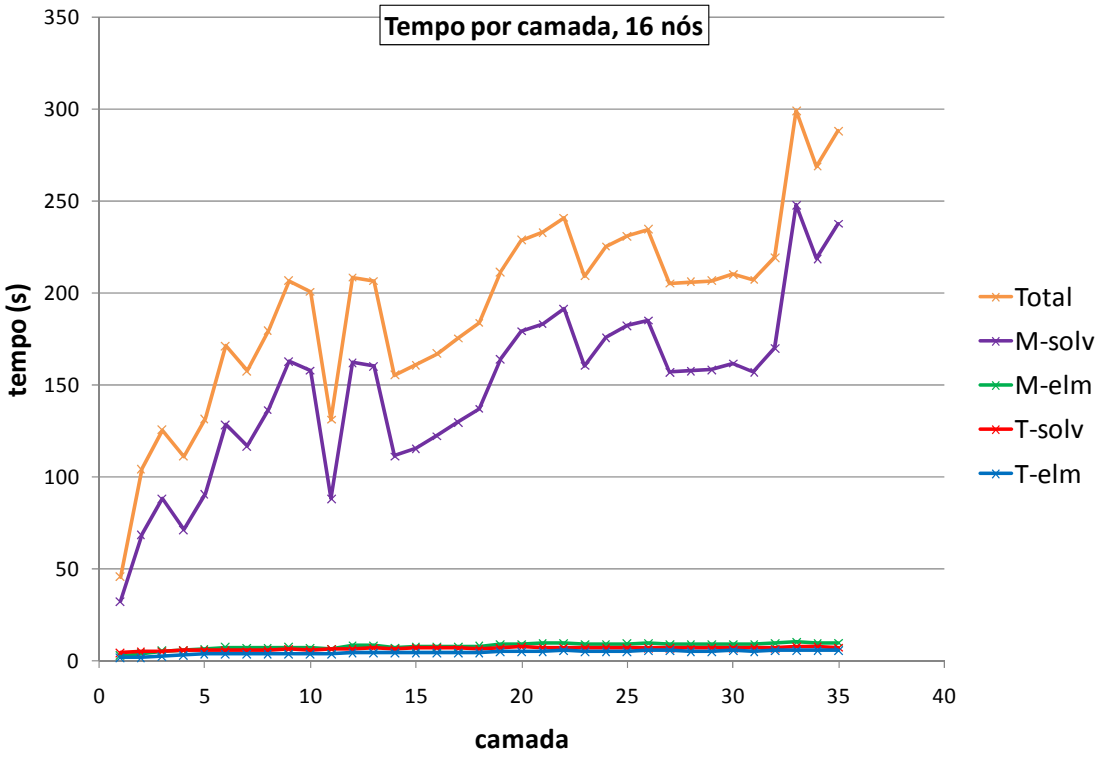


Figura 5.26: Tempo paralelo (n=16, p=32) por camada [s].

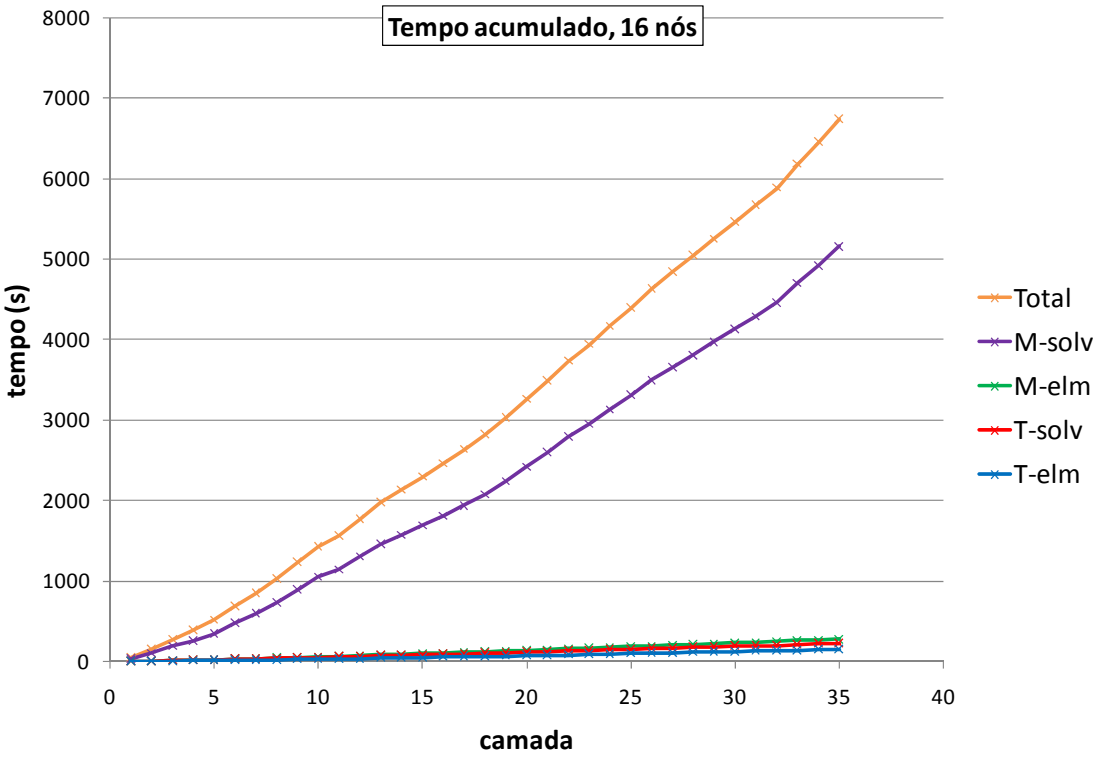


Figura 5.27: Tempo paralelo (n=16, p=32) acumulado [s].

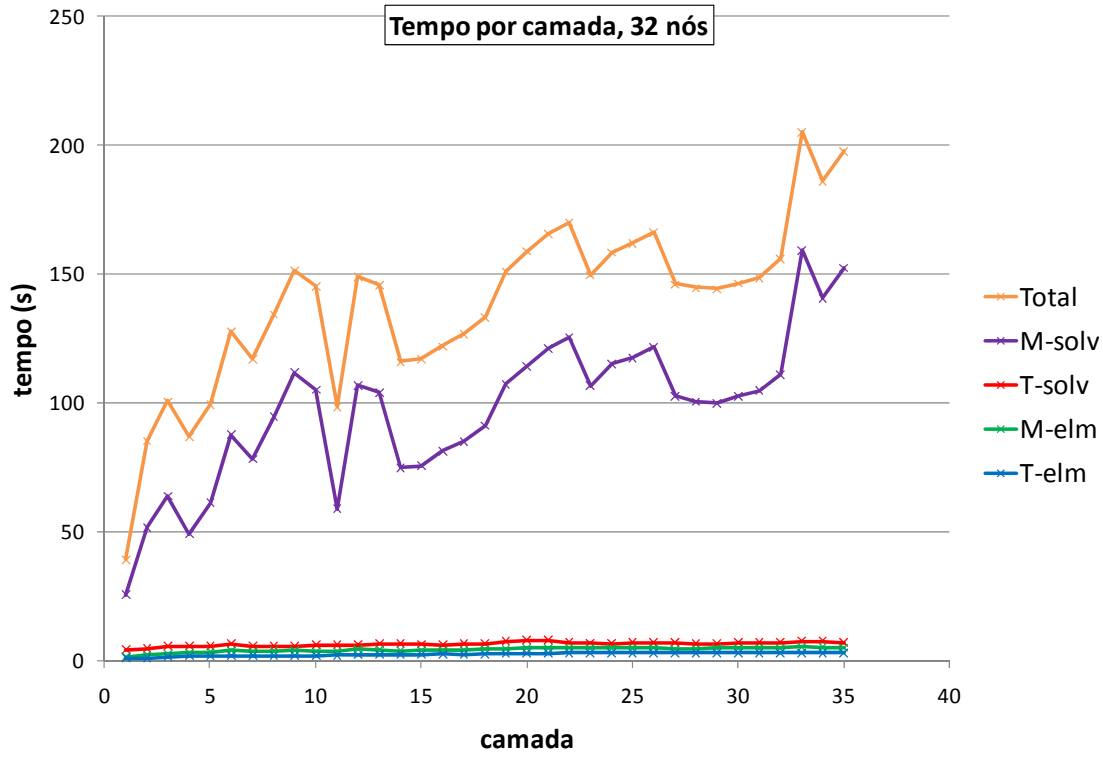


Figura 5.28: Tempo paralelo (n=32, p=64) por camada [s].

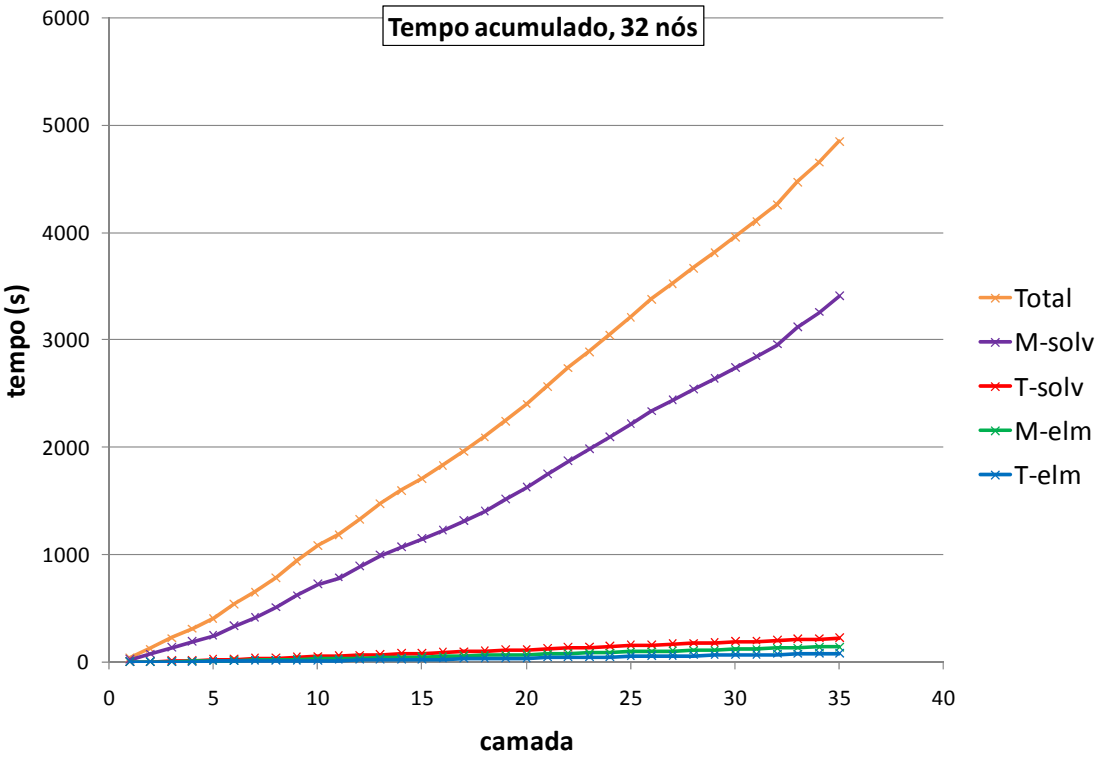


Figura 5.29: Tempo paralelo (n=32, p=64) acumulado [s].

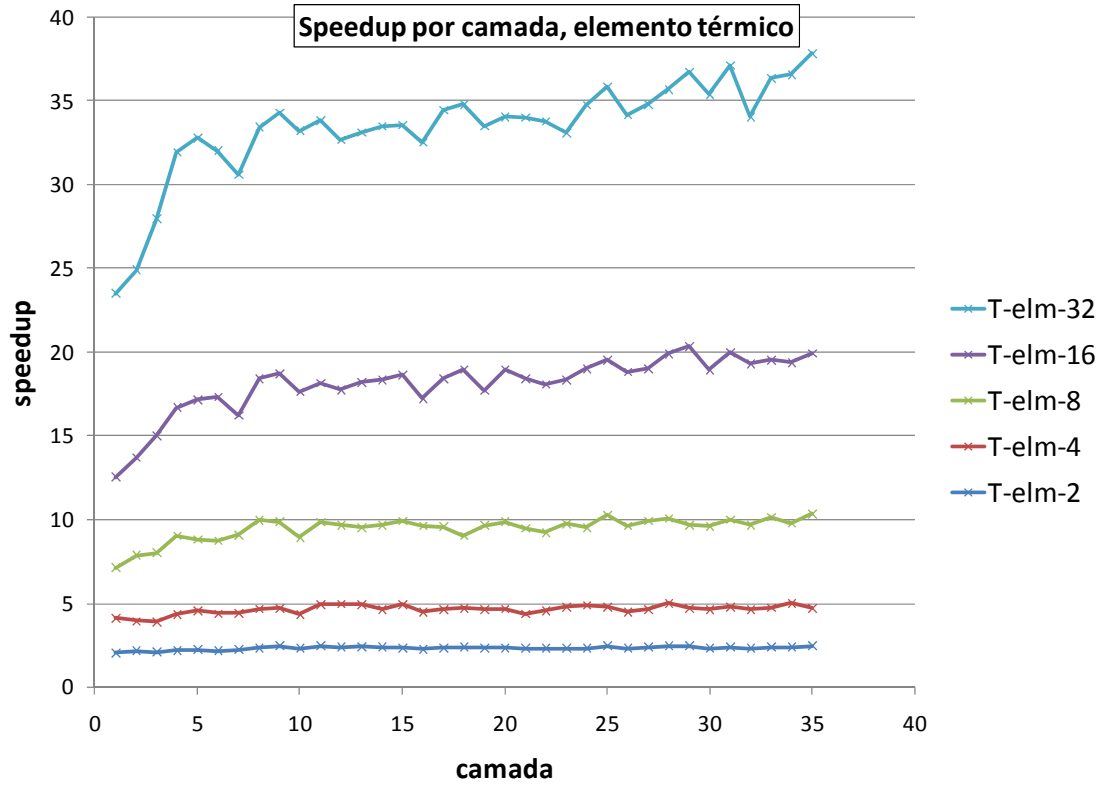


Figura 5.30: Speedups nos elementos, problema térmico, por camada.

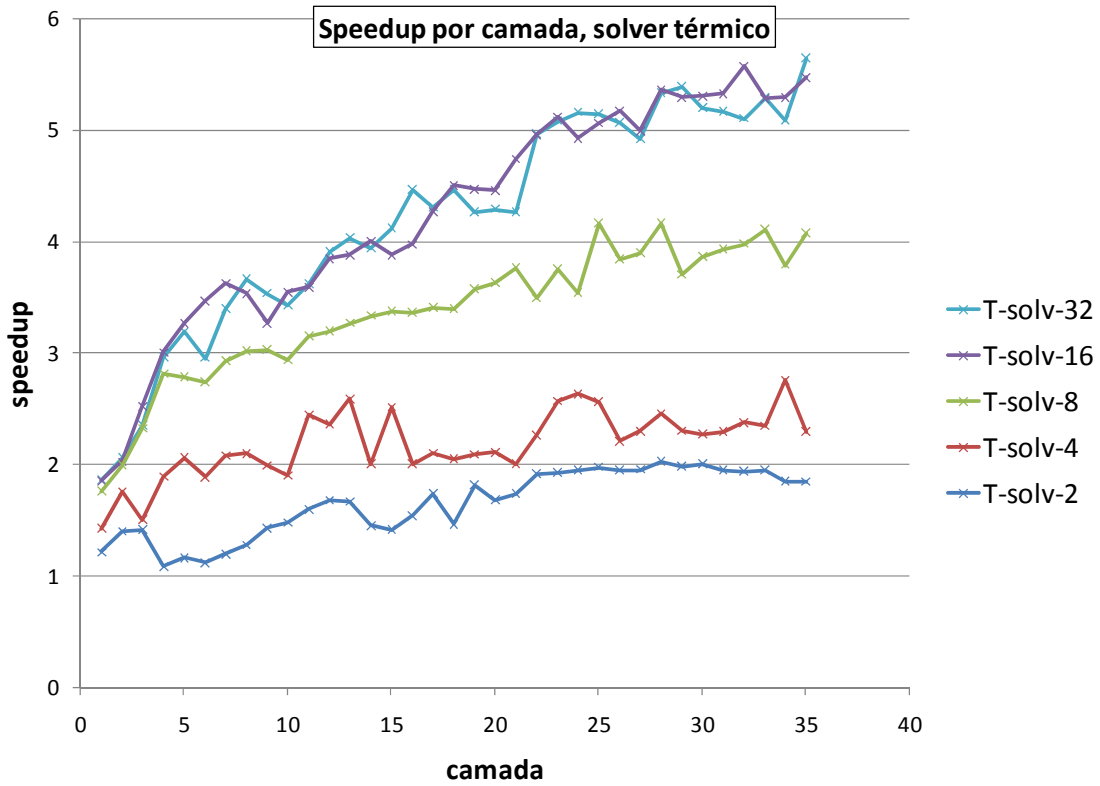


Figura 5.31: Speedups no solver PCGD, problema térmico, por camada.

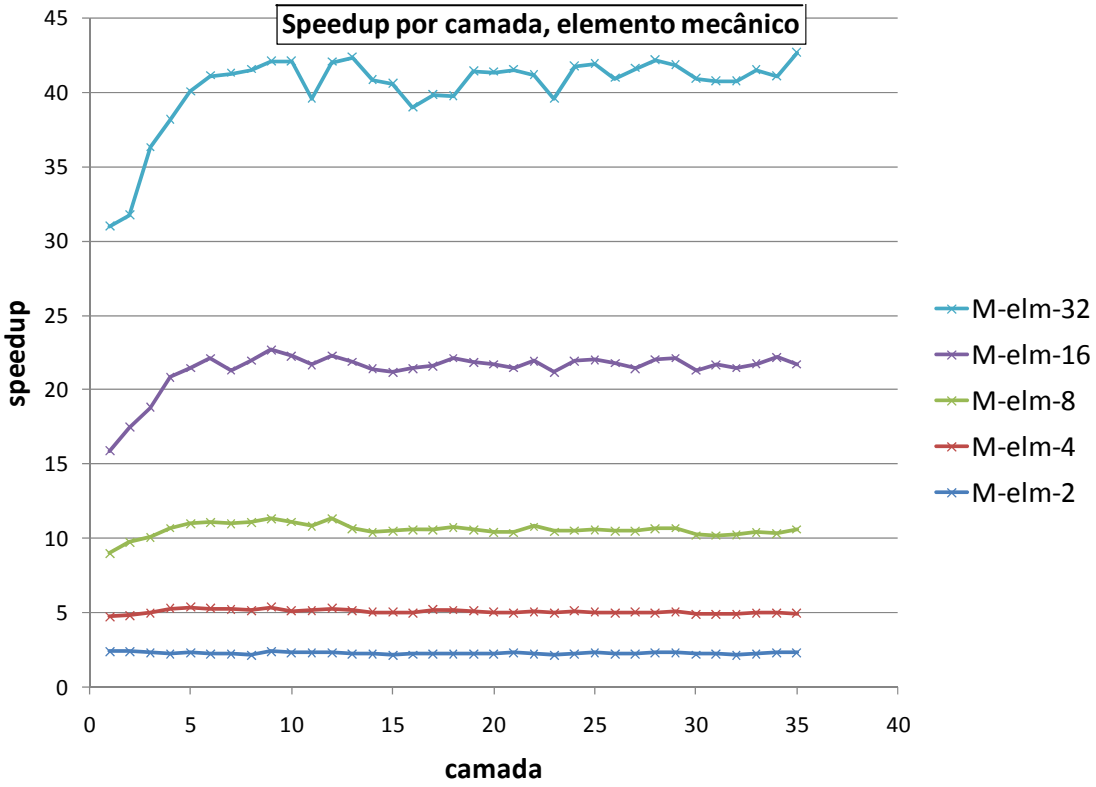


Figura 5.32: Speedups nos elementos, problema mecânico, por camada.

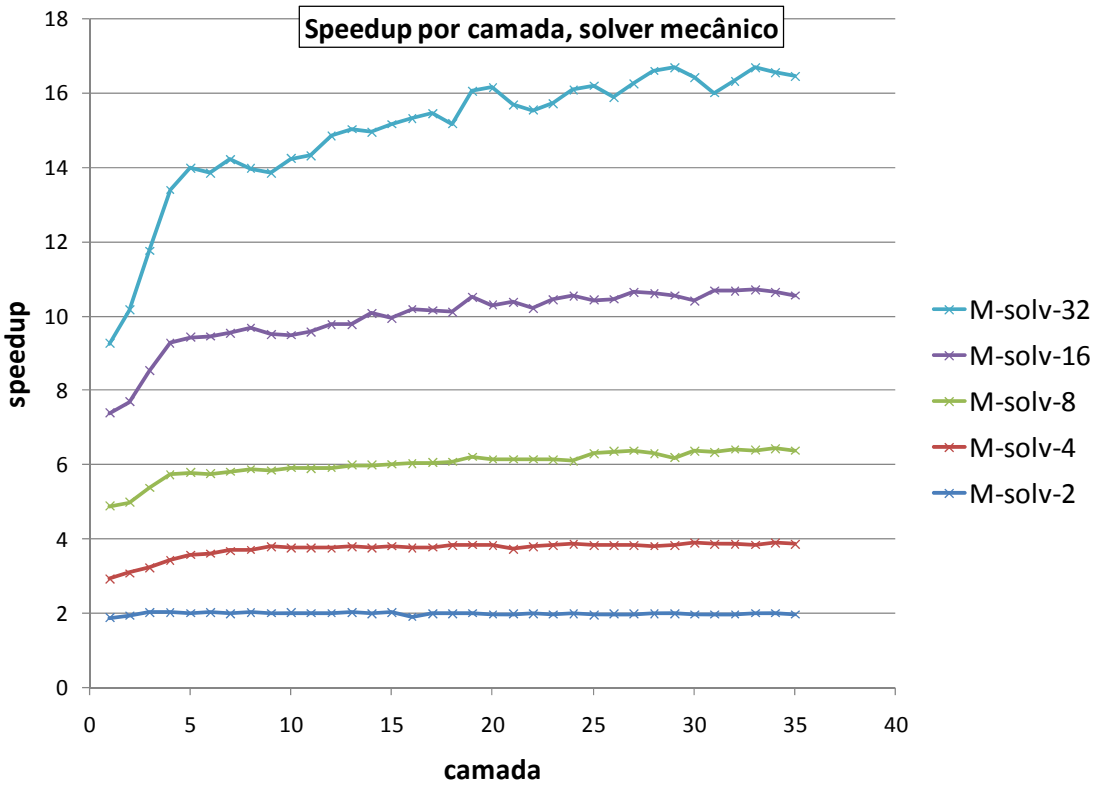


Figura 5.33: Speedups no solver PCGD, problema mecânico, por camada.

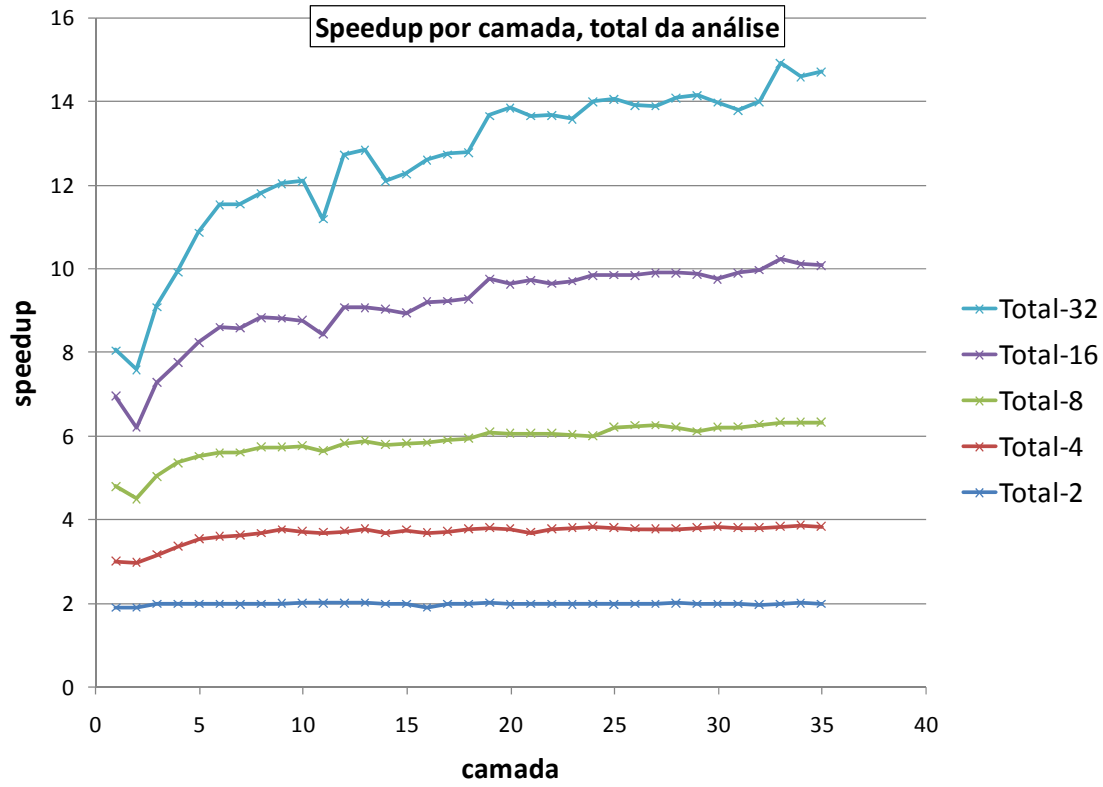


Figura 5.34: Speedups totais da análise por camada.

