

UNIVERSIDADE FEDERAL DE OURO PRETO – ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CIVIL
PROGRAMA DE PÓS – GRADUAÇÃO EM ENGENHARIA CIVIL

**DECOMPOSIÇÃO GENÉRICA DE DOMÍNIO PARA ANÁLISES 3D VIA
M.E.C. – ESTUDO DE EFICIÊNCIA DE ESTRATÉGIAS VOLTADAS PARA A
ANÁLISE DE PROBLEMAS DE ENGENHARIA DE GRANDE ORDEM**

AUTORA: Kátia Inácio da Silva

ORIENTADOR: Prof. Dr.–Ing. Francisco Célio de Araújo

Dissertação apresentada ao Programa de Pós–Graduação do Departamento de Engenharia Civil da Escola de Minas da Universidade Federal de Ouro Preto, como parte integrante dos requisitos para obtenção do título de Mestre em Engenharia Civil, área de concentração: Construções Metálicas.

Ouro Preto, setembro de 2005.

S586d

Silva, Kátia Inácio da.

Decomposição genérica de domínio para análises 3D via M.E.C. – estudo de eficiência de estratégias voltadas para a análise de problemas de engenharia de grande ordem [manuscrito]. / Kátia Inácio da Silva. - 2005.
xiv, 119f.: il., graf., tabs.

Orientador: Prof. Dr. Francisco Célio de Araújo.

Área de concentração: Construção Metálica.

Dissertação (Mestrado) – Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia Civil. Programa de Pós Graduação em Engenharia Civil.

1. Estruturas metálicas - Teses. 2. Análise numérica - Teses. 3. *Solvers* de Krylov – Teses. 4. Acoplamentos. 5. Métodos de elementos de contorno – Teses. 6. Integração numérica – Teses. I. Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia Civil. Programa de Pós-graduação em Engenharia Civil. II. Título.

CDU: 624.014

Catálogo: sisbin@sisbin.ufop.br

**DECOMPOSIÇÃO GENÉRICA DE DOMÍNIO PARA ANÁLISES 3D VIA
M.E.C. – ESTUDO DE EFICIÊNCIA DE ESTRATÉGIAS VOLTADAS PARA A
ANÁLISE DE PROBLEMAS DE ENGENHARIA DE GRANDE ORDEM**

AUTOR: KÁTIA INÁCIO DA SILVA

Esta dissertação foi apresentada em sessão pública e aprovada em 20 de setembro de 2005, pela Banca Examinadora composta pelos seguintes membros:



Prof. Dr. Francisco Célio de Araújo (Orientador / UFOP)



Prof. Dr. Francisco de Assis das Neves (UFOP)



Prof. DR. José Cláudio de Faria Telles (UFRJ)

“Cada um que passa em nossa vida, passa sozinho, pois cada pessoa é única e nenhuma substitui a outra. Cada um que passa sozinho, mas não vai só, leva um pouco de nós mesmos, deixa um pouco de si mesmo. Há os que levaram muito, mas não há os que deixaram nada. Isto é a maior responsabilidade de nossa vida, é a prova evidente de que duas almas não se encontram por acaso.”

Antoine de Saint-Exupéry

Aos meus queridos pais.

AGRADECIMENTOS

*“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota.”*

Madre Teresa de Calcutá

A Deus por, simplesmente, tudo.

Aos meus pais, Sebastião e Judite, pelo amor, compreensão, incentivo e orientação ao longo de toda minha vida.

Aos meus irmãos, Néia, Lene, Fábio e Paulinho, pela torcida, amizade, paciência e carinho.

Ao meu orientador, Prof. Francisco Célio de Araújo, pela oportunidade, amizade, ensinamentos e dedicação constante durante a realização deste trabalho e nesses quatro anos e meio de convívio. Muito obrigada.

Aos professores do Programa de Pós-graduação em Engenharia Civil pelos ensinamentos e incentivo.

Aos funcionários da Escola de Minas, em especial ao Prof. Walter Dornelas, à secretária Róvia e ao bolsista Renato, por estarem sempre disponíveis.

Aos meus companheiros