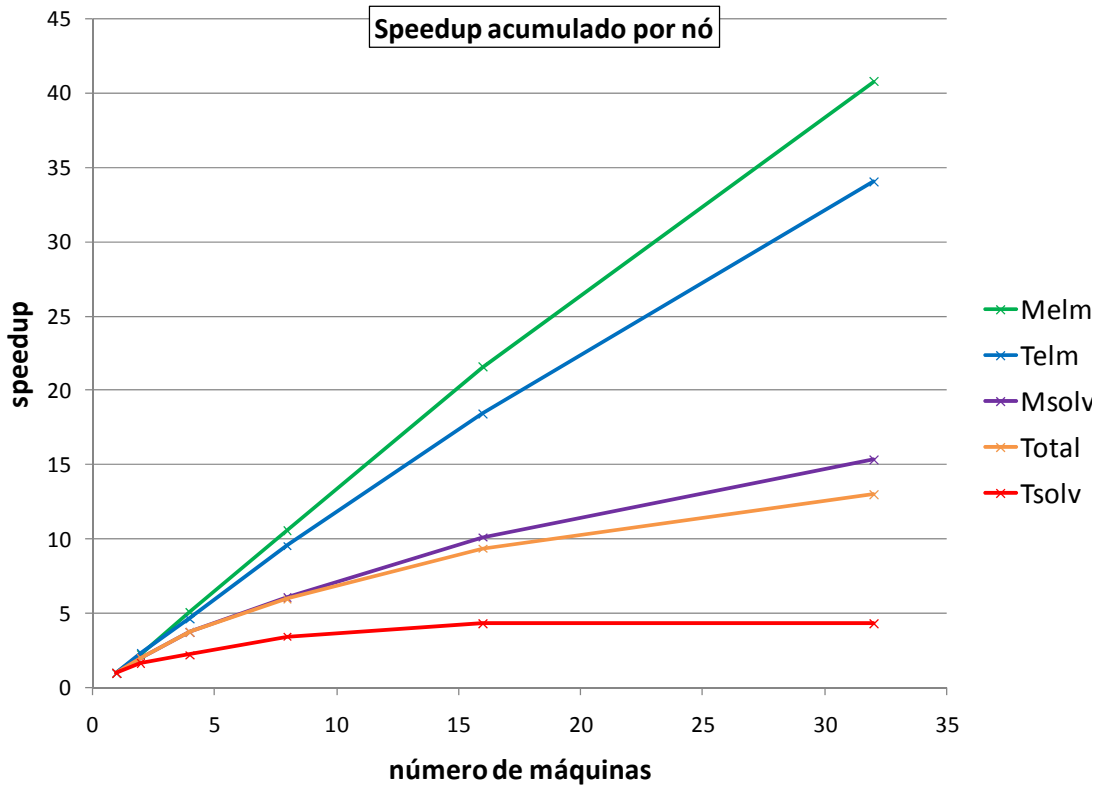


Figura 5.35: Speedups acumulados por nó.

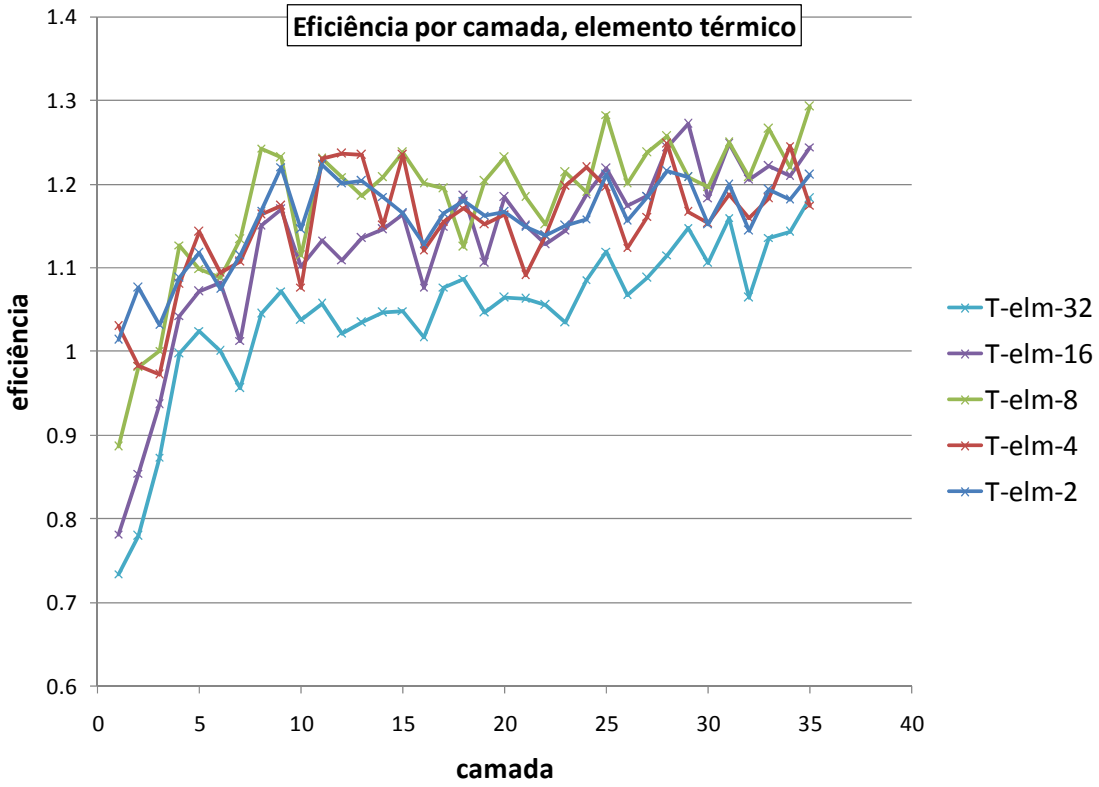


Figura 5.36: Eficiência nos elementos, problema térmico, por camada.

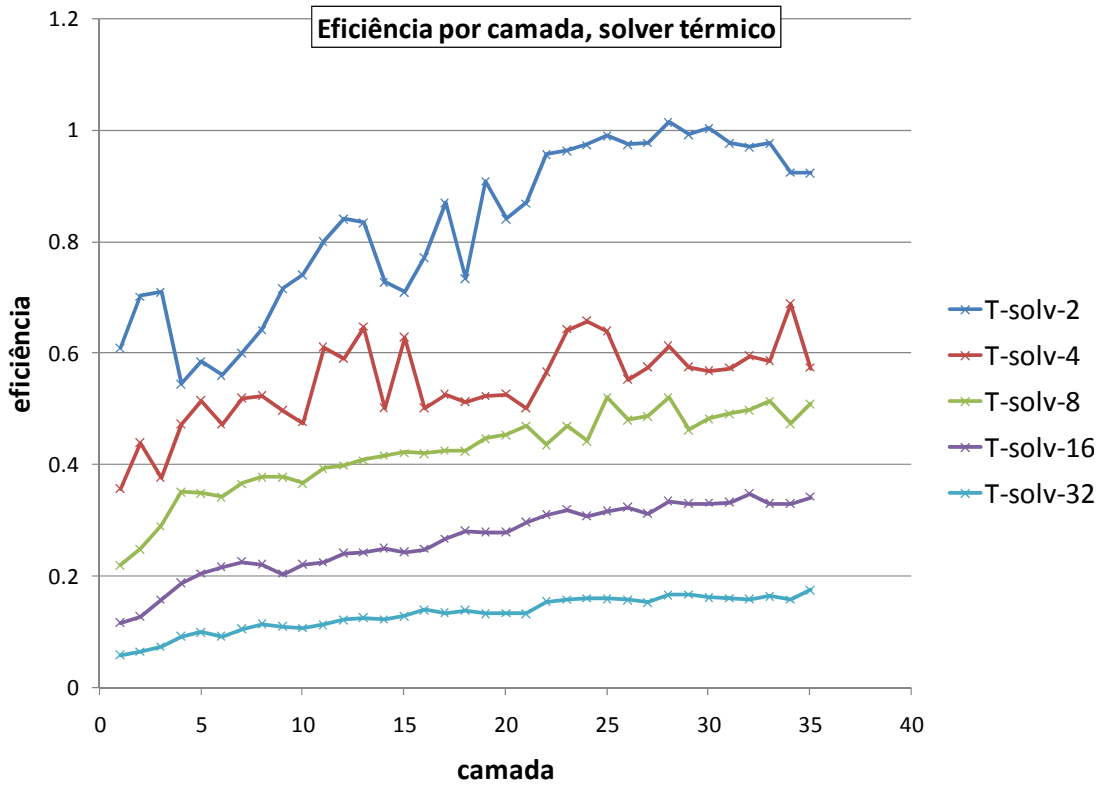


Figura 5.37: Eficiência no solver PCGD, problema térmico, por camada.

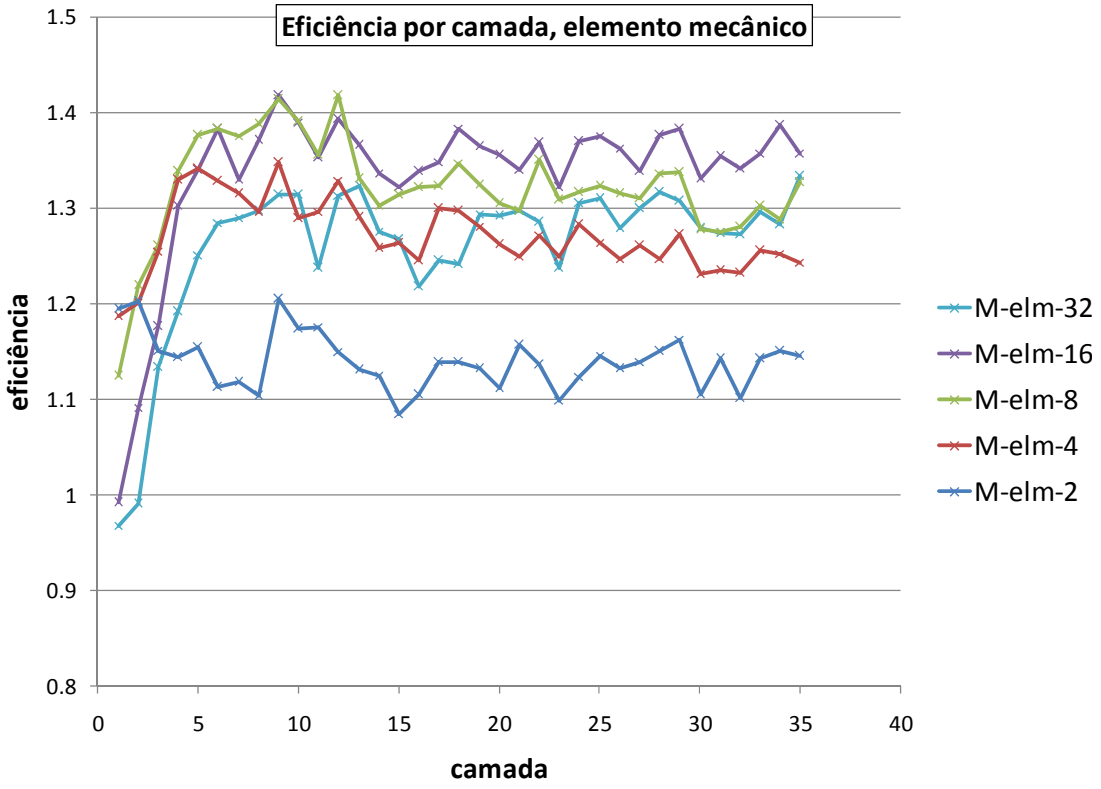


Figura 5.38: Eficiência nos elementos, problema mecânico, por camada.

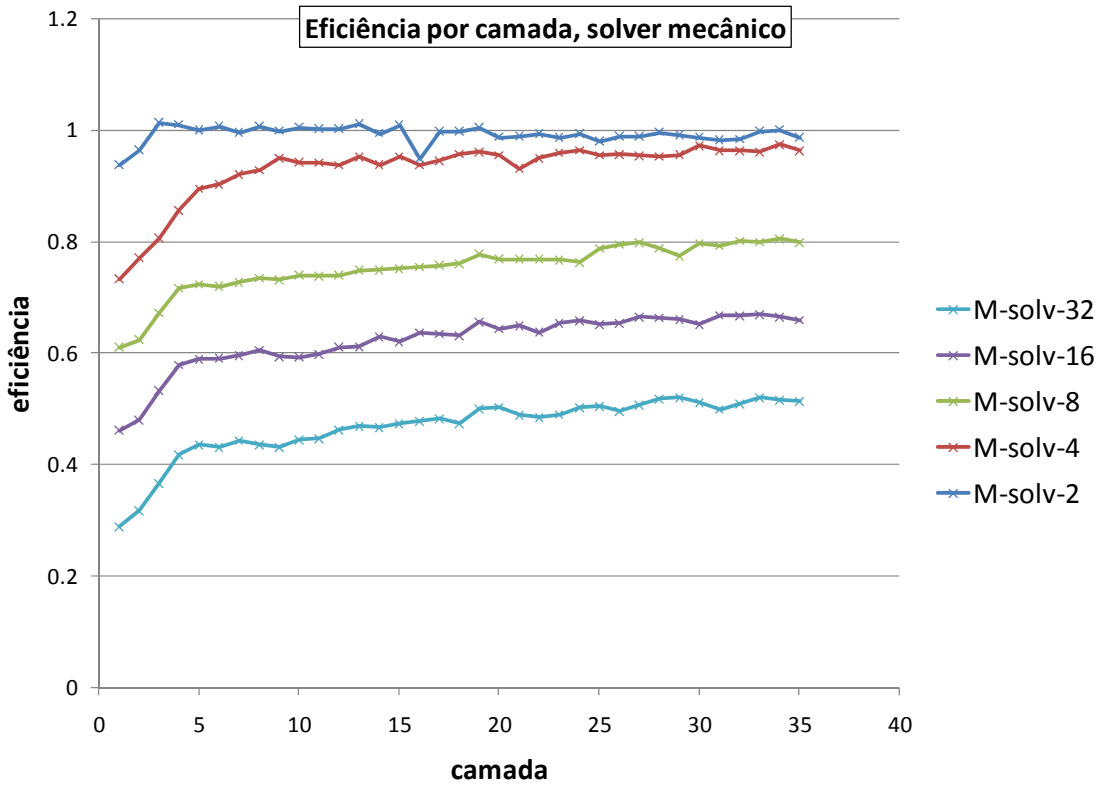


Figura 5.39: Eficiência no solver PCGD, problema mecânico, por camada.

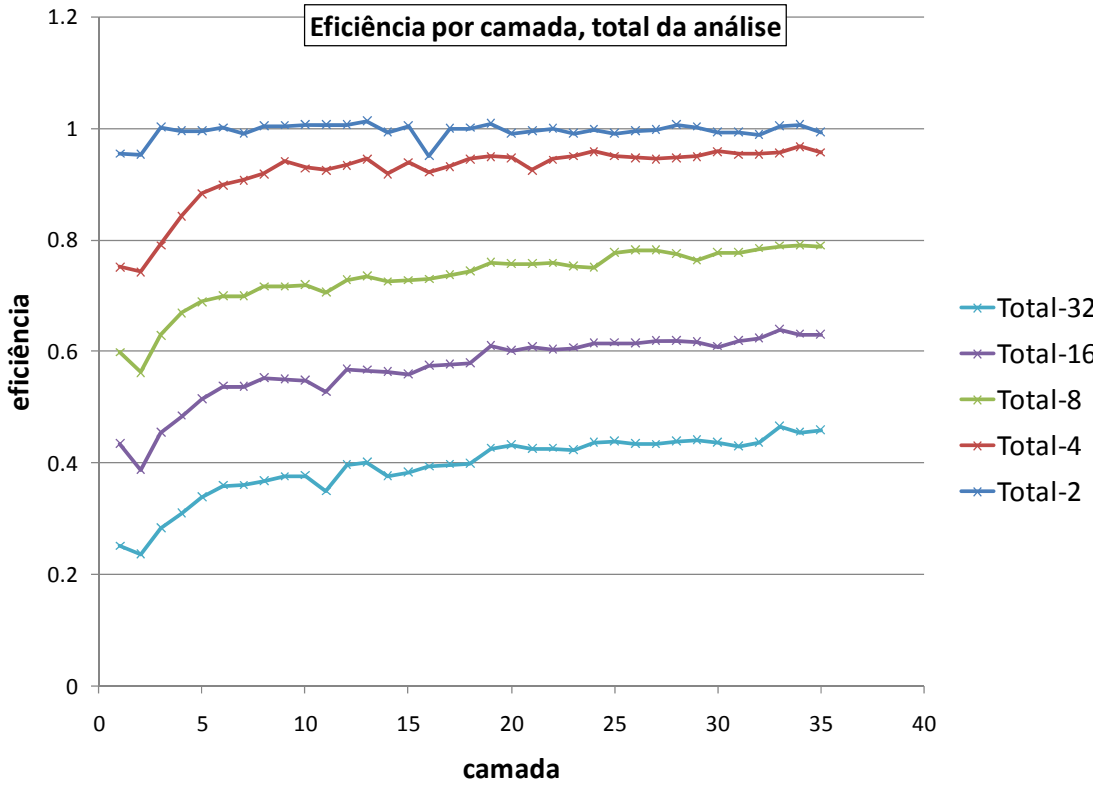


Figura 5.40: Eficiência total da análise por camada.

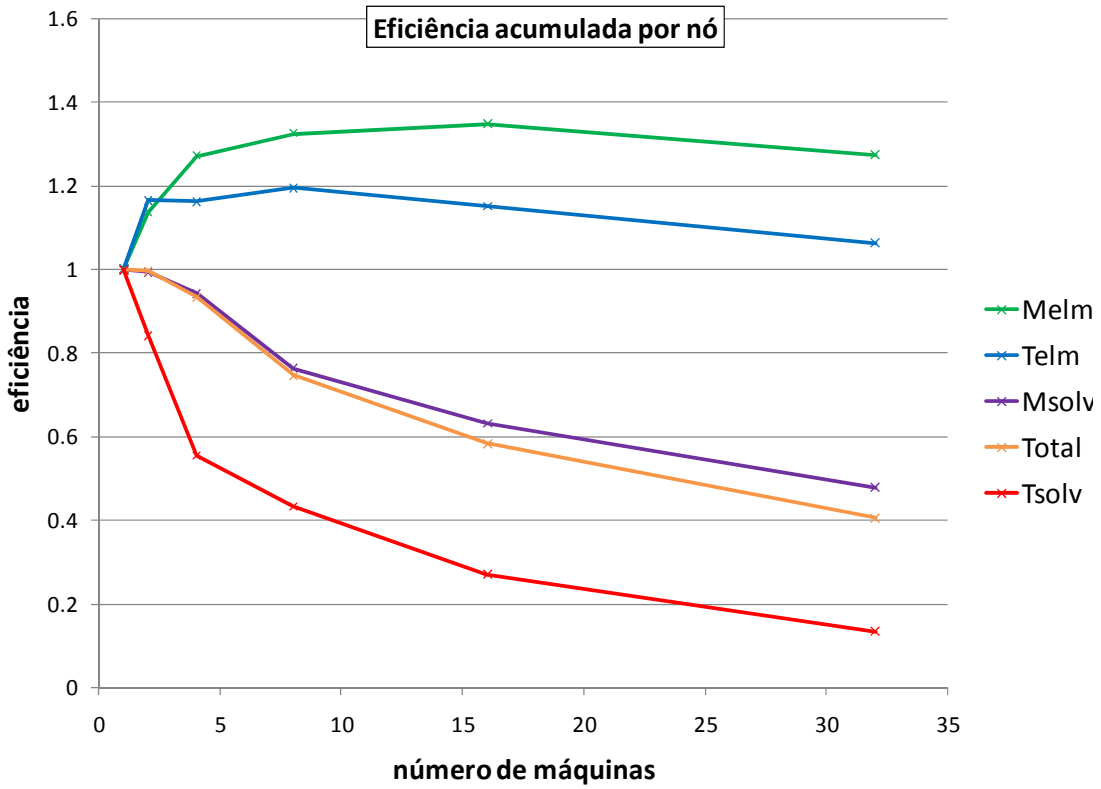


Figura 5.41: Eficiência acumulada por nó.

Nas Figuras a seguir as legendas maior, menor e média referem-se aos diversos processos da execução em paralelo. Cada processo tem sua taxa de comunicação dada por $neq^{i,f}/neq^{i}$, que varia de acordo com a quantidade de vizinhos do processo i ($nviz^{i}$) e com o tamanho da sua interface com estes vizinhos.

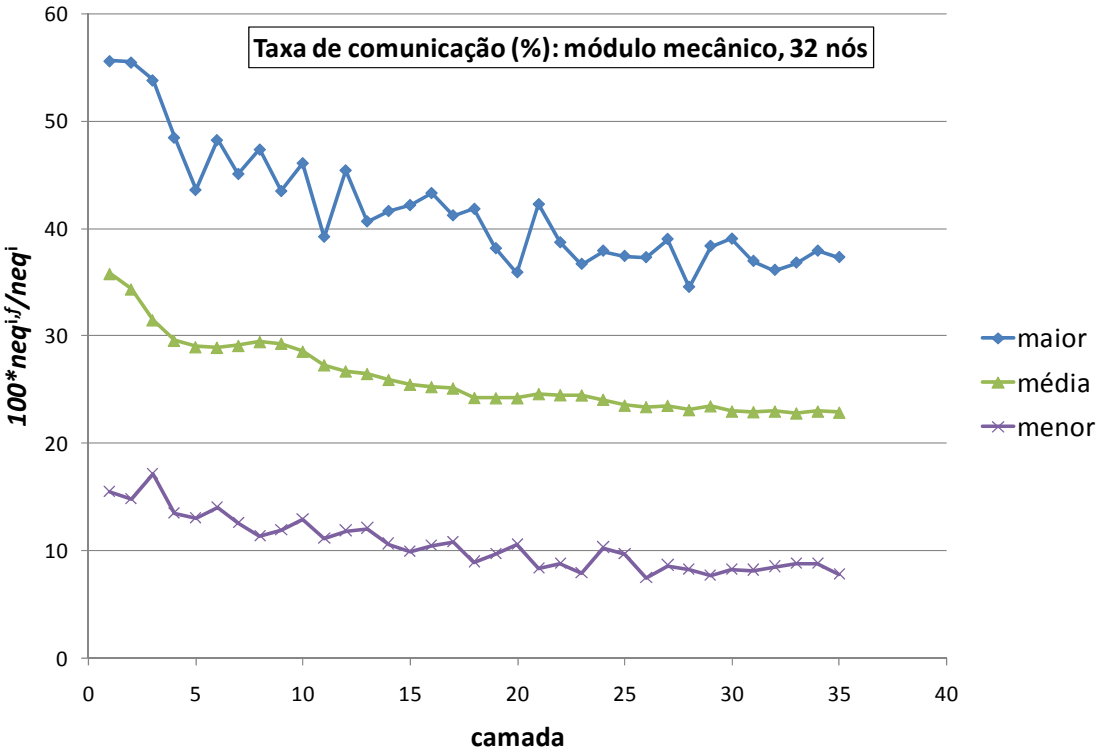


Figura 5.42: Taxa de comunicação por processo para 32 nós.

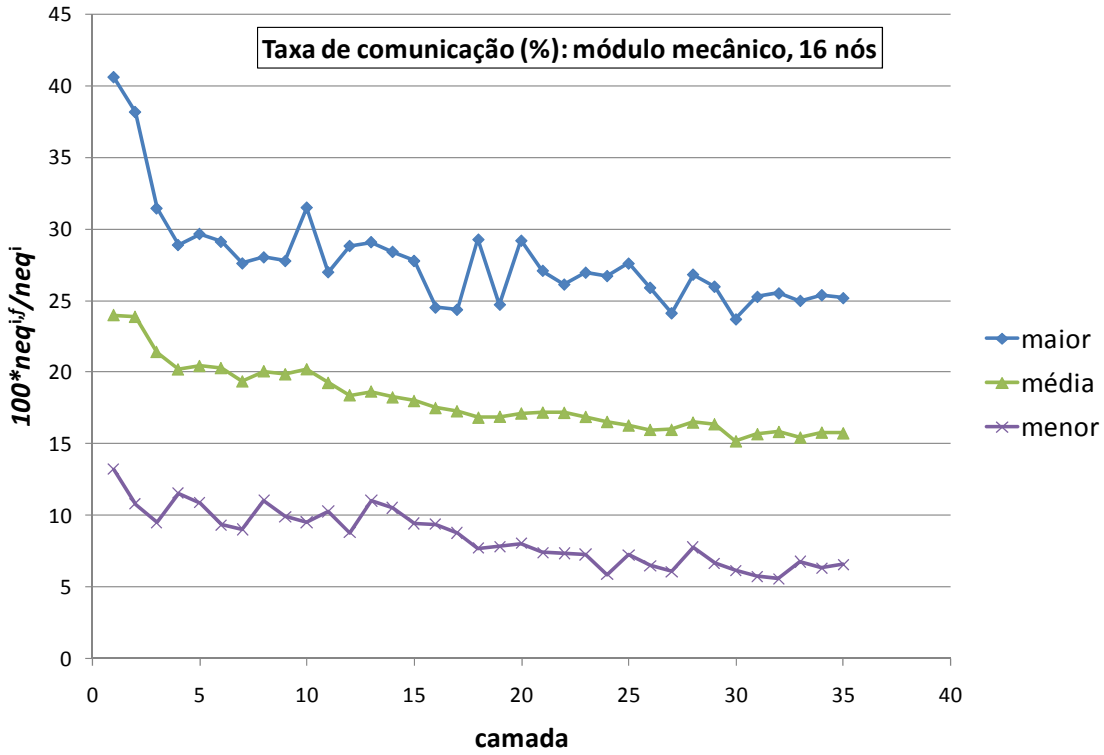


Figura 5.43: Taxa de comunicação por processo para 16 nós.

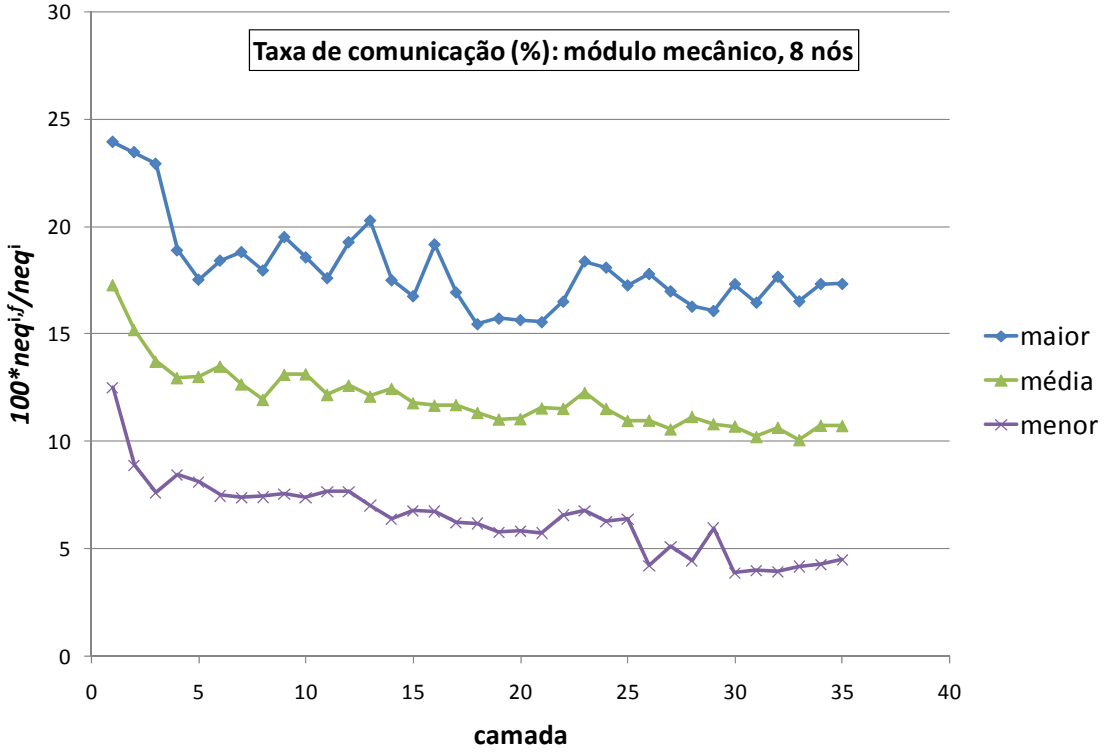


Figura 5.44: Taxa de comunicação por processo para 8 nós.

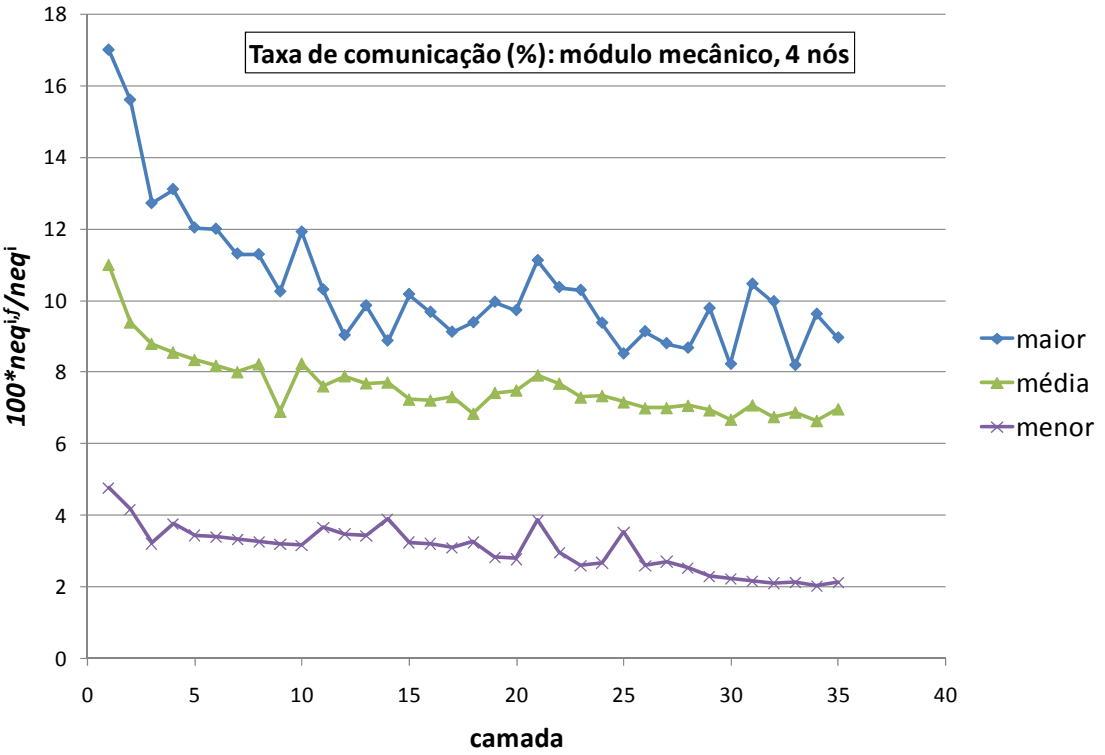


Figura 5.45: Taxa de comunicação por processo para 4 nós.

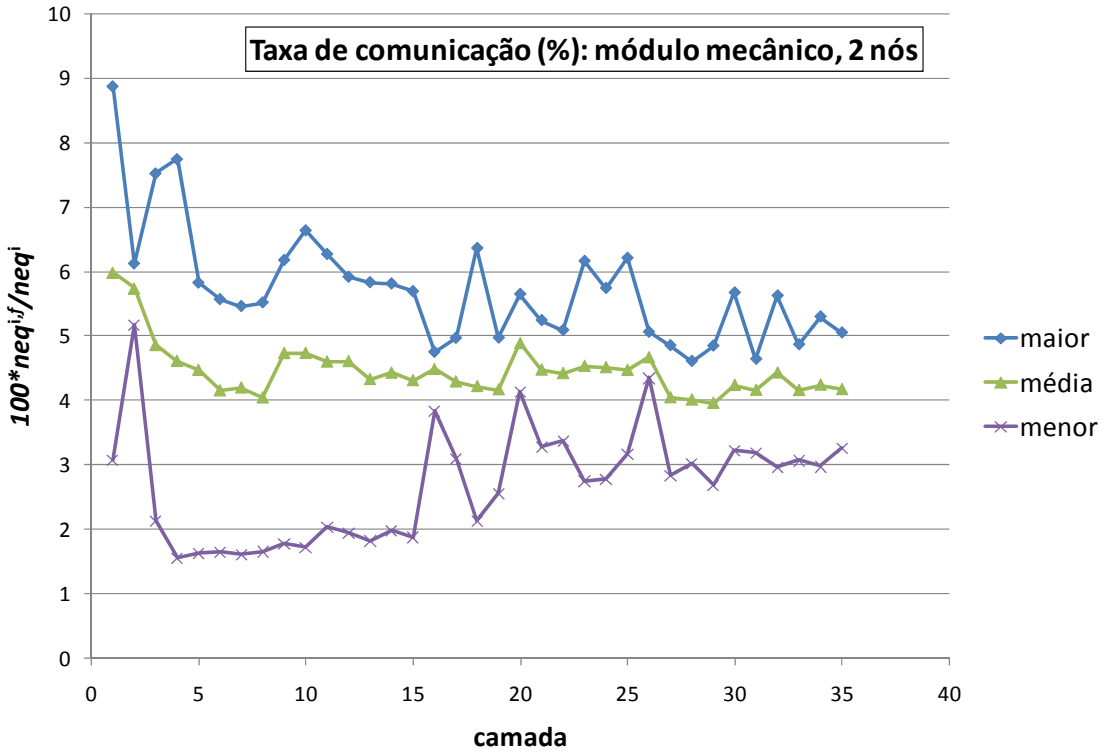


Figura 5.46: Taxa de comunicação por processo para 2 nós.

A observação das Figuras 5.16-29 permite identificar a similaridade entre os diversos gráficos de tempo por camada. Esta similaridade resulta da uniformidade da convergência na solução dos sistemas lineares, isto é, o *nit* não muda significativamente com a paralelização (n), fato apontado como diferencial dos métodos SBS em relação aos métodos DD. Aqui o termo significativamente indica que pode haver uma diferença de umas poucas iterações (uma ou duas) devido à alteração na ordem das OPF causada pela paralelização.

Os speedups de elemento por camada (Figuras 5.30 e 5.32) diferem pouco (cerca de 10%) entre os módulos térmico e mecânico. Já os speedups no solver por camada (Figuras 5.31 e 5.33) apresentam forte diferença em favor do módulo mecânico (3 vezes superior). Isto acontece porque o problema mecânico tem três graus de liberdade por nó, o que gera maiores valores de *neq* e *nad* para a mesma

malha, e é sabido que o speedup cresce com o tamanho do problema (Ribeiro e Ferreira 2007), pois o tempo de comunicação cresce menos, proporcionalmente, que o tempo de processamento (OPFs).

O speedup por nó (Figura 5.35) diminui com o aumento do número de nós. Isto se deve ao aumento da taxa de comunicação à medida que se particiona mais e mais a malha, como pode ser visto pela comparação entre as diversas Figuras de taxa de comunicação (5.42-46). Cada uma destas Figuras (5.42-46) mostra haver uma correspondência entre a diminuição da taxa de comunicação e o aumento do speedup no solver observado nas Figuras 5.31 e 5.33, à medida que a malha cresce com o lançamento das camadas.

A eficiência da paralelização na montagem de elementos (Figuras 5.36 e 5.38) não se altera muito com o número de nós (máquinas) utilizado (note que as curvas são bem agrupadas). Isto se deve ao fato desta operação não envolver comunicação. Já o solver PCGD tem sua eficiência fortemente reduzida com o aumento da paralelização (n) (ver Figuras 5.37 e 5.39).

6 SIMULAÇÃO DE UMA CASA DE FORÇA

Caso a barragem seja uma também uma usina hidrelétrica, ela terá uma casa de força. Nesta estrutura ficam as turbinas que convertem energia mecânica (potencial) da água em energia elétrica. Por abrigar e suportar as turbinas esta estrutura está sujeita a grandes esforços, que são majorados pelas vibrações do conjunto quando em funcionamento.

6.1 ESTRUTURA

As figuras a seguir mostram aspectos da estrutura criada para simular a construção de uma casa de força real (rio Tocoma, na Venezuela).

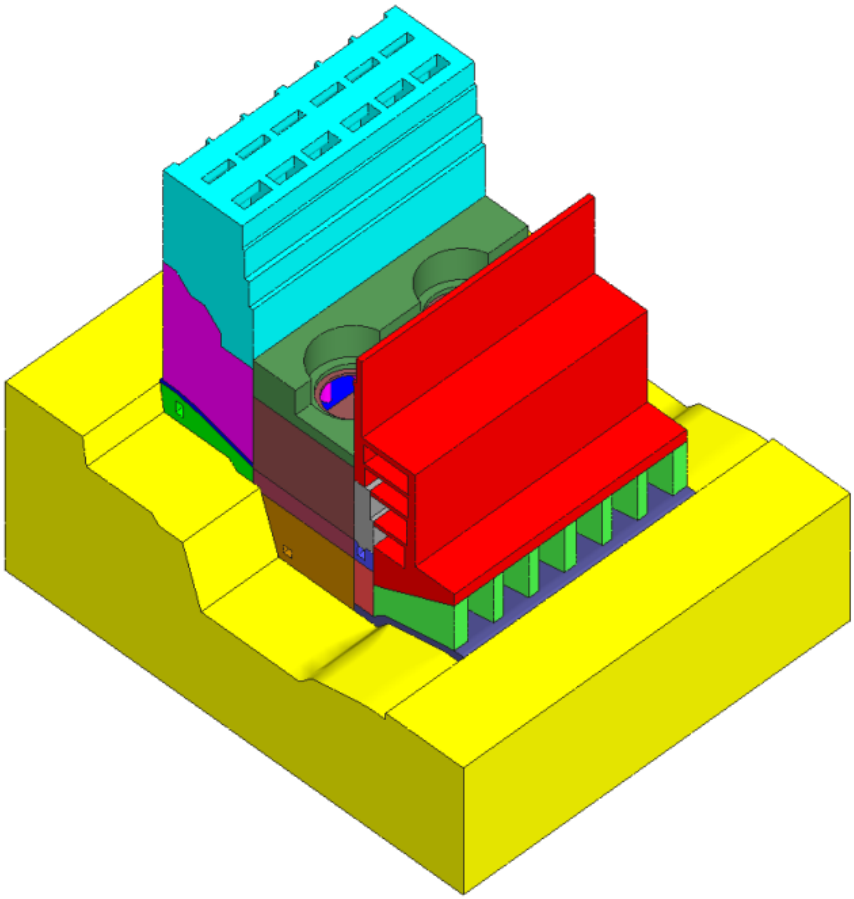


Figura 6.1: Geometria da cada de força: vista isométrica.

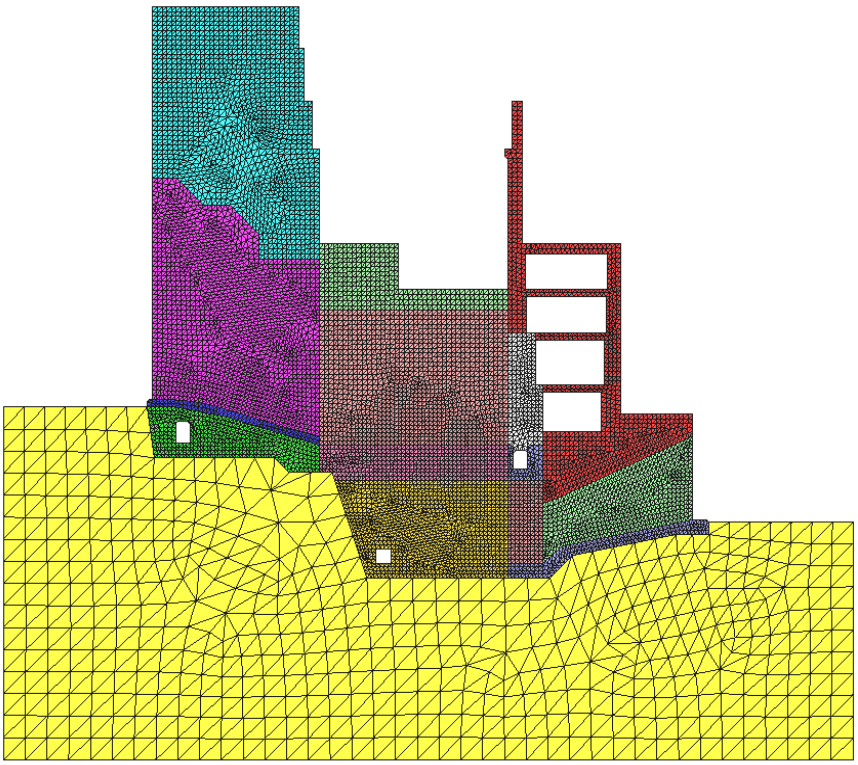


Figura 6.2: Malha da cada de força: vista lateral.

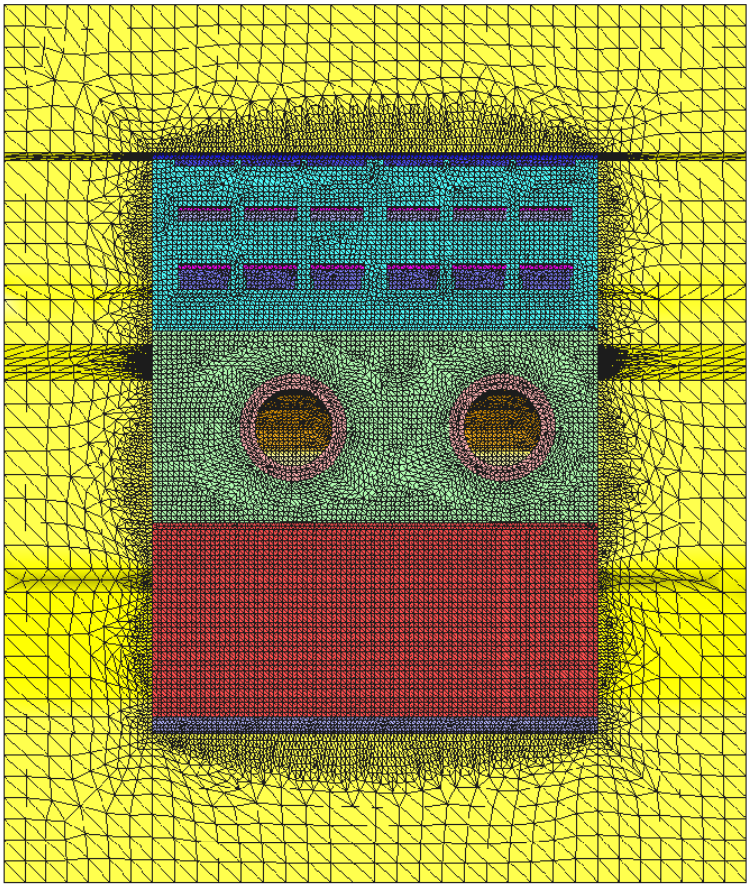


Figura 6.3: Malha da cada de força: vista superior.

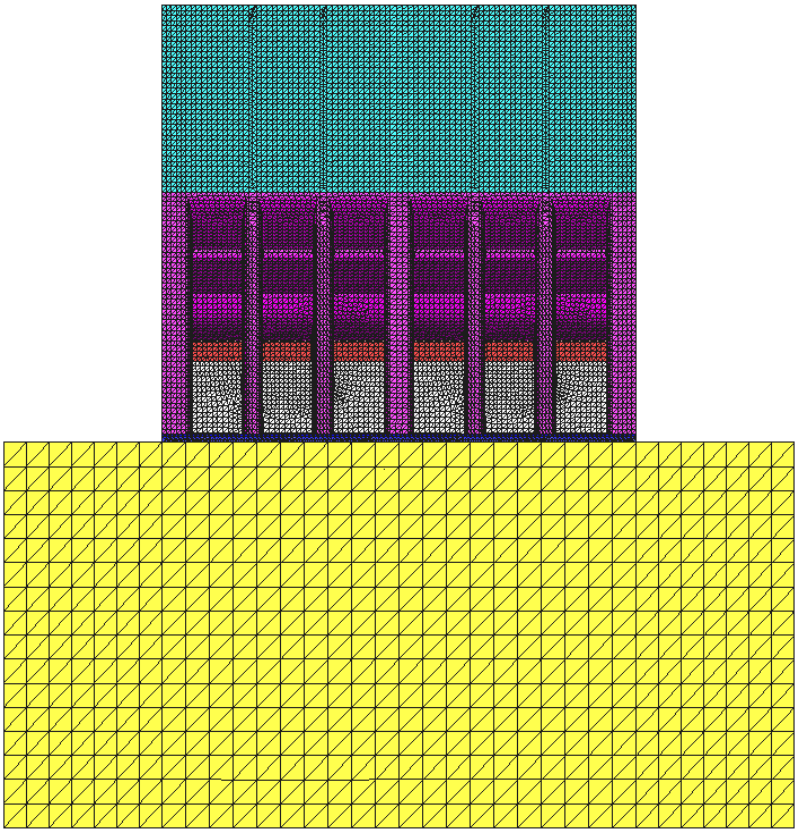


Figura 6.4: Malha da cada de força: vista de montante.

6.2 MATERIAIS

Apenas um material foi considerado, cujas propriedades estão arroladas na Tabela 6.1. Nesta análise não foi considerada plasticidade. Foram consideradas fluência e retração autógena. A variação das propriedades com o grau de hidratação foi adotada como sendo uma função tipo rampa.

Tabela 6.1: Propriedades de materiais: casa de força.

	concreto	solo
E_∞ [GPa]	28,049	40
H [Pa]	0,035	
J_∞^v [Pa^{-1}]	11*10^{-12}	
Γ_0 [Pa^{-1}dd^{-1}]	4	
$\tau_{w,\infty}$ [h]	26	
β [1]	5*10^{-5}	
α [K^{-1}]	9,06*10^{-6}	11*10^{-6}
c_e [J/KgK]	1140	1140
k [W/mK]	2,7	4
ρ [Kg/m^3]	2330	2760

Foi adotado coeficiente de Poisson v = 0,2 para os dois materiais.

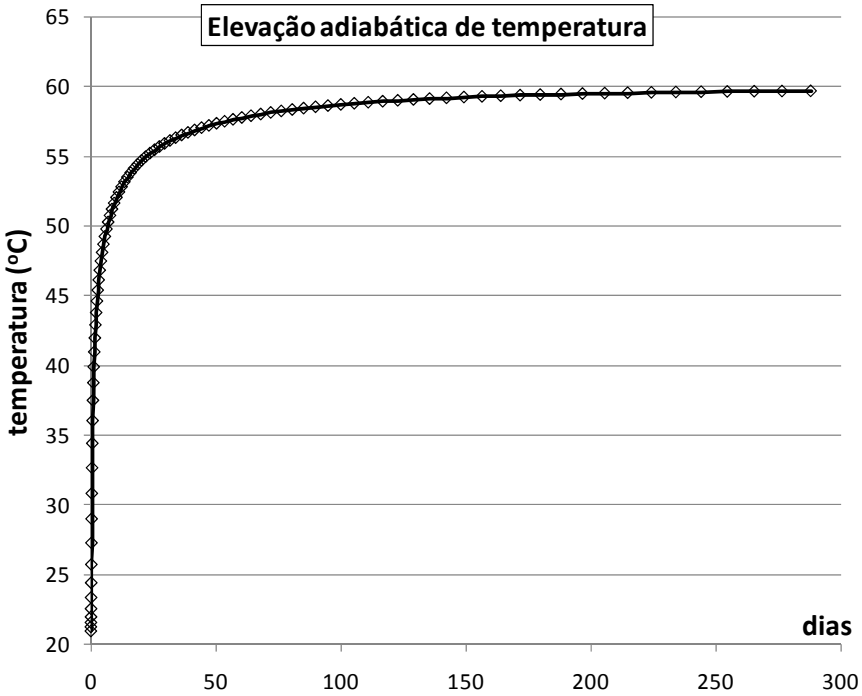


Figura 6.5: Curva de elevação adiabática de temperatura.

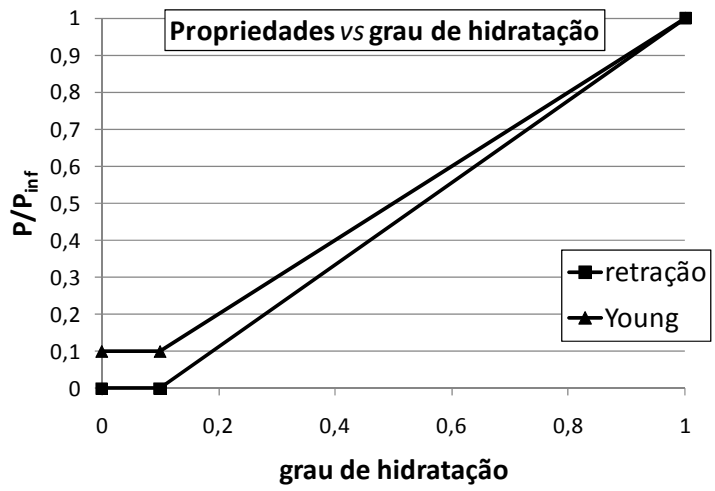


Figura 6.6: Propriedades *vs* grau de hidratação, casa de força.

6.3 RESULTADOS

A Figura 6.7 mostra campos de temperatura em um corte no sentido montante-jusante passando por pilares a montante e a jusante e pela cavidade de escoamento. A Figura 6.8 mostra resultados do índice de fissuração obtidos para alguns instantes selecionados na simulação. O índice de fissuração mencionado é a razão entre a tensão principal máxima de tração e a resistência à tração $f_t(\xi)$ desenvolvida pelo material em um determinado ponto da estrutura em um determinado instante de tempo na simulação. Este índice é utilizado como parâmetro para indicar regiões críticas quanto à fissuração por tração, uma vez que o modelo de plasticidade não foi usado nesta análise.

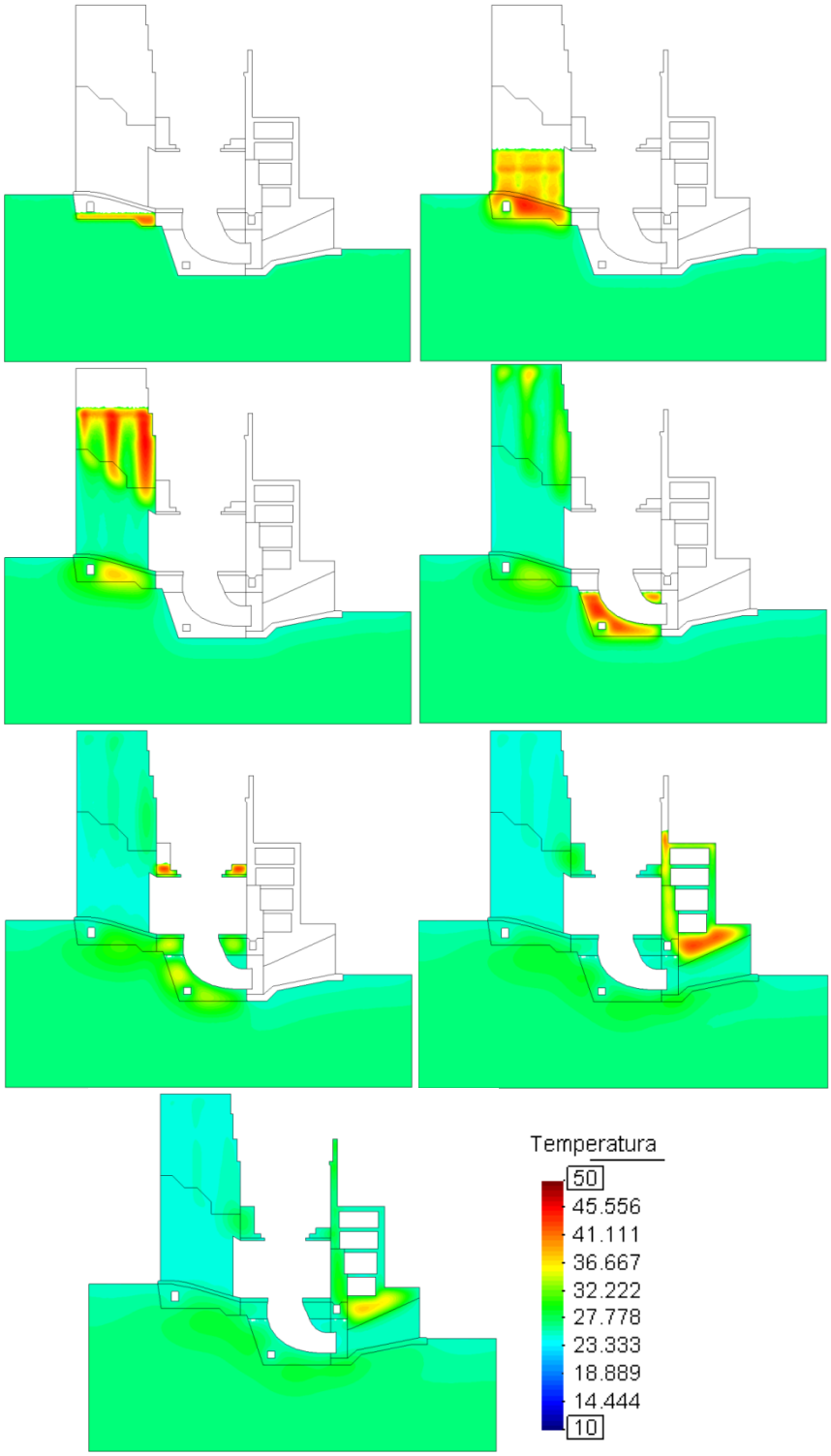


Figura 6.7: Temperatura em corte longitudinal da casa de força.

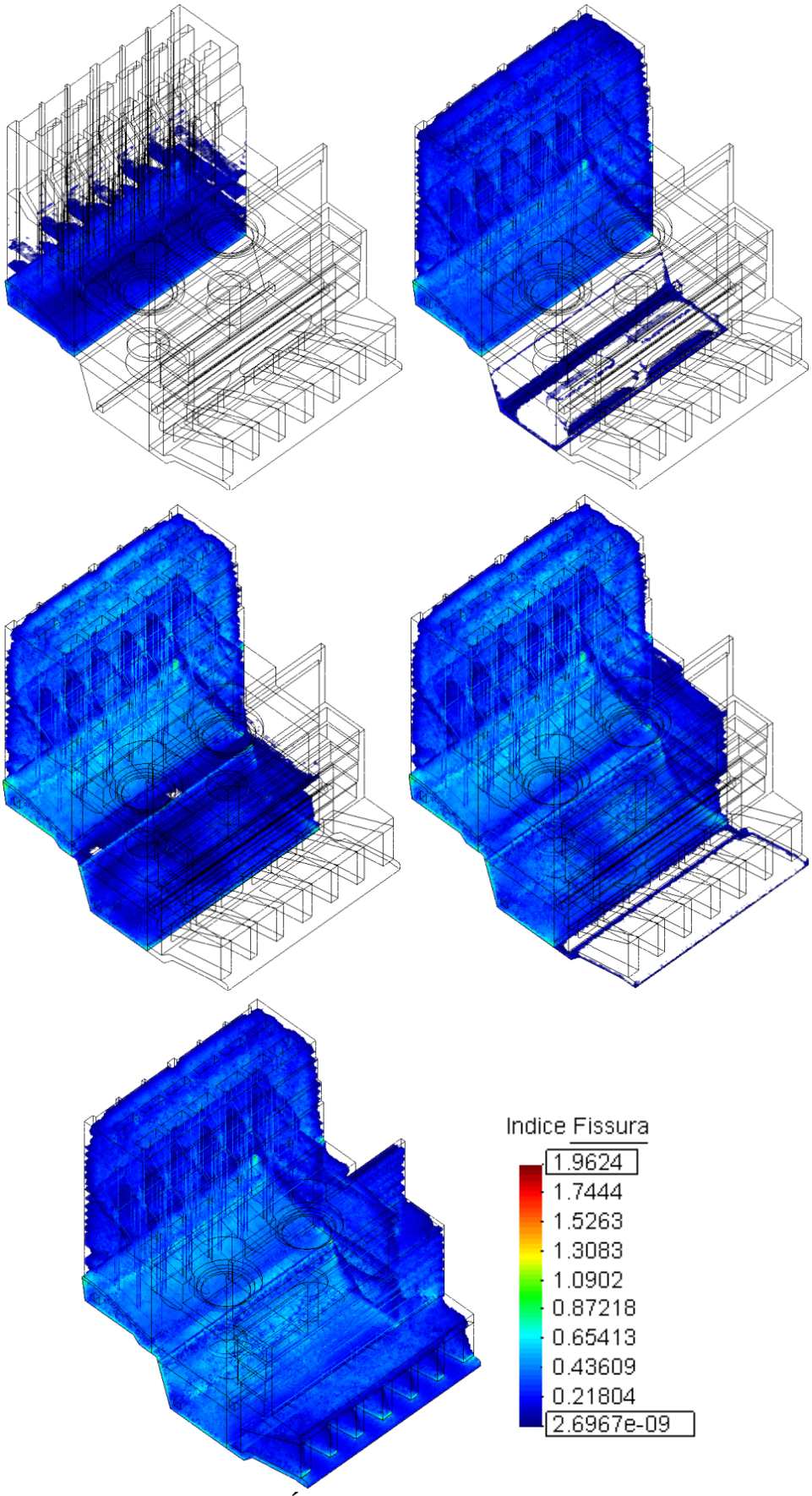


Figura 6.8: Índice de fissuração ($\sigma_1/f_t(\xi)$).

6.4 DESEMPENHO

As figuras a seguir mostram curvas de tempo de execução (Wall clock time), Speedup (de acordo com (4.20)) e eficiência (de acordo com (4.21)b) variando ao longo da evolução (crescimento) do domínio. O problema foi resolvido em um cluster de PCs cujas características estão na Tabela 5.2. a simulação de 391 dias em 7227 passos de tempo de tamanho variável correspondeu a 13243 soluções PCGD do problema térmico e a 29167 soluções PCGD do problema mecânico, devido às não-linearidades nos dois problemas. As Figuras 6.3-4 mostram respectivamente a quantidade de coeficientes da matriz $\{al\}$ armazenados e o número de equações crescendo com o avanço das camadas de construção.

A mesma notação do capítulo anterior é adotada neste quanto à legenda das curvas nas Figuras a seguir.

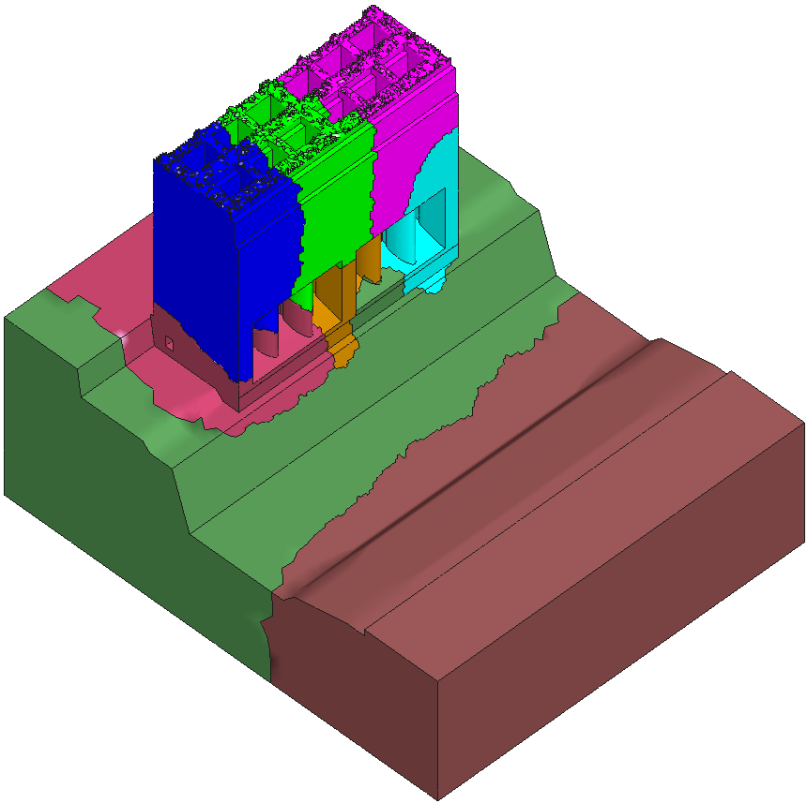


Figura 6.9: Particionamentos com 8 processos, camada 20.

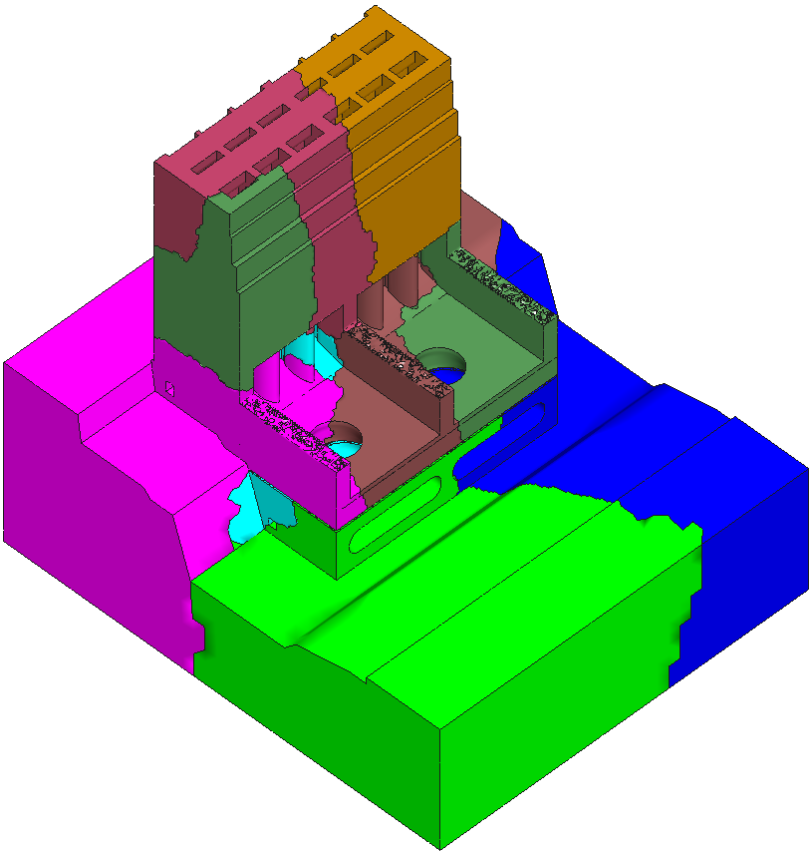


Figura 6.10: Particionamentos com 8 processos, camada 40.

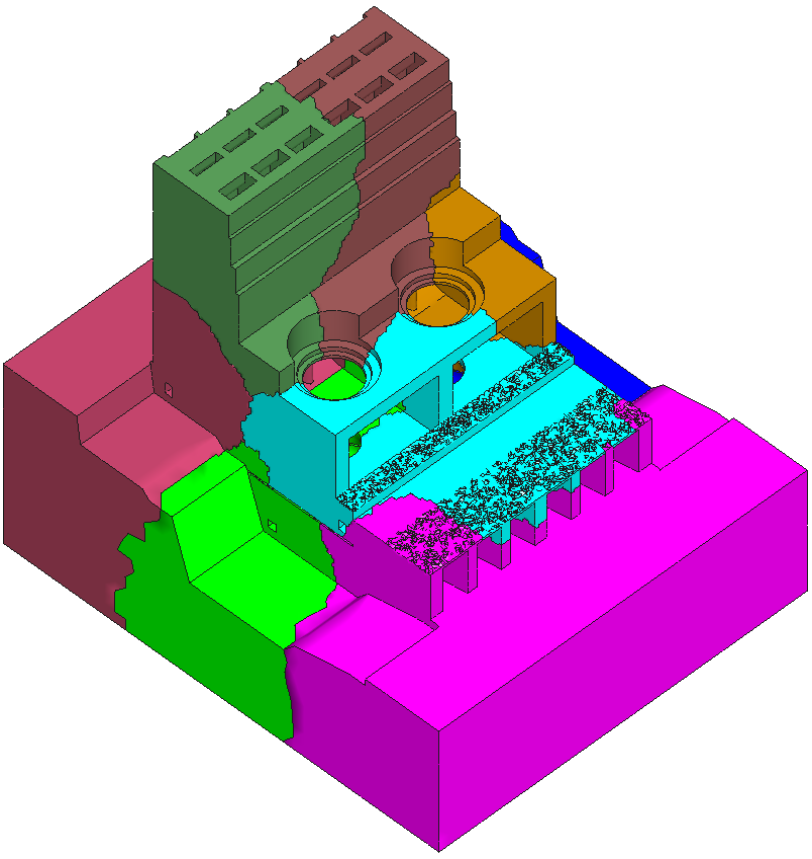


Figura 6.11: Particionamentos com 8 processos, camada 60.

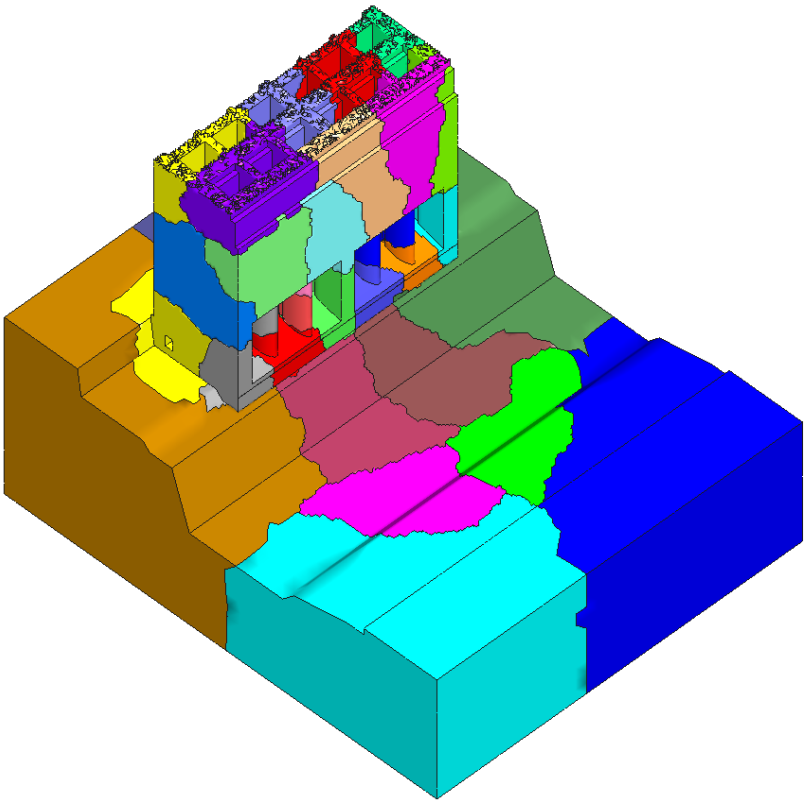


Figura 6.12: Particionamentos com 32 processos, camada 20.

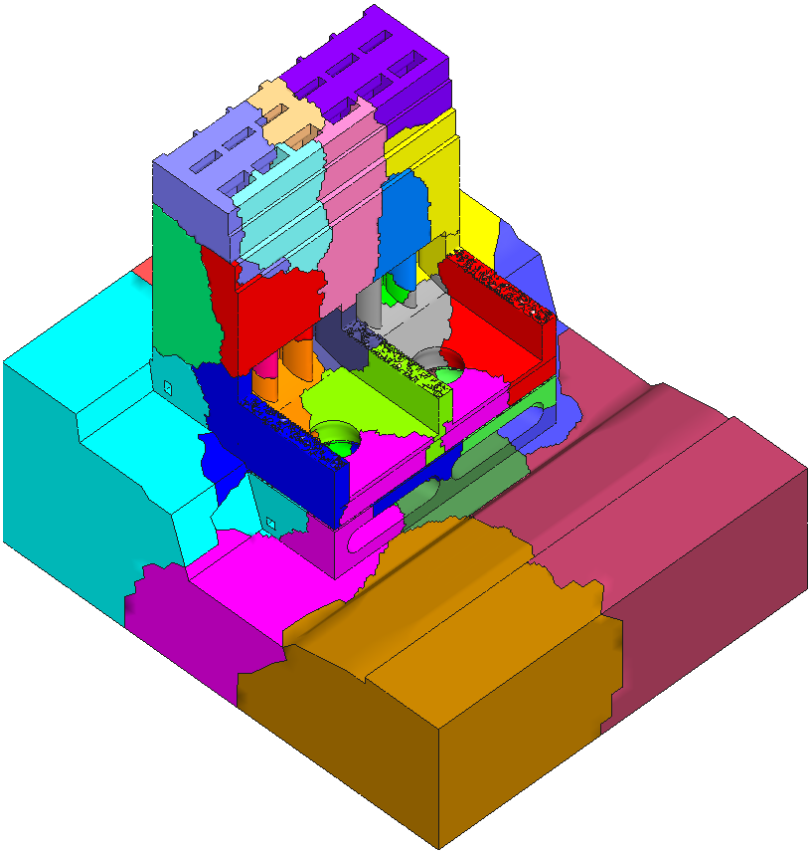


Figura 6.13: Particionamentos com 32 processos, camadas 40.

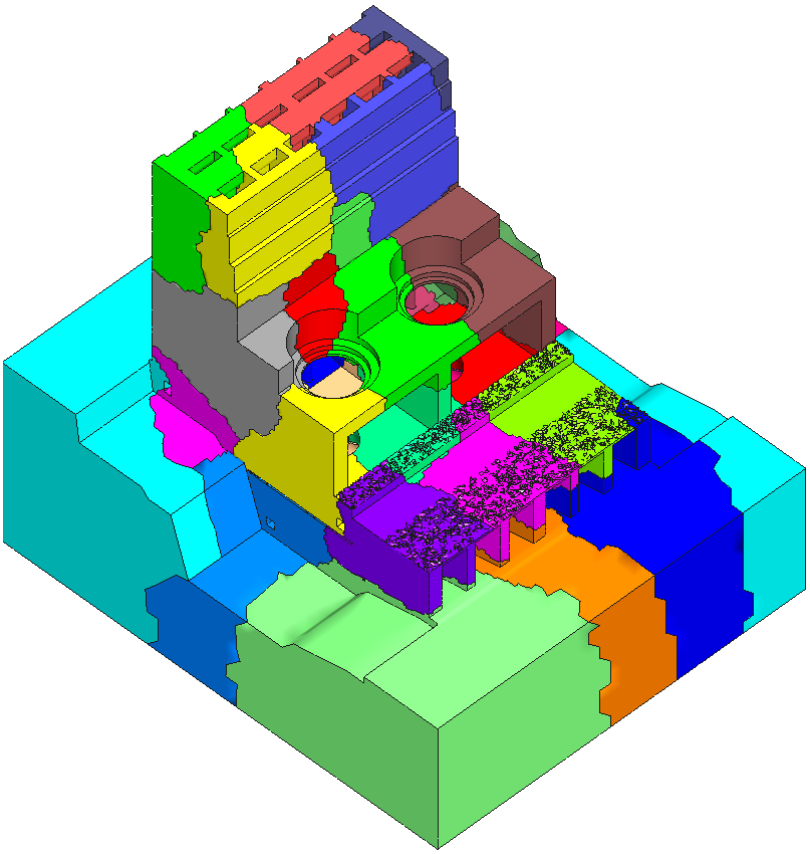


Figura 6.14: Particionamentos com 32 processos, camada 60.

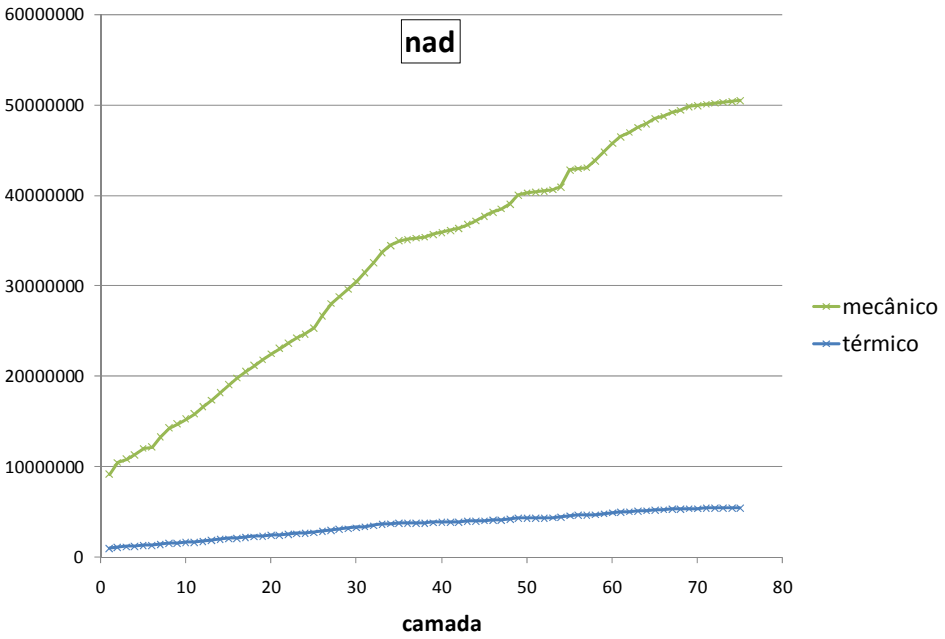


Figura 6.15: Número de coeficientes da matriz armazenados (nad) por camada.

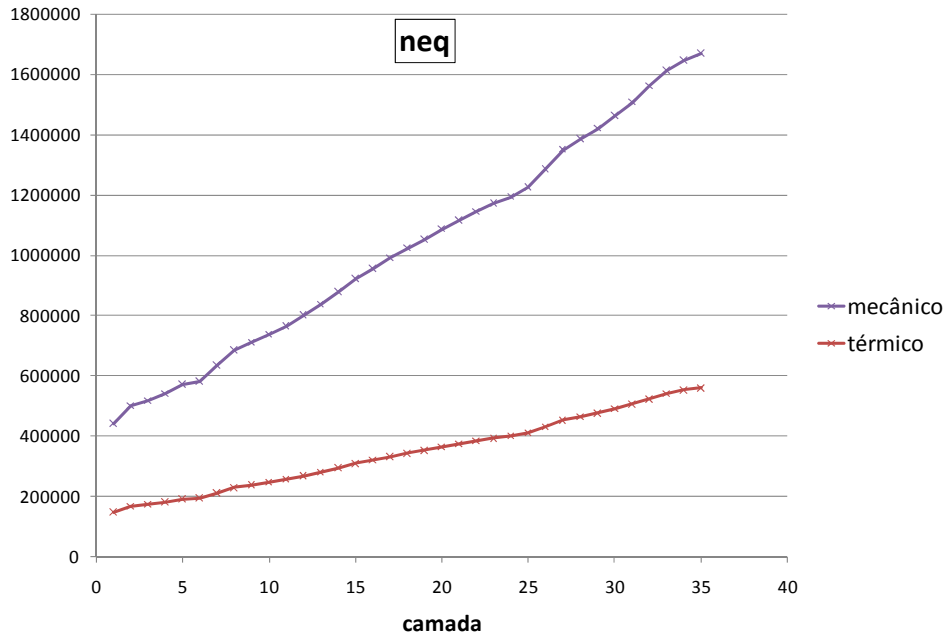


Figura 6.16: Número de equações (neq) por camada.

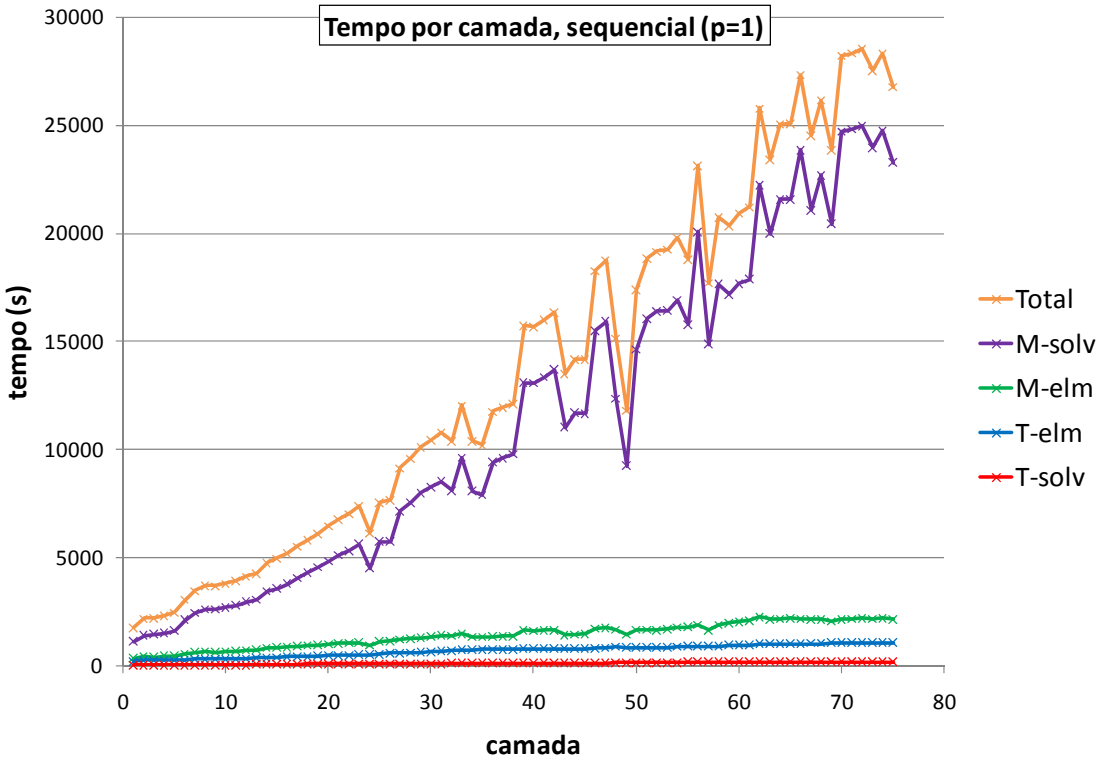


Figura 6.17: Tempo seqüencial (n=1, p=1) por camada [s].

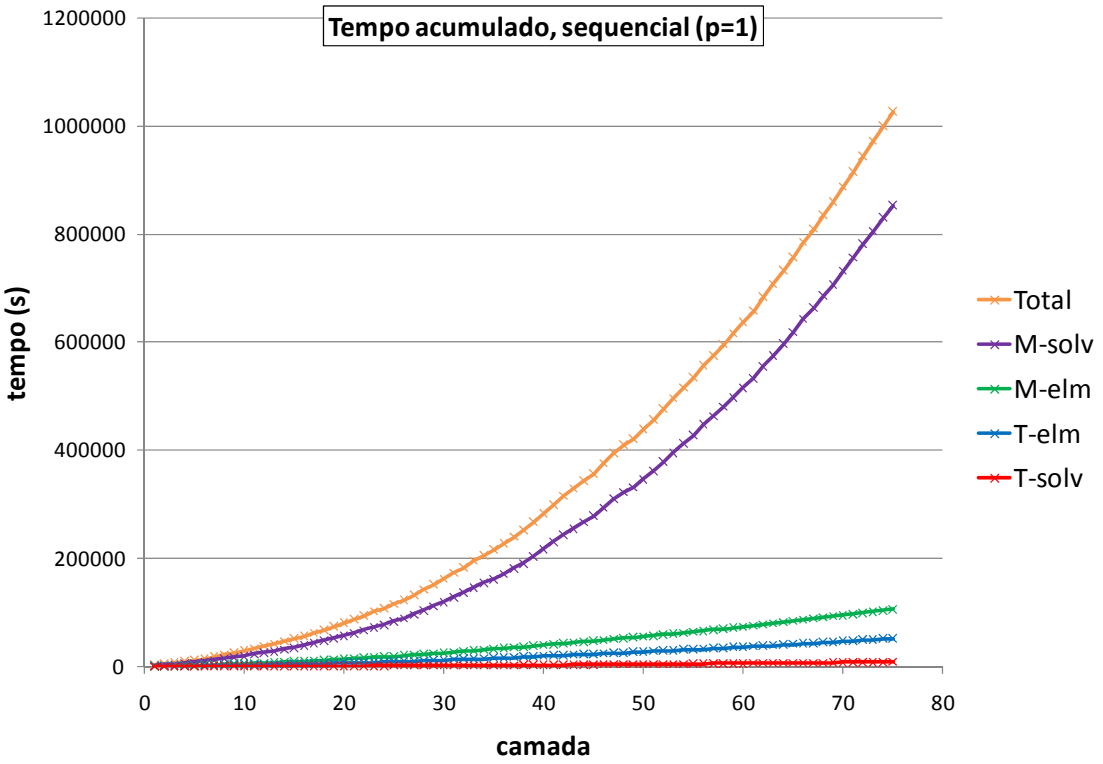


Figura 6.18: Tempo seqüencial (n=1, p=1) acumulado [s].

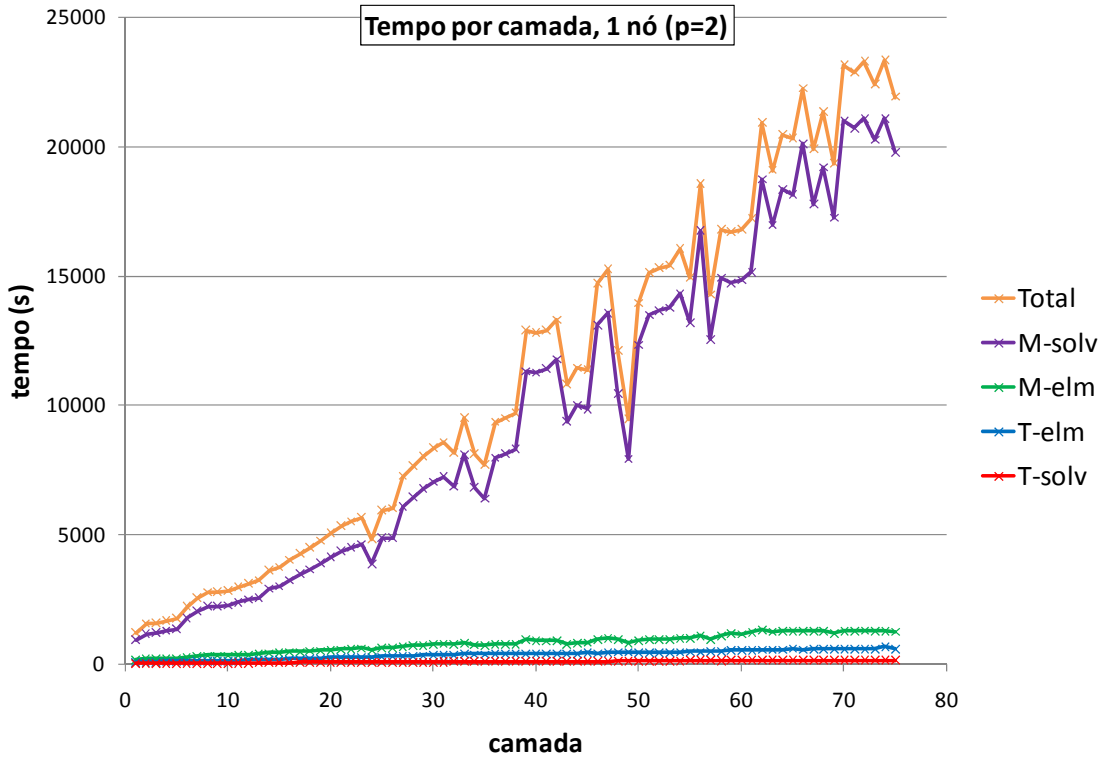


Figura 6.19: Tempo paralelo (n=1, p=2) por camada [s].

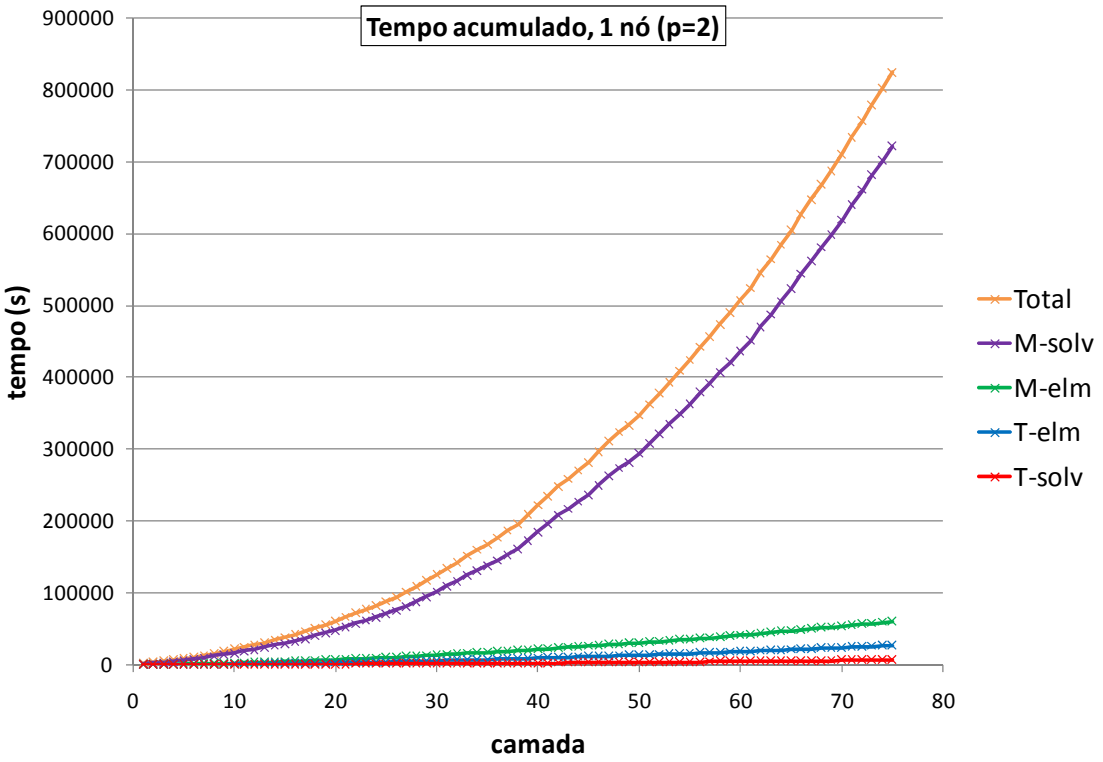


Figura 6.20: Tempo paralelo (n=1, p=2) acumulado [s].

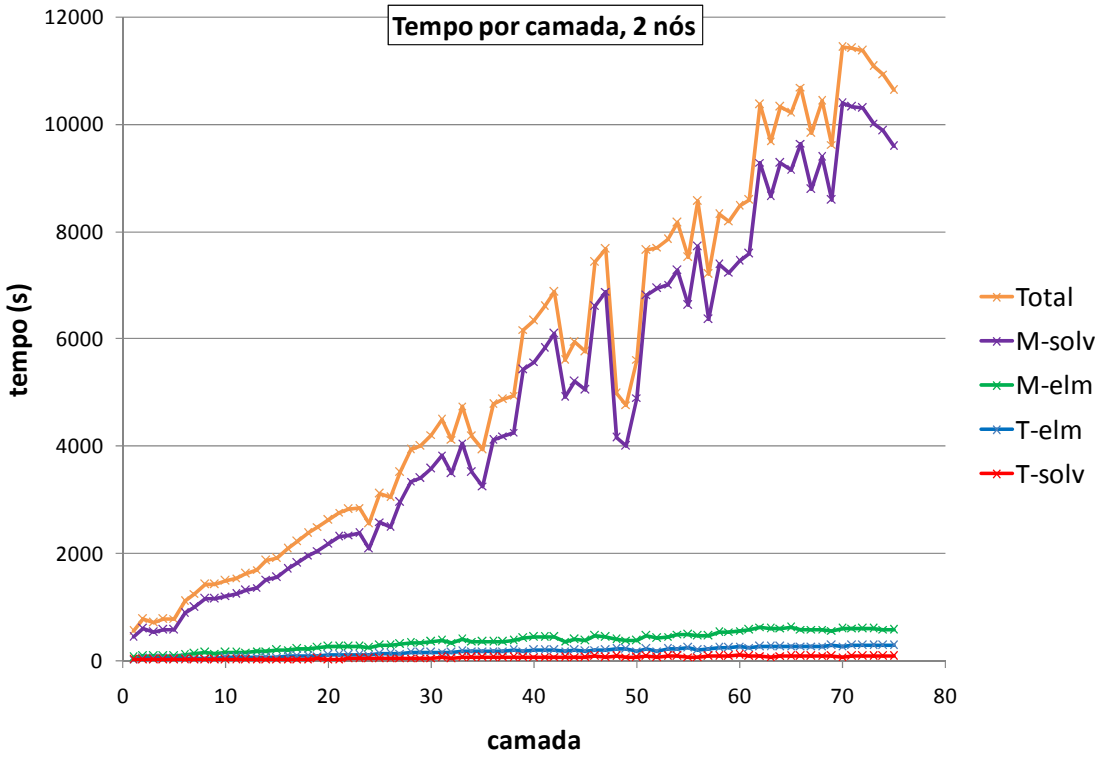


Figura 6.21: Tempo paralelo (n=2, p=4) por camada [s].

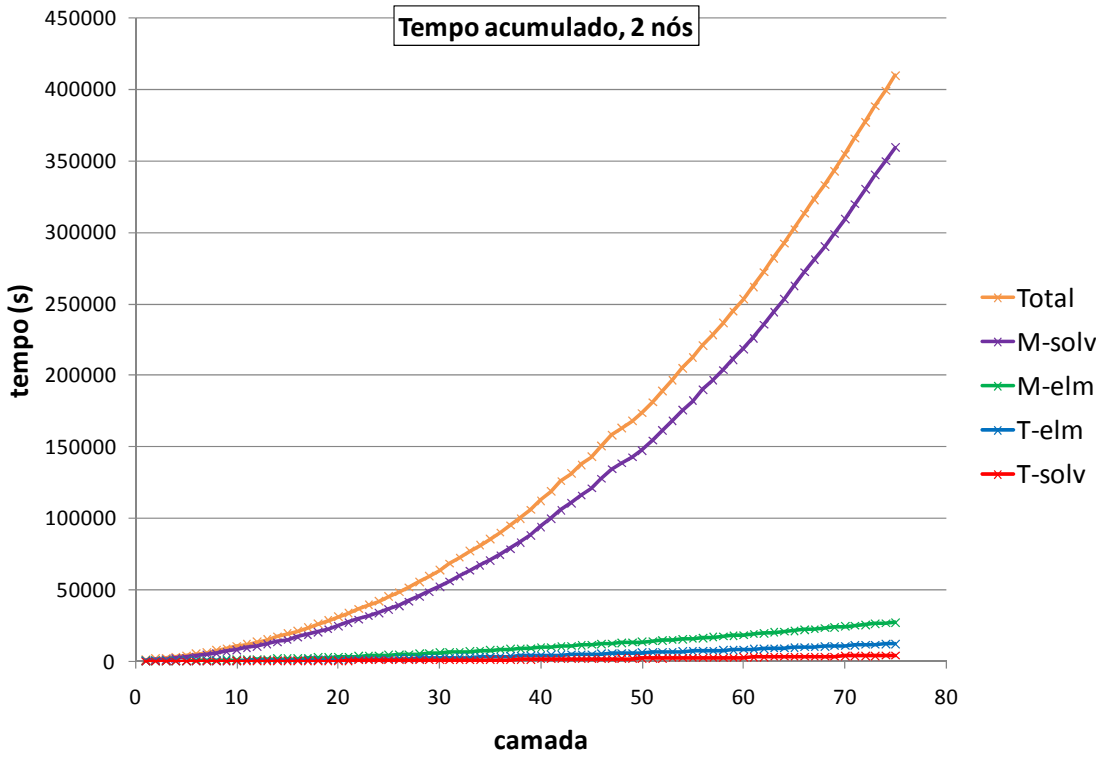


Figura 6.22: Tempo paralelo (n=2, p=4) acumulado [s].

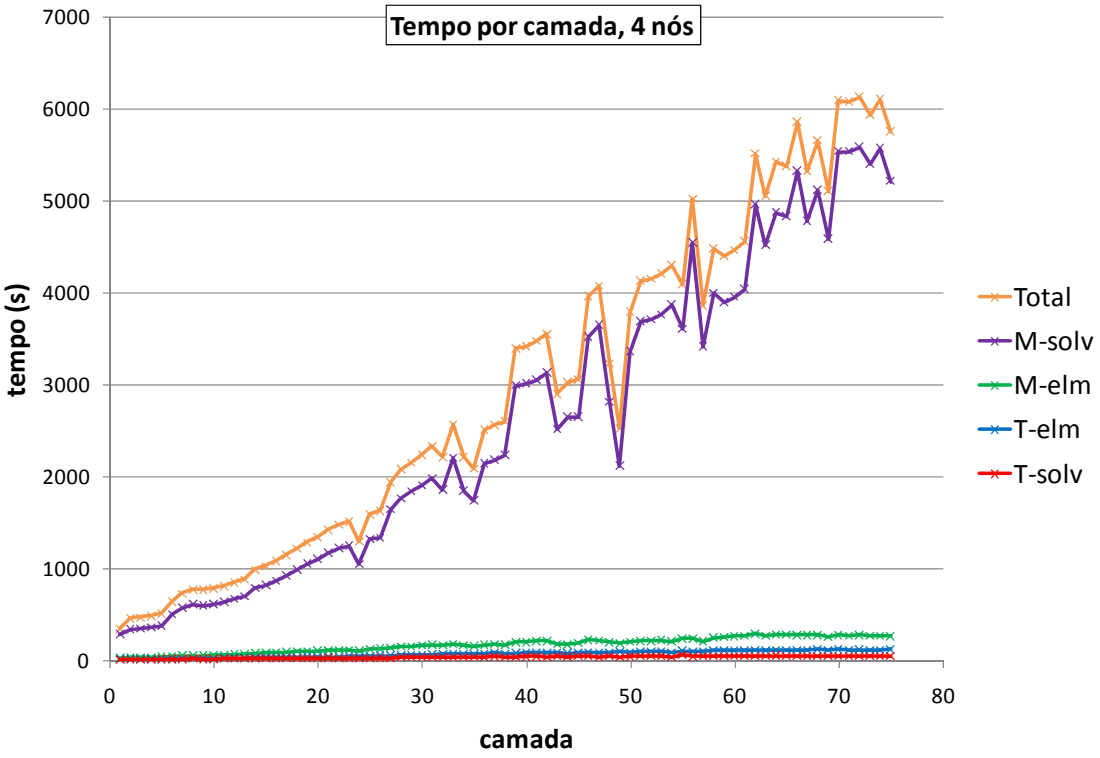


Figura 6.23: Tempo paralelo (n=4, p=8) por camada [s].

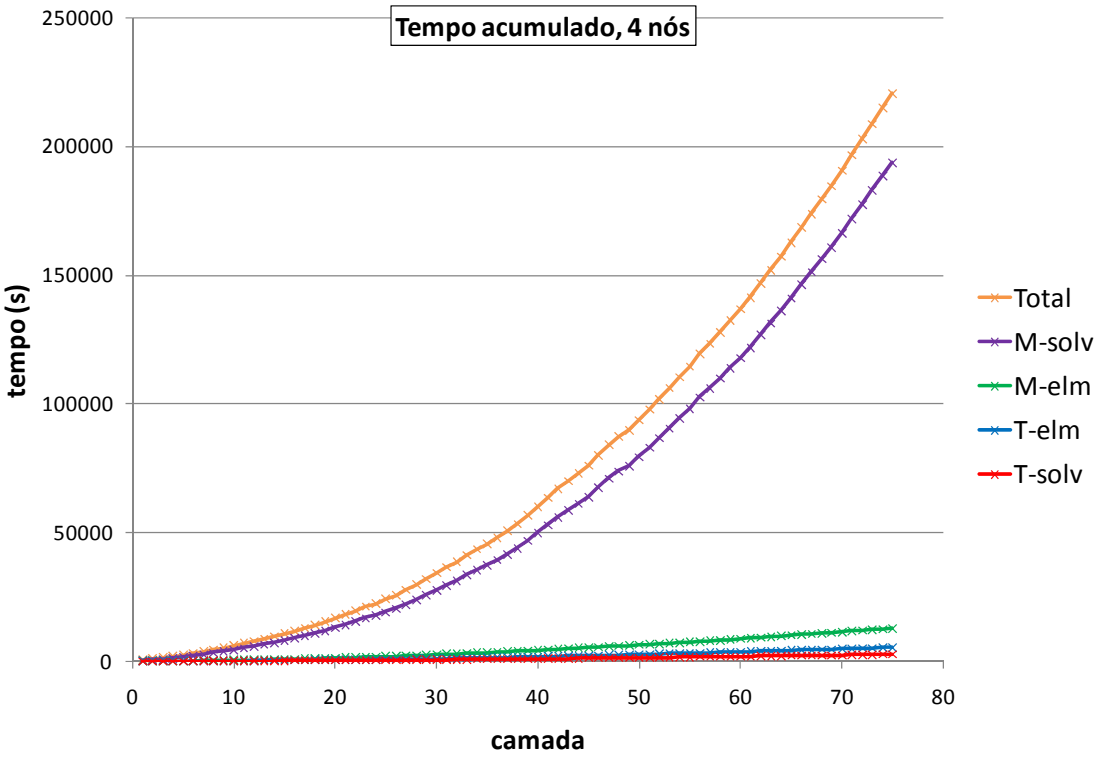


Figura 6.24: Tempo paralelo (n=4, p=8) acumulado [s].

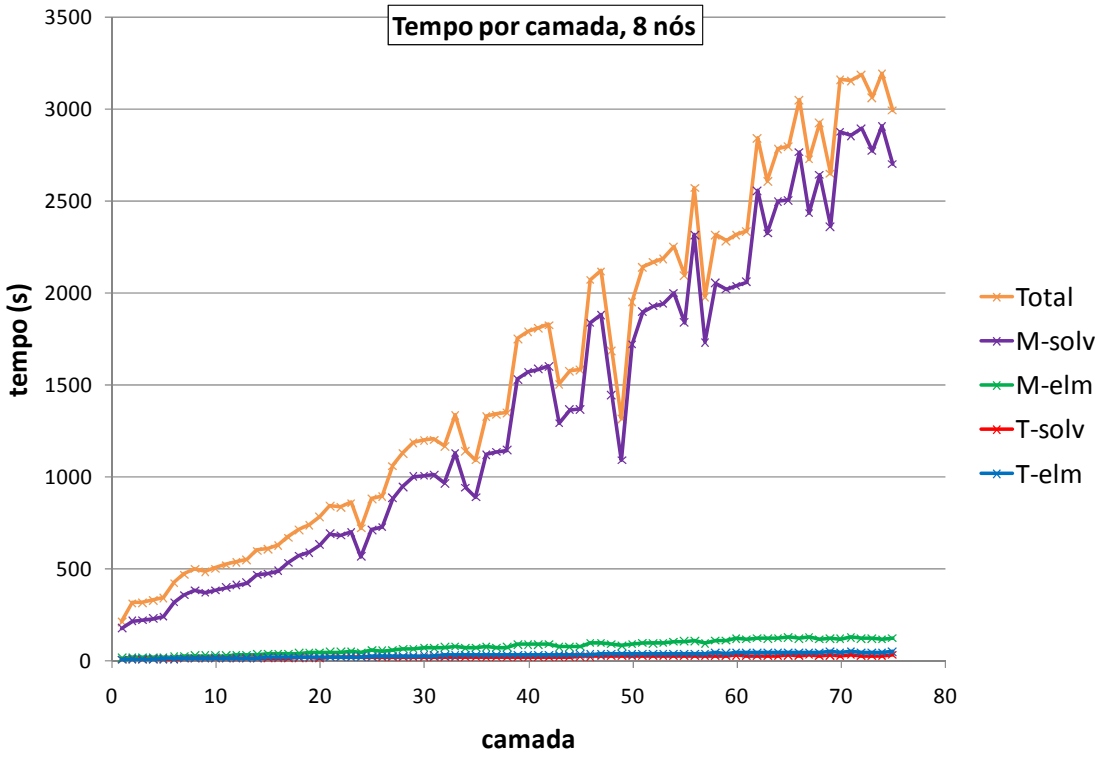


Figura 6.25: Tempo paralelo (n=8, p=16) por camada [s].

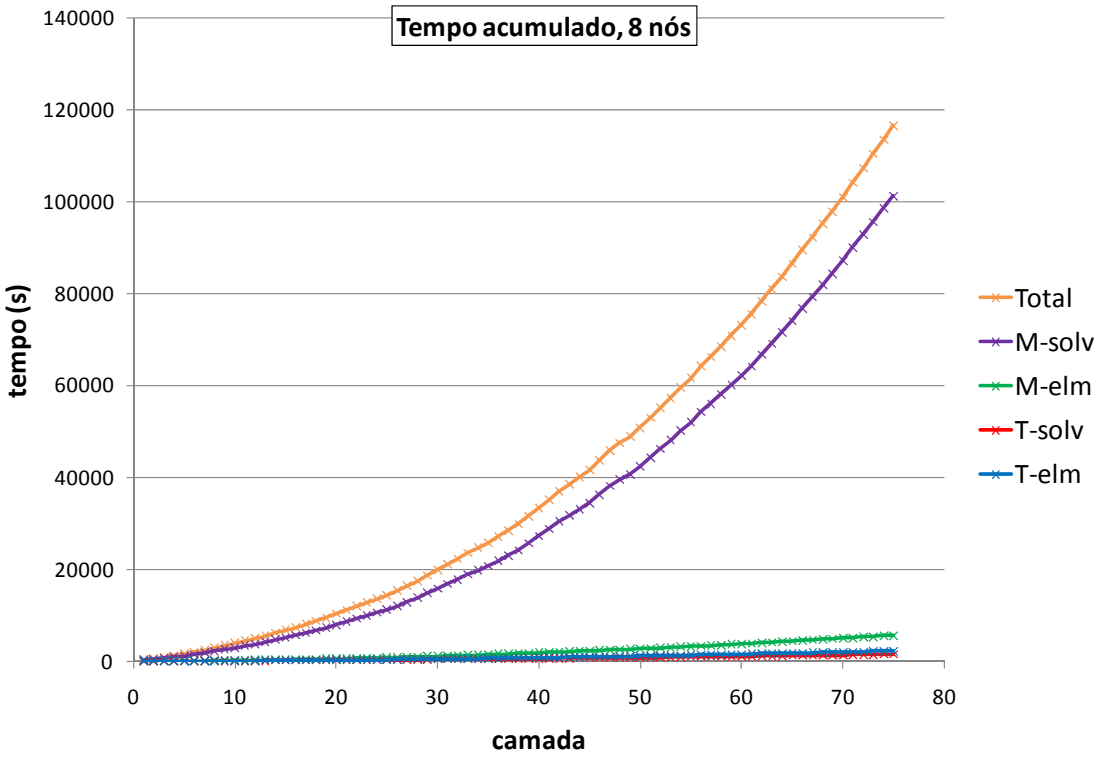


Figura 6.26: Tempo paralelo (n=8, p=16) acumulado [s].

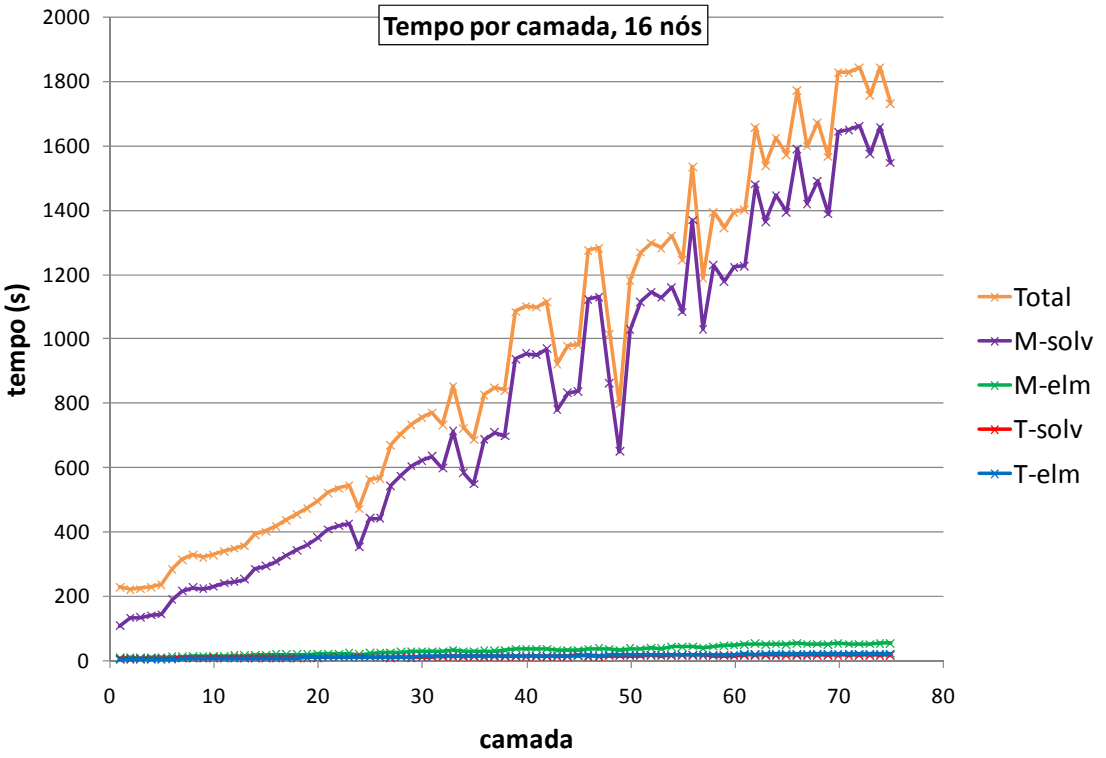


Figura 6.27: Tempo paralelo (n=16, p=32) por camada [s].

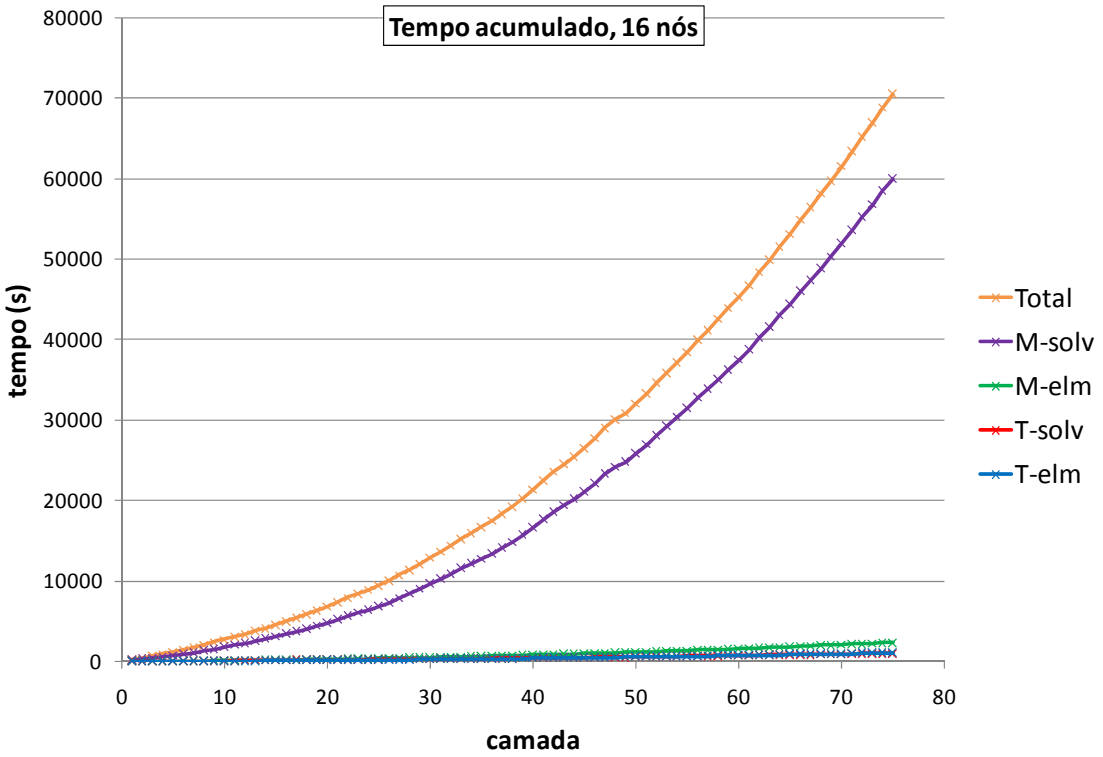


Figura 6.28: Tempo paralelo (n=16, p=32) acumulado [s].

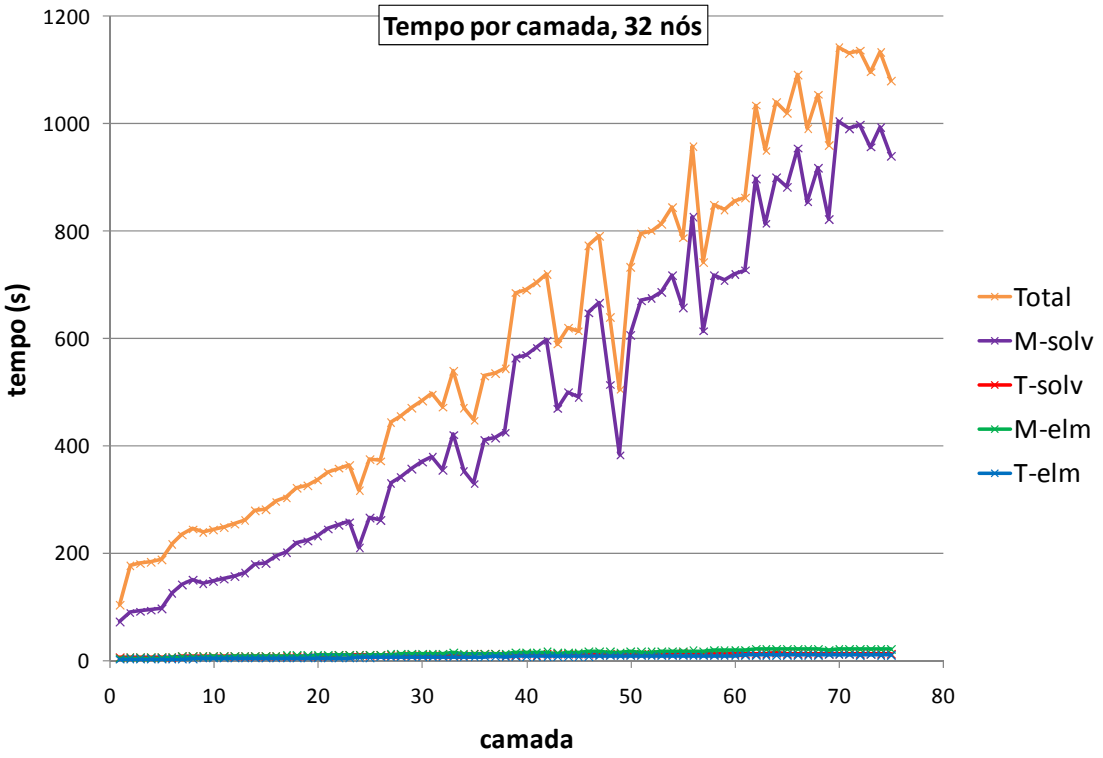


Figura 6.29: Tempo paralelo (n=32, p=64) por camada [s].

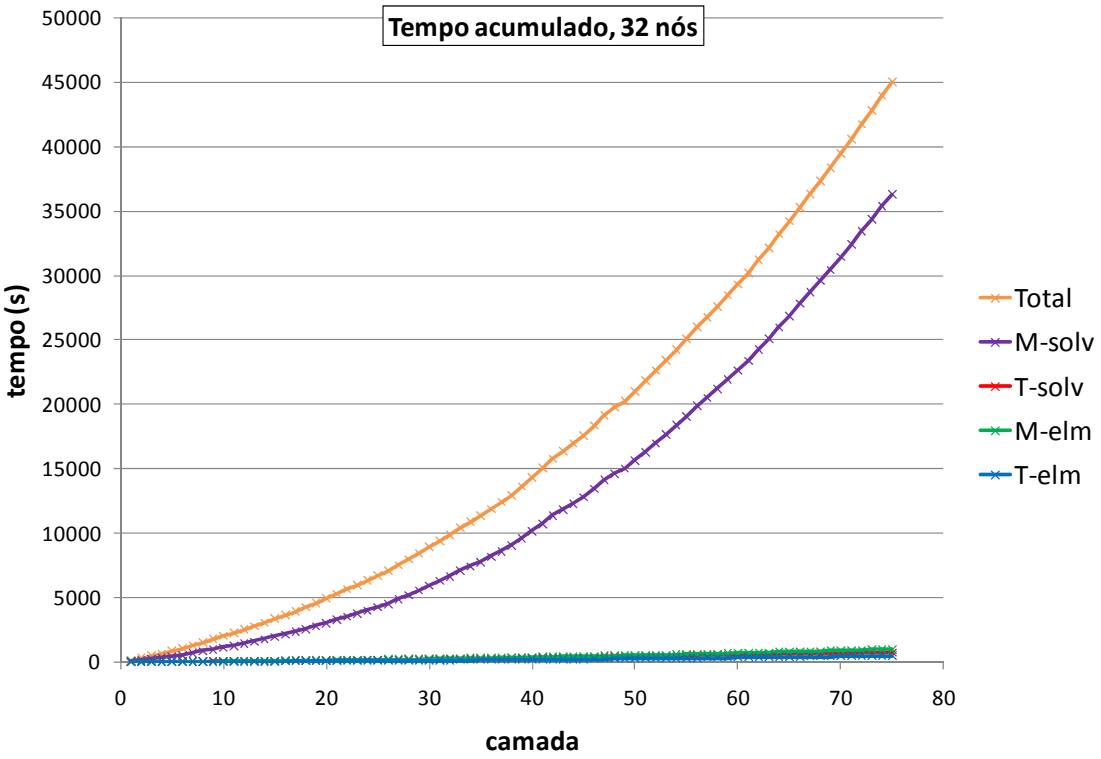


Figura 6.30: Tempo paralelo (n=32, p=64) acumulado [s].

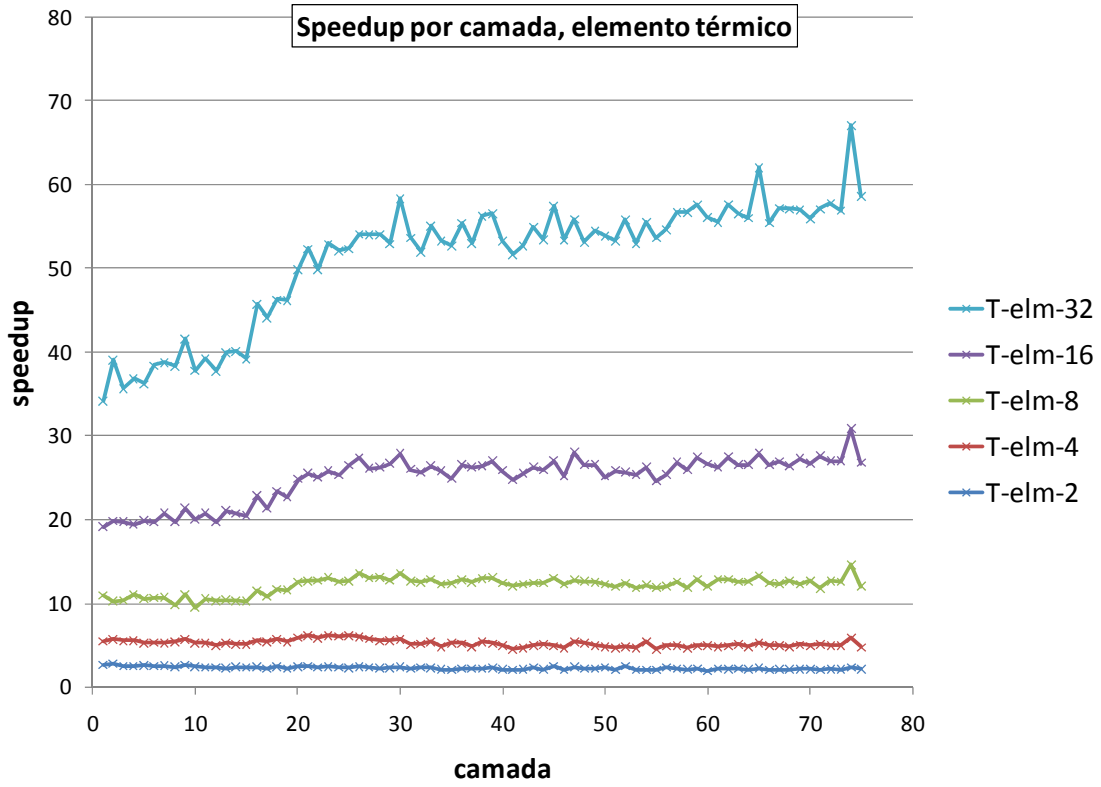


Figura 6.31: Speedups nos elementos, problema térmico, por camada.

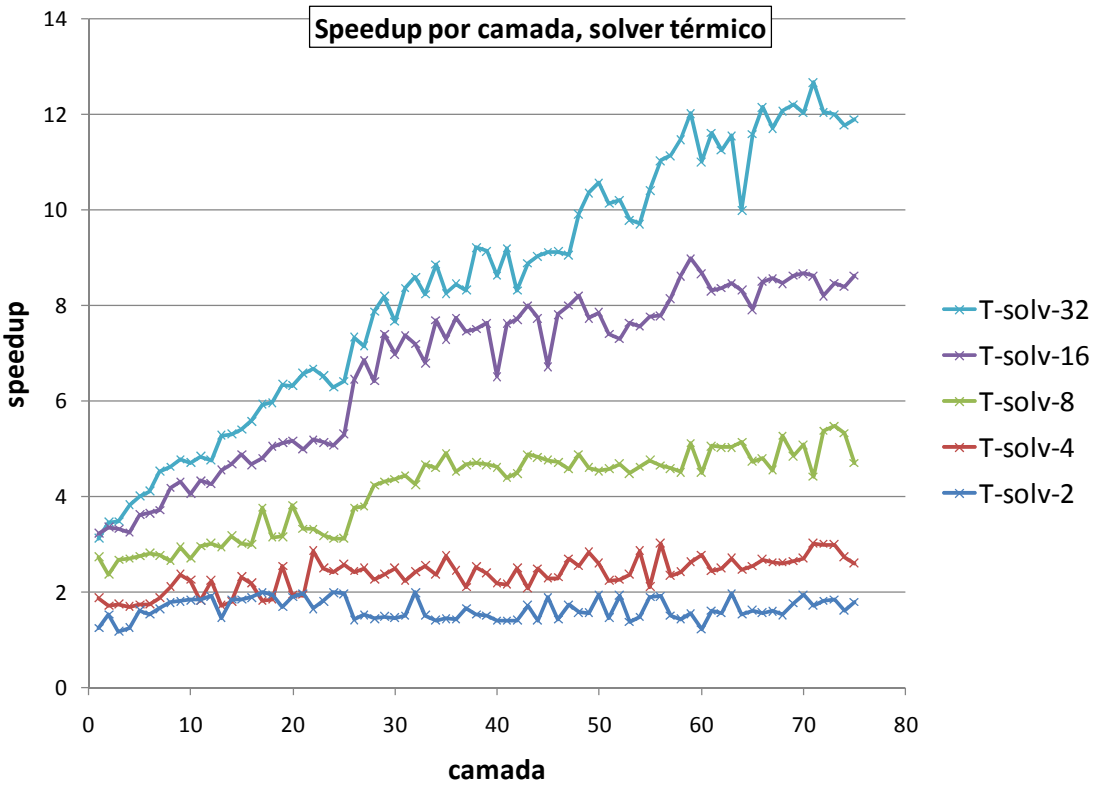


Figura 6.32: Speedups no solver PCGD, problema térmico, por camada.

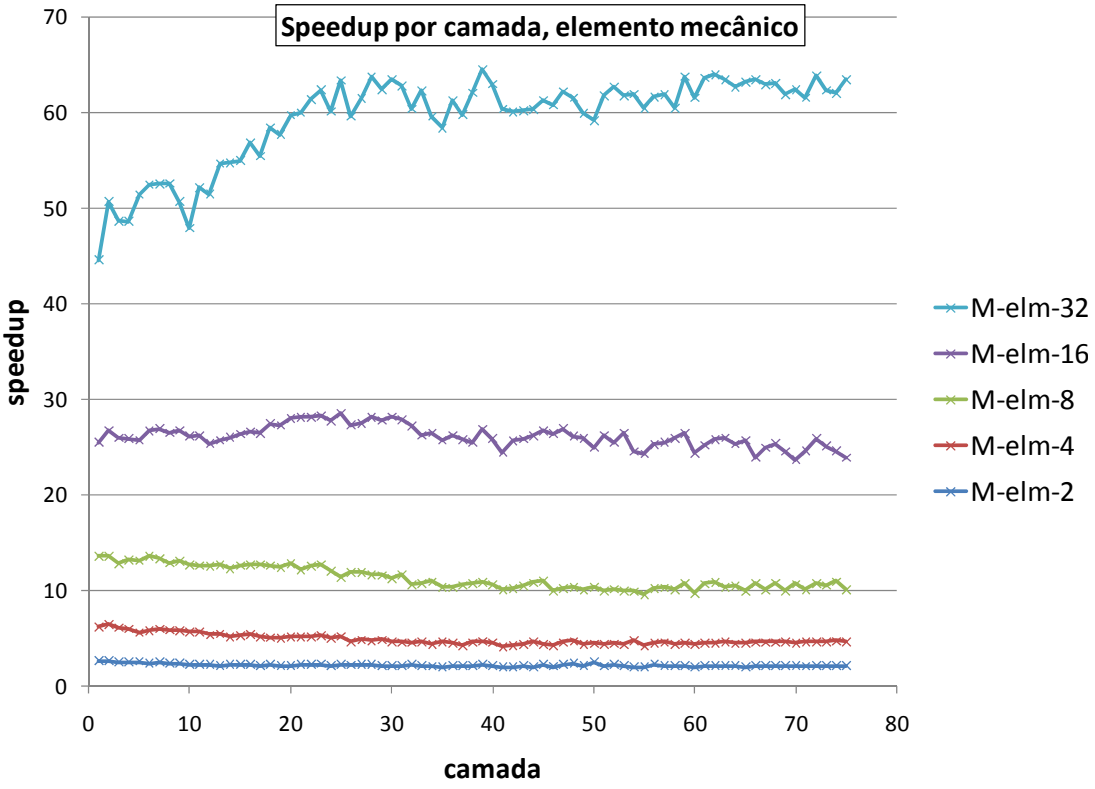


Figura 6.33: Speedups nos elementos, problema mecânico, por camada.

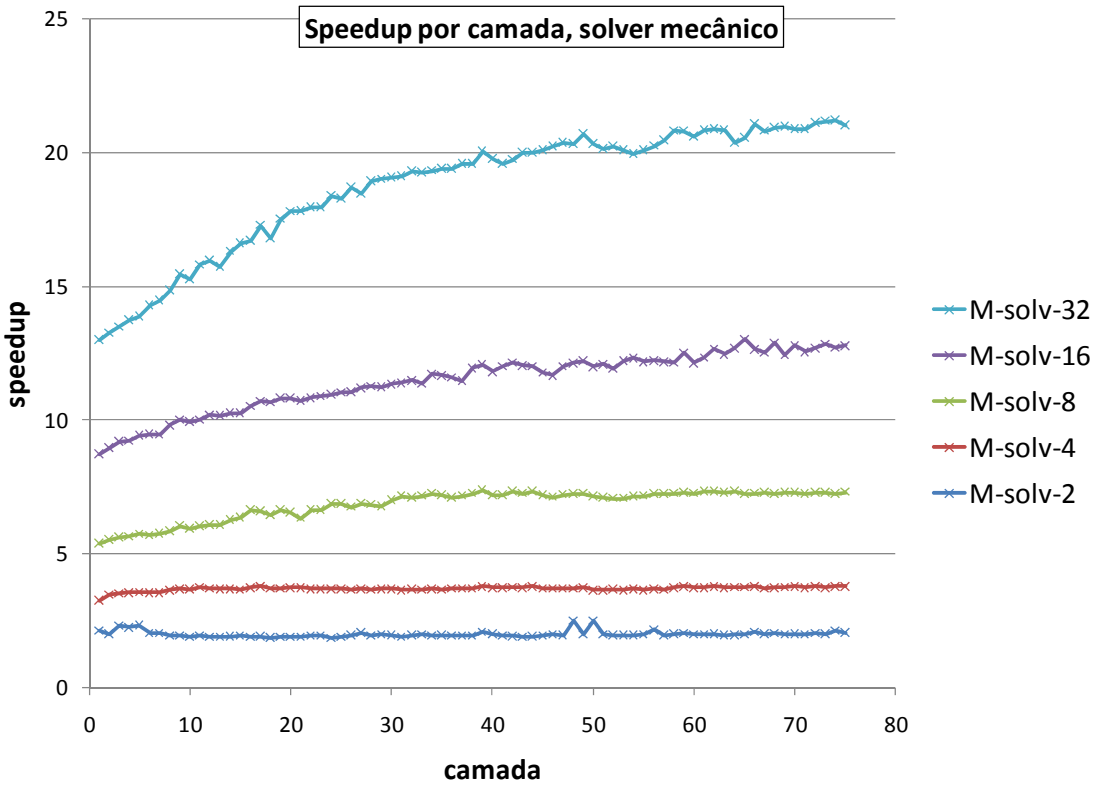


Figura 6.34: Speedups no solver PCGD, problema mecânico, por camada.

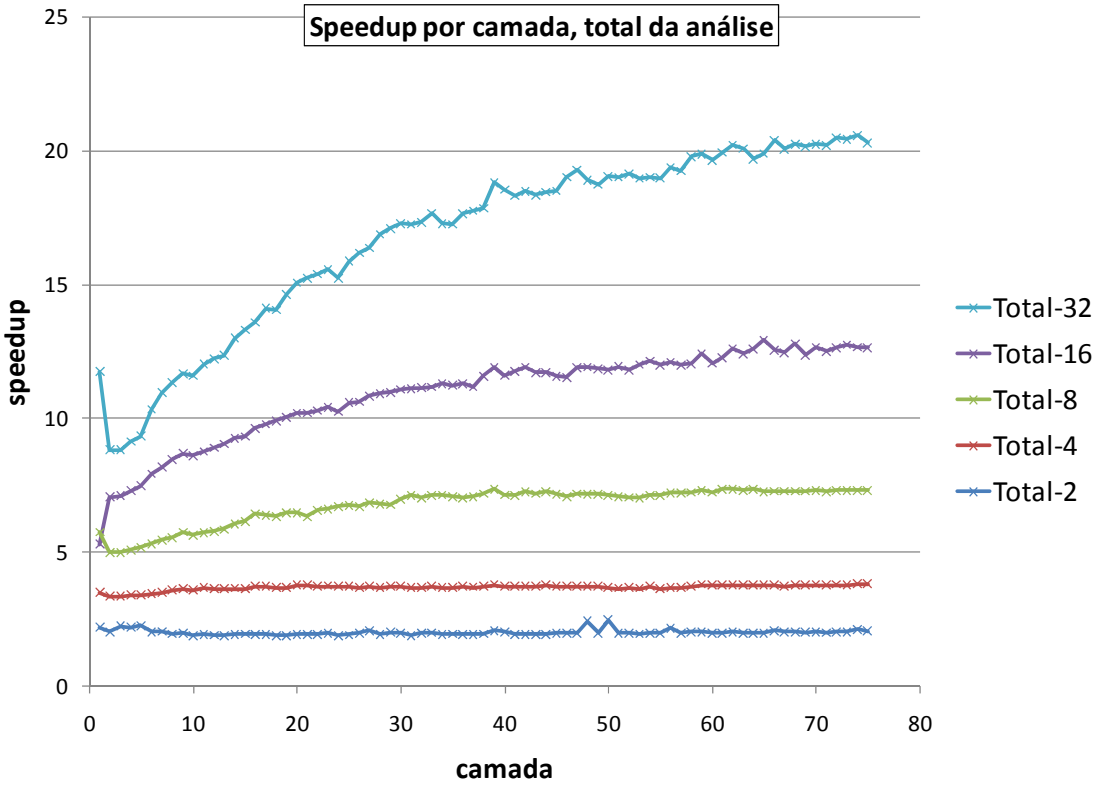


Figura 6.35: Speedups totais da análise por camada.

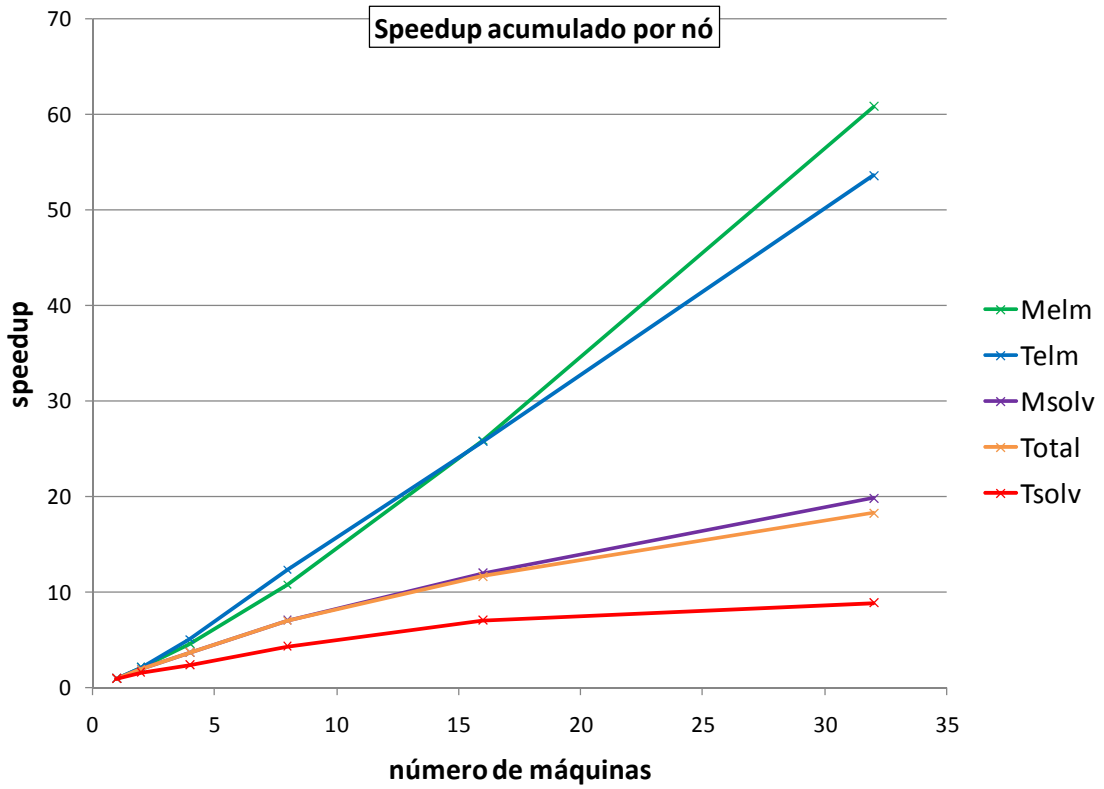


Figura 6.36: Speedups acumulados por nó.

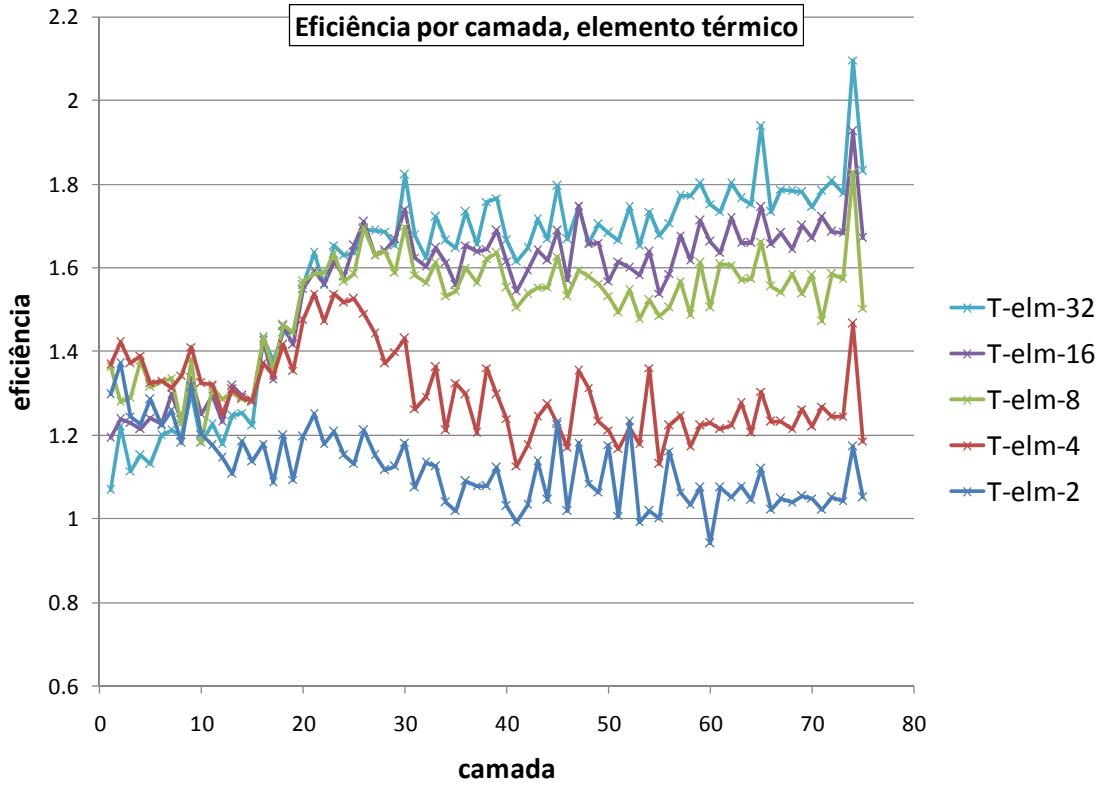


Figura 6.37: Eficiência nos elementos, problema térmico, por camada.

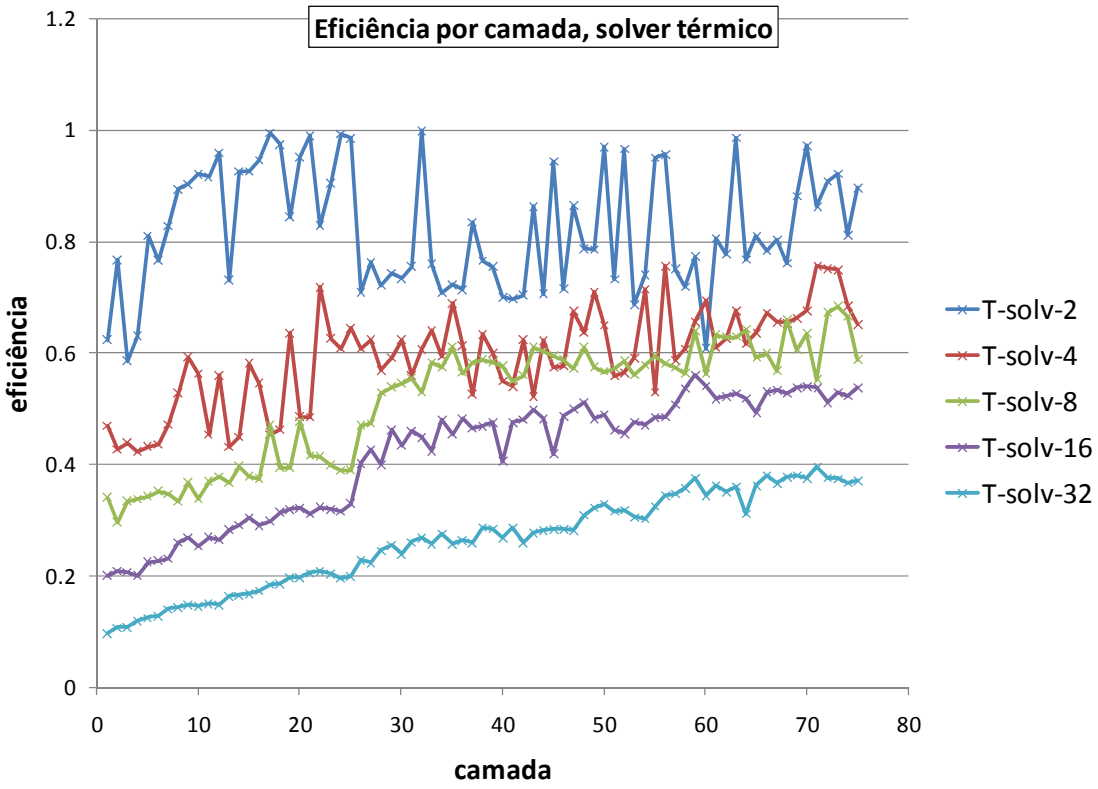


Figura 6.38: Eficiência no solver PCGD, problema térmico, por camada.

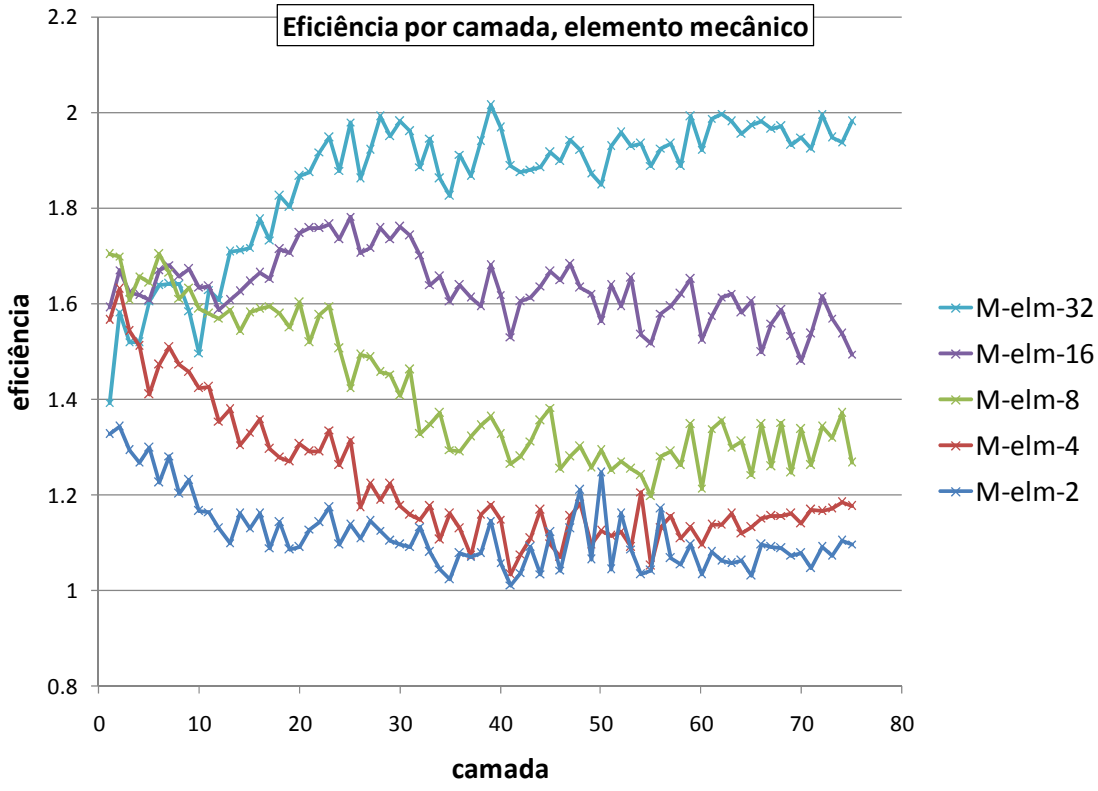


Figura 6.39: Eficiência nos elementos, problema mecânico, por camada.

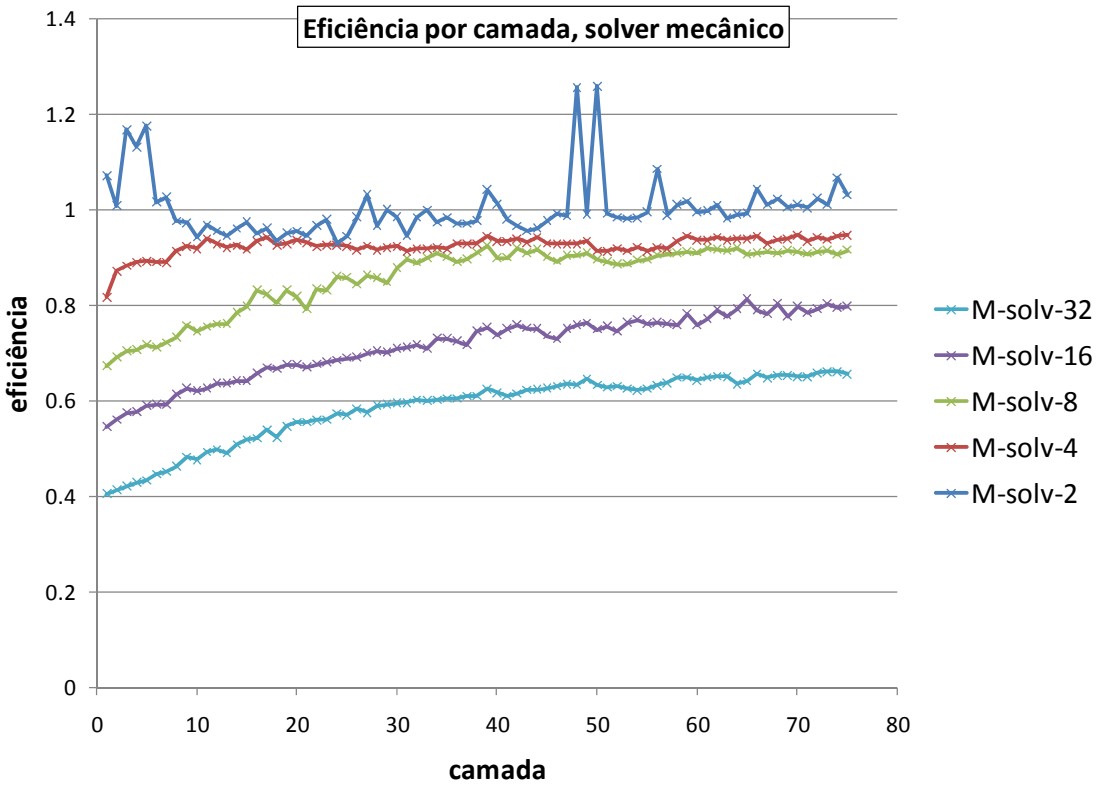


Figura 6.40: Eficiência no solver PCGD, problema mecânico, por camada.

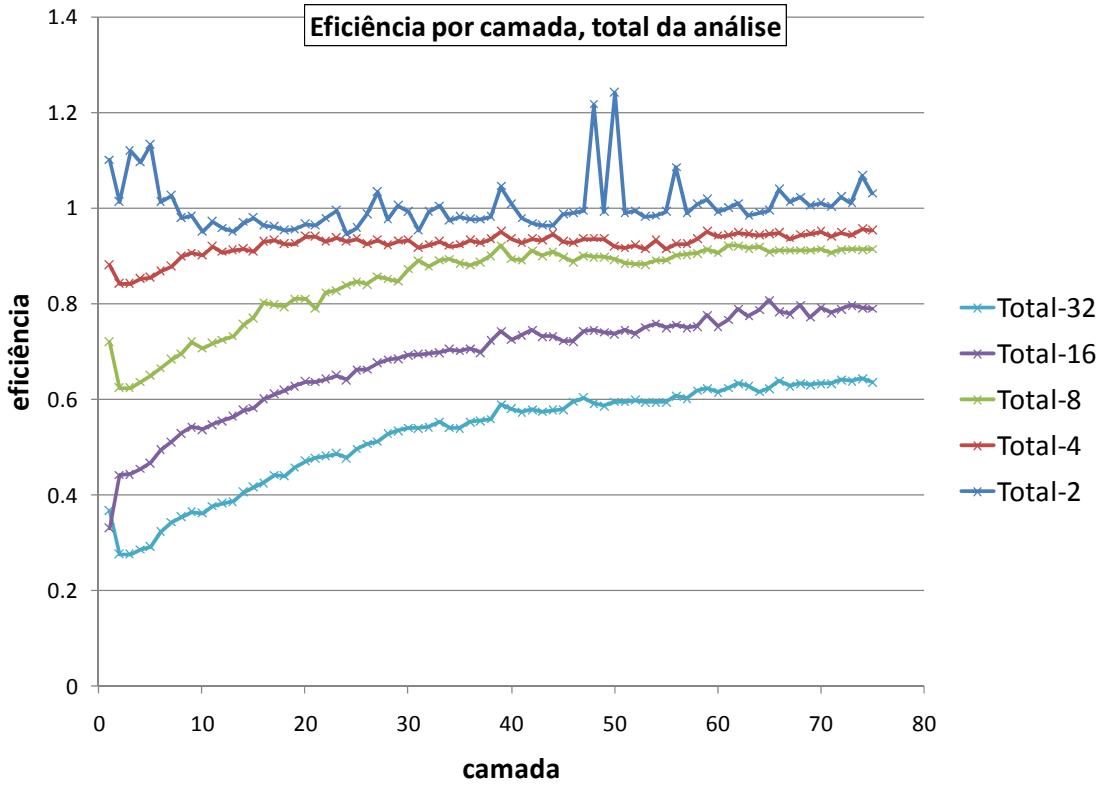


Figura 6.41: Eficiência total da análise por camada.

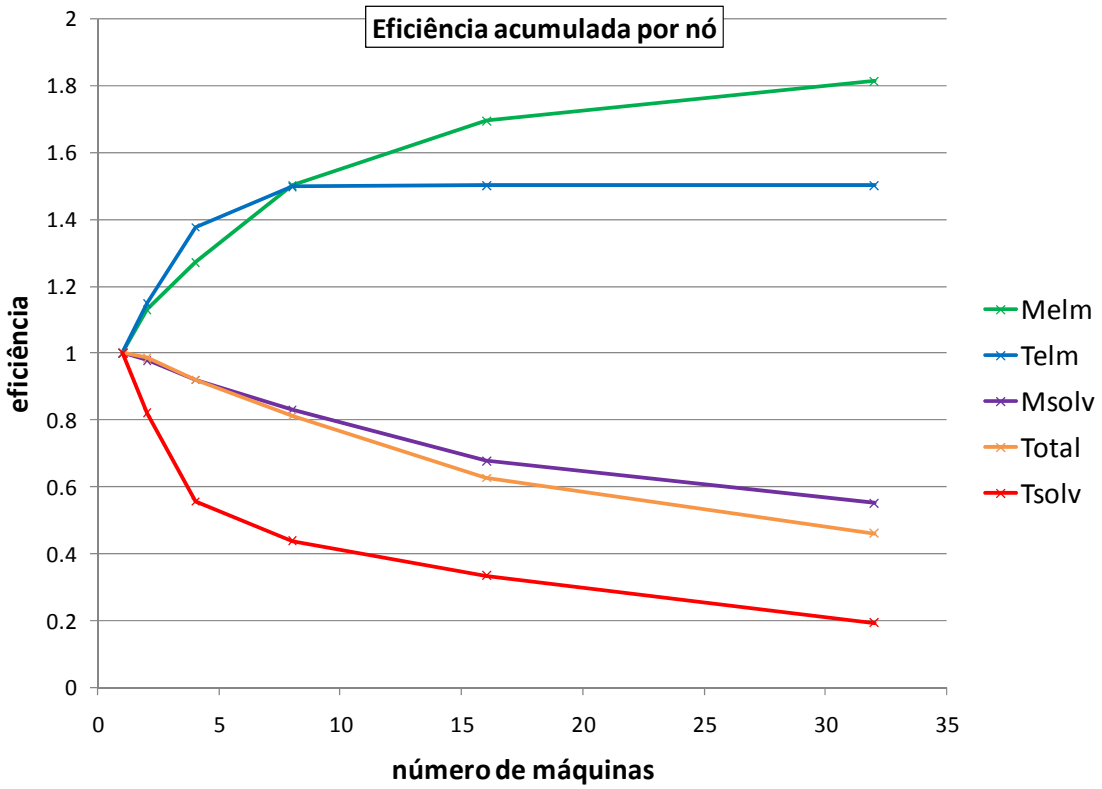


Figura 6.42: Eficiência acumulada por nó.

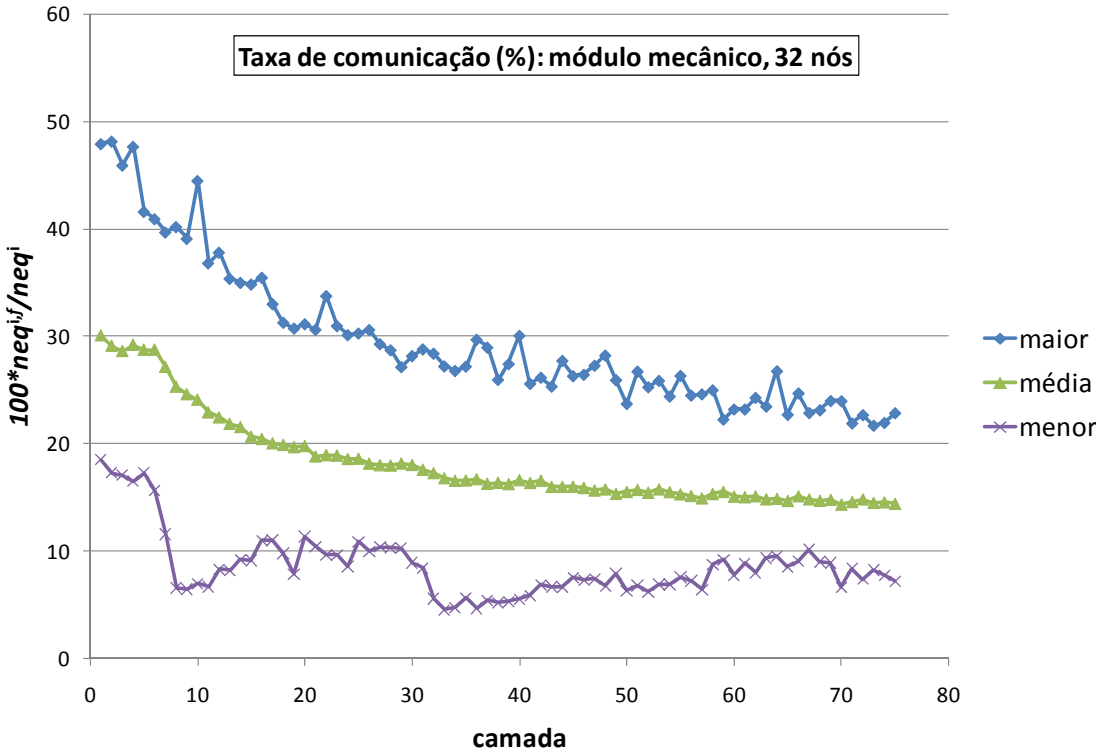


Figura 6.43: Taxa de comunicação por processo para 32 nós.

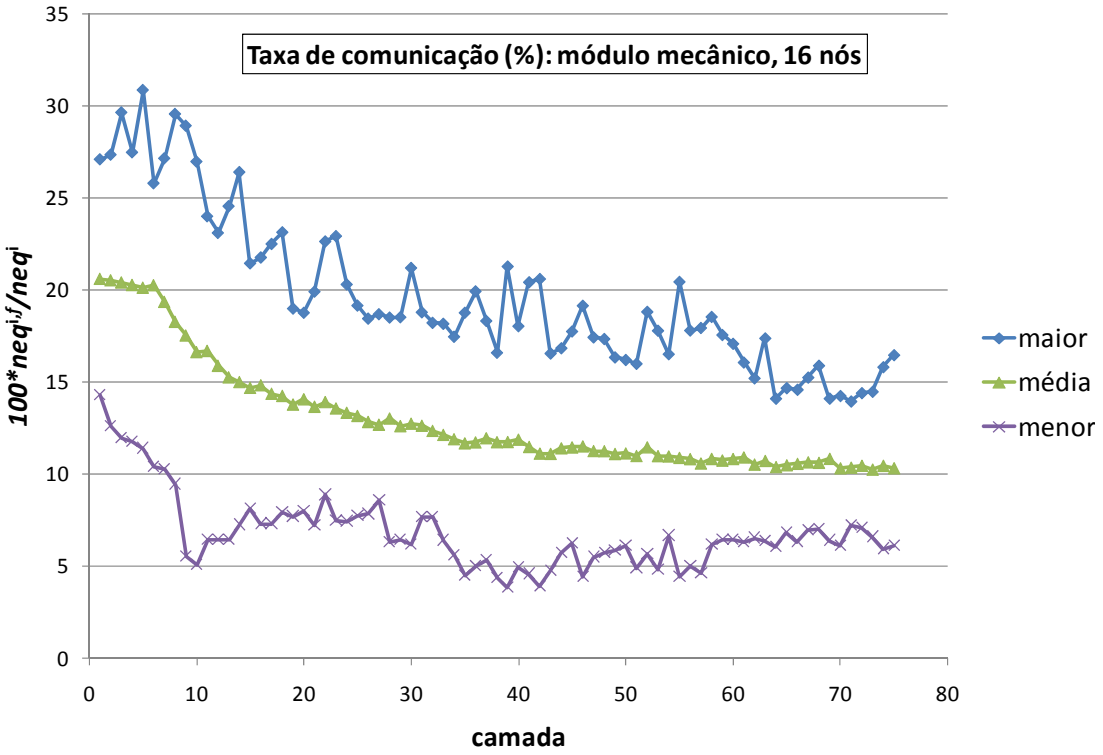


Figura 6.44: Taxa de comunicação por processo para 16 nós.

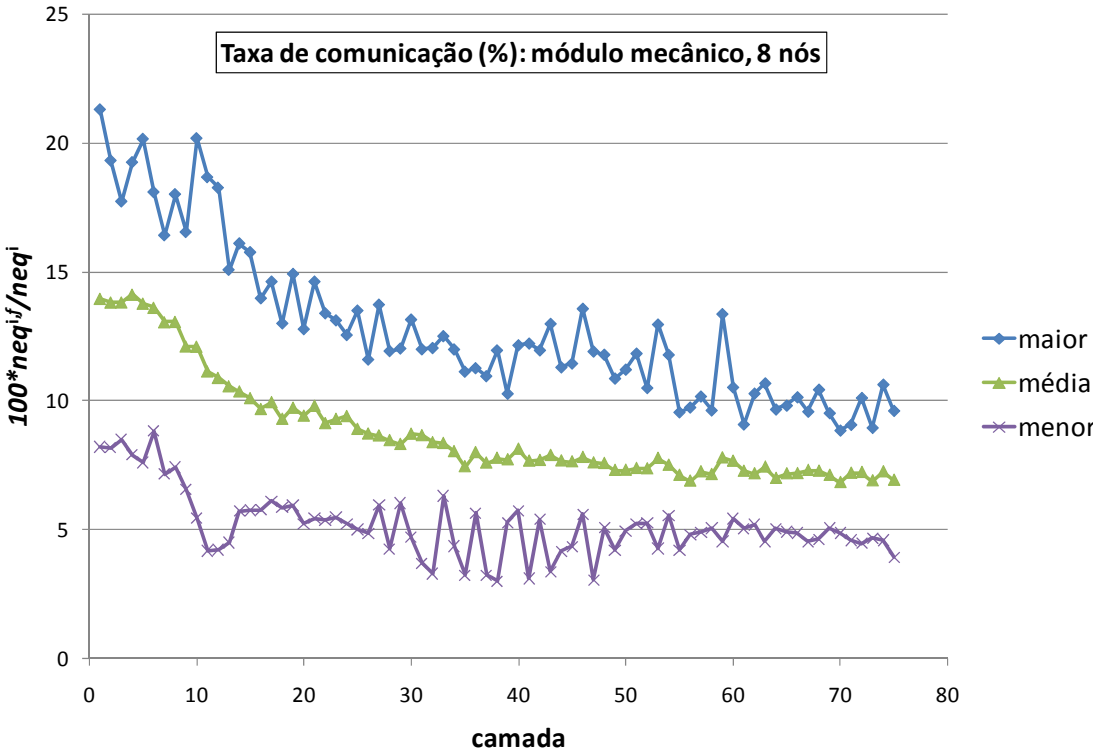


Figura 6.45: Taxa de comunicação por processo para 8 nós.

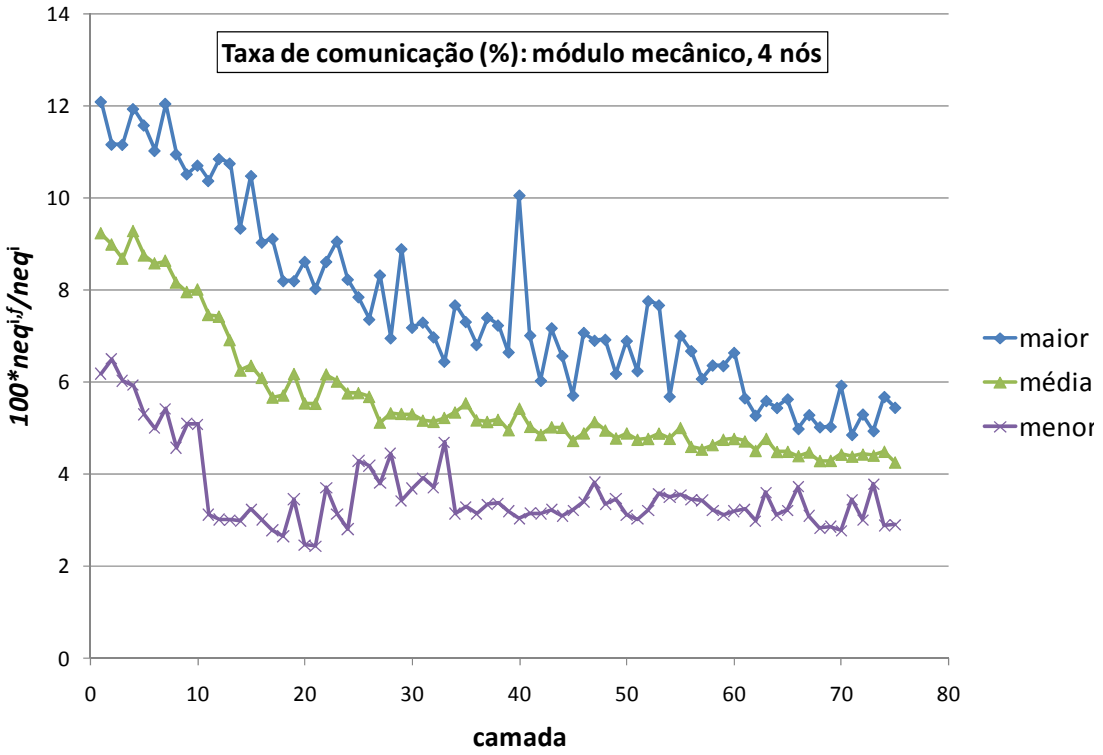


Figura 6.46: Taxa de comunicação por processo para 4 nós.

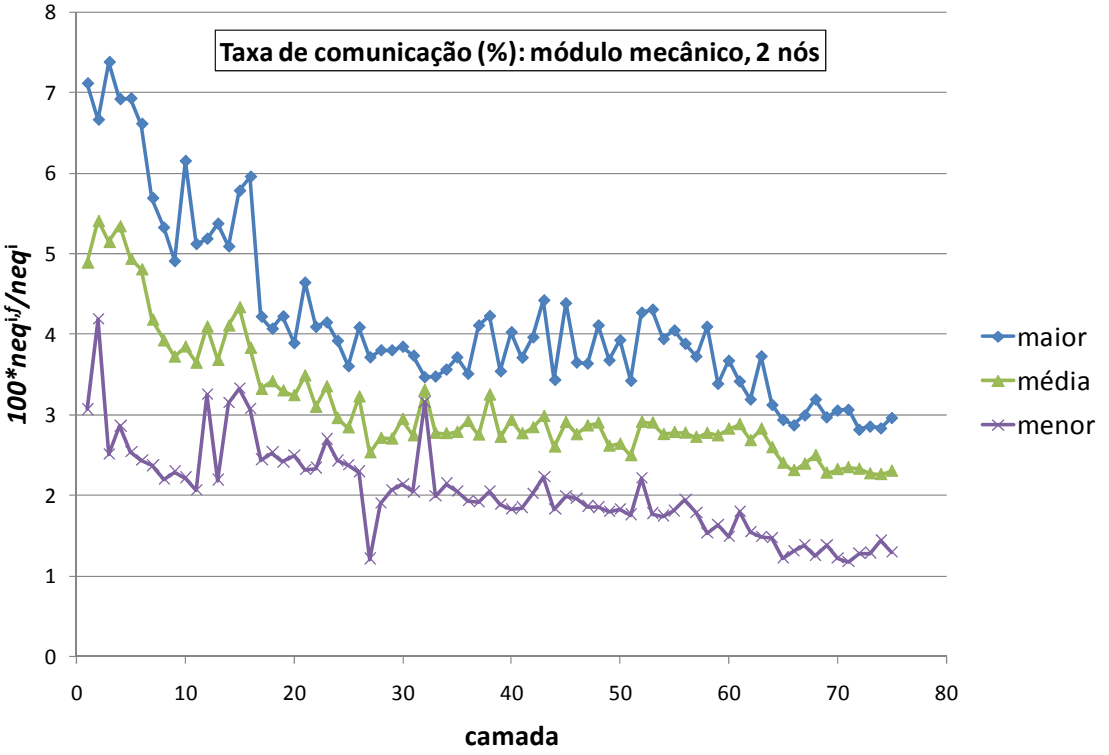


Figura 6.47: Taxa de comunicação por processo para 2 nós.

A observação das Figuras 6.17-30 permite identificar a similaridade entre os diversos gráficos de tempo por camada. Esta similaridade resulta da uniformidade da convergência na solução dos sistemas lineares, isto é, o nit não muda significativamente com a paralelização (n), fato apontado como diferencial dos métodos SBS em relação aos métodos DD. Aqui o termo significativamente indica que pode haver uma diferença de umas poucas iterações (uma ou duas) devido à alteração na ordem das OPF causada pela paralelização.

Os speedups de elemento por camada (Figuras 6.31 e 6.33) diferem muito pouco entre os módulos térmico e mecânico. Já os speedups no solver por camada (Figuras 6.32 e 6.34) apresentam forte diferença em favor do módulo mecânico (quase o dobro). Isto acontece porque o problema mecânico tem três graus de liberdade por nó, o que gera maiores valores de neq e nad para a mesma malha, e é sabido que o speedup cresce com o tamanho do problema (Ribeiro e Ferreira 2007), pois o tempo de comunicação cresce menos, proporcionalmente, que o tempo de processamento (OPFs).

O speedup por nó (Figura 6.36) diminui com o aumento do número de nós. Isto se deve ao aumento da taxa de comunicação à medida que se particiona mais e mais a malha, como pode ser visto pela comparação entre as diversas Figuras de taxa de comunicação (6.43-47). Cada uma destas Figuras (6.43-47) mostra haver uma correspondência entre a diminuição da taxa de comunicação e o aumento do speedup no solver observado nas Figuras 6.32 e 6.34, à medida que a malha cresce com o lançamento das camadas.

A eficiência da paralelização na montagem de elementos (Figuras 6.37 e 6.39) mostrou grande dispersão neste problema, tendo permanecido porém sempre

acima de 100% (speedup superlinear). O solver PCGD tem sua eficiência fortemente reduzida com o aumento da paralelização (n) (ver Figuras 6.38 e 6.40).

Os comentários acima refletem a uniformidade no comportamento do código entre os dois problemas resolvidos: vertedouro e casa de força. O desempenho pela paralelização obtido no problema da casa de força foi porém superior ao desempenho obtido pela paralelização no problema do vertedouro. Isto se dá por ser o problema da casa de força maior que o do vertedouro em número de elementos, nós e equações, o que favorece o ganho oferecido pela paralelização SBS em questão.

7 CONCLUSÃO

Apresentou-se neste trabalho uma metodologia de paralelização, em arquiteturas de memória distribuída, de um código de elementos finitos para a solução de um modelo termo-químico-mecânico, visando a simulação tridimensional da fase construtiva de uma barragem. O modelo empregado é o resultado de anos de pesquisas no Laboratório de Estruturas do Programa de Engenharia Civil da COPPE e representa o que existe de mais avançado na área. O esquema de paralelização desenvolvido, baseado em uma estratégia SBS, mostrou-se eficiente a ponto de tornar viáveis análises numéricas deste tipo de problema. A comunicação foi minimizada considerando-se um esquema de mapeamento particular a cada par de partições vizinhas e utilizando-se rotinas ponto a ponto do padrão MPI. A estrutura de dados foi extensamente estudada, tendo-se chegado a uma variante do CSR, o CSRC, que pode ser utilizada em matrizes simétricas e não-simétricas, em códigos seqüenciais e em códigos paralelos com particionamento non-overlapping, ou overlapping indistintamente. O algoritmo para produto matriz-vetor

desenvolvido para esta estrutura de dados também mostrou ser um dos pontos-chave para a eficiência da implementação, com um bom equilíbrio em termos de linearização do acesso a memória, endereçamentos indiretos e operações de ponto flutuante [(Ribeiro e Coutinho 2005) e (Ribeiro e Ferreira 2007)]. O grande desafio da paralelização de um código de elementos finitos não é o tamanho do problema a ser resolvido, mas sim a repetição de soluções de problemas não tão grandes a ponto de favorecer a relação interface/domínio. O exemplo real analisado, uma casa de força, se enquadra exatamente nesta categoria, onde dois sistemas não-lineares, um aproximadamente três vezes maior que o outro, são resolvidos em cada passo de tempo. Este exemplo apresenta uma característica mais desfavorável ainda, que é a construção em camadas, ou seja, os sistemas de equações, tanto o térmico quanto o mecânico, só utilizam a malha inteira ao final da análise. Neste ponto, a utilização de um esquema de particionamento dinâmico é obrigatória. Para este problema, obteve-se um speedup total de 18 em 32 máquinas, o que neste caso significa que um problema em que eram necessários 12 dias para se conhecer a resposta, passou a ser resolvido em 16 horas. Os resultados obtidos mostram que a utilização de clusters de PC's é de fato uma solução de excelente custo/benefício e que pode ser utilizada na prática corrente da engenharia de hoje. Os processadores utilizados são processadores duais (2 cores), e o padrão MPI permite que cada máquina de fato execute dois processos de memória compartilhada como se fossem de memória distribuída. Este procedimento resultou em speedups iguais ou superiores a 2, na fase de montagem das matrizes (loop nos elementos) e, em contrapartida, speedups muito pobres na fase de solução do sistema de equações. Deve-se isto ao fato de que durante o loop nos elementos, o número de operações de ponto flutuante prepondera sobre o

acesso contínuo à memória, enquanto que na solução do sistema a situação se inverte, com o acesso a memória sendo realizado de forma continuada e em larga escala, criando assim um gargalo no fluxo da execução do código, uma vez que os dois cores passam a disputar o acesso à mesma memória física. Esta questão parece ser o ponto crucial no que diz respeito às modernas arquiteturas de hoje, ou seja, como acessar grandes quantidades de dados em sistemas de memória compartilhada. Como trabalhos futuros sugere-se a investigação de implementações híbridas, combinando-se simultaneamente protocolos de memória distribuída e de memória compartilhada.

8 REFERÊNCIAS BIBLIOGRÁFICAS

Amestoy, P. R., I. S. Duff, e J.-Y. L'Excellent. "Multifrontal parallel distributed symmetric and unsymmetric solvers." *Comput Methods Appl Mech Eng*, 2000: 501–520.

Bazant, Z. P., e J. Planas. *Fracture and Size Effect in concrete and other quasibrittle materials.* Boca Raton: CRC Press, 1998.

Bernard, Olivier, Franz Joseph Ulm, e Eric Lemarchand. "A multiscale micromechanics-hydration model for the early-age elastic properties of cement-based materials." *Cement and concrete Research* 33 (2003): 1293-1309.

Bitzarakis, S., M. Papadrakakis, e A. Kotsopulos. "Parallel solution techniques in computational structural mechanics." *Comput Methods Appl Mech Eng*, 1997, 1 ed.: 75–104.

Bova, S. W., e G. F. Carey. "A distributed memory parallel element-by element scheme for semi-conductor device simulation." *Comput Methods Appl Mech Eng*, 2000: 403–423.

Charmpis, D.C., e M. Papadrakis. "Enhancing the performance of the FETI method with preconditioning techniques implemented on clusters of networked computers." *Computational Mechanics*, 2002: 12–18.

Coussy, Olivier. *Mechanics of Porous Continua.* Chichester: Wiley, 1995.

Dongarra, Jack, ed. *Sourcebook of parallel computing.* San Francisco, California: Morgan Kauffmann Publishers, 2005.

Dostal, Z., D. Horak, e R. Kucera. "FETI based algorithms for contact problems: scalability, large displacements and 3D Coulomb friction." *Comput Methods Appl Mech Eng*, 2005: 395–409.

Fairbairn, Eduardo de Moraes Rego, Romildo Dias Toledo Filho, Guilherme Chagas Cordeiro, Fernando Luiz Bastos Ribeiro, e Iuri Alves Ferreira. "Reducing CO2 emissions by the substitution of cement by agro-industrial residues ." *Civil engineering symposium 2008: applied and advanced concrete technology in Palestine.* Birzeit: Birzeit University, 2008. 32-38.

Ferreira, Iuri Alves. *Modelagem numérica do acoplamento termo-químico-mecânico no concreto jovem.* Rio de Janeiro, RJ: COPPE-UFRJ, 1998.

Ferreira, Iuri Alves, Eduardo de Moraes Rego Fairbairn, Fernando Luiz Bastos Ribeiro, e Romildo Dias Toledo. "Retro-ajuste de modelo de cinética multi-fase para clínquer." *XXXI Jornadas Sud-Americanas de Ingenieria Estructural.* Mendoza, 2004.

Ferreira, Iuri Alves, Eduardo de Moraes Rego Fairbairn, Fernando Luiz Bastos Ribeiro, Marcos Martinez Silvoso, e Romildo Dias Toledo. "A coupled numerical model for concrete parallelized with MPI." *XXVII CILAMCE.* Belém: ABMEC, 2006.

Ferreira, Iuri Alves, Fernando Luiz Bastos Ribeiro, Eduardo De Moraes Rego Fairbairn, e Romildo Dias Toledo. "Uma implementação em paralelo do MEF usando PCG e armazenamento CSR." *XXV CILAMCE.* Recife: ABMEC, 2004.

Grama, A., A. Gupta, G. Karypis, e V. Kumar. *Introduction to Parallel Computing.* 2. Essex: Pearson Edication - Addison Wesley, 2003.

Grimmett, G. *Percolation.* 2. Berlin: Springer-Verlag, 1999.

Hellmich, Christian. "Shotcrete as part of the new austrian tunneling method: from thermochemomechanical material modeling to structural analysis and safety assessment of tunnels." *Tese de Doutorado.* Viena: Technischen Universität Wien, Agosto de 1999.

Hellmich, Christian, e Franz Joseph Ulm. "Multisurface chemoplasticity I: Material Model for shotcrete." *Journal of Engineering Mechanics* 125 (1999): 692-701.

Hewlett, Peter C., ed. *Lea's Chemistry of Cement and Concrete.* 4. Oxford: Butterworth-Heinemann, 2004.

Hill, R. *The Mathematical Theory of Plasticity.* Oxford University Press, 1998.

Komarovsky, A. A. *Physics of Strength and Fracture Control.* 2. Boca Raton, Florida: CRC Press, 2003.

Krauss, Matias, e Karim Hariri. "Determination of initial degree of hydration for improvement of early age properties of concrete using ultrasonic wave propagation." *Cement & Concrete Composites*, 2006: 299-306.

Lemaitre, J., e J.-L. Chaboche. *Mécanique des matériaux solides.* 2. Paris: Dunod, 1998.

Lemarchand, Eric, Franz Joseph Ulm, e Luc Dormieux. "Effect of Inclusions on Friction Coefficient of Highly Filled Composite Materials." *Journal of Engineering Mechanics* 128, n. 8 (2002): 876-884.

Liu, C.H., D. Y. C. Leung, e C. M. Woo. "Development of a scalable finite element solution to the Navier–Stokes equations." *Computational mechanics*, 2003, 3 ed.: 185–198.

Nemat-Nasser, S., e M. Hori. *Micromechanics: overall properties of heterogeneous materials.* 2. Amsterdan: North-Holland, 1999.

Papadrakis, M., e S. Bitzarakis. "Domain decomposition methods for serial and parallel processing." *Adv Eng Softw*, 1996: 291–307.

Pensée, V., D. Kondo, e Luc Dormieux. "Micromechanical Analysis of Anisotropic Damage in Brittle Materials." *Journal of Engineering Mechanics* 128, n. 8 (agosto 2002): 889-897.

Rao, A.R.M. "MPI-based parallel finite element approaches for implicit nonlinear dynamic analysis employing sparse PCG solvers." *Advances in Engineering Software*, 2005: 181–198.

Rebollo, T.C., e E.C. Vera. "Study of a non-overlapping domain decompositionmethod: Poisson and Stokes problems." *Appl Numer Math*, 2004: 169–194.

Ribeiro, F. L. B., e A. L. G. A. Coutinho. "Comparison Between Element, Edge and Compressed Storage Schemes for Iterative Solutions in Finite Element Analyses."

International Journal for Numerical Methods in Engineering, 28 de maio de 2005, 4 ed.: 569-588.

Ribeiro, Luiz Fernando Bastos, e Iuri Alves Ferreira. "Parallel implementation of the finite element method using compressed data structures." *Computational Mechanics*, dezembro de 2007: 1-187.

Ribeiro, Luiz Fernando Bastos, Iuri Alves Ferreira, e Eduardo de Moraes Rego Fairbairn. "Parallel FE iterative solutions for distributed memory architectures using mpi." *XXVI CILAMCE.* Guarapari: ABMEC e AMC, 2005.

Rosa, Luiz Pinguelli, e Louise Land B. Lomardo. "The Brazilian energy crisis and a study to support building efficiency legislation." *Energy and Buildings* 36 (2004): 89-95.

Saad, Yousef. *Iterative methods for sparse linear systems.* 2a ed. Philadelphia: SIAM, 2003.

Sercombe, Jérôme, Christian Hellmich, Franz-Joseph Ulm, e Herbert Mang. "Modeling of early age creep of shotcrete. I: model and model parameters." *Journal of Engineering Mechanics*, março de 2000: 284-291.

Simo, Juan C., e T. J. R. Hughes. *Computational Inelasticity.* New york: Springer-Verlag, 1997.

Skalny, J. *Materials Science of Concrete I.* Columbus, 1997.

Soulaimani, A., Y. Saad, e A. Rebaine. "An edge based stabilized finite element method for solving compressible flows: formulation and parallel implementation." *Comput Methods Appl Mech Eng*, 2001: 6735–6761.

Stauffer, D., e A. Aharony. *Introduction to Percolation Theory.* 2. Londres: Taylor & Francis Ltd, 1994.

Taylor, H. W. *Cement Chemistry.* 2. Telford, 1997.

Ulm, Franz Joseph, e Olivier Coussy. "Strength growth as chemo-plastic hardening in early age concrete." Edição: ASCE. *Journal of Engineering Mechanics* 122, n. 12 (1996): 1123-1132.

Van Mier, J. G. M. *Fracture Processes of Concrete.* CRC Press, 1996.

Velez, K., S. Maximilien, D. Damidot, G. Fantozzi, e F. Sorrentino. “Determination by nanoindentation of elastic modulus and hardness of pure constituents of portland cement clinker.” *Cement and Concrete Research*, 2001: 555-561.

Waller, Vincent. “Relations entre composition des betons, exothemie en cours de prise et resistance en compression.” *Tese de Doutorado.* Paris, 1999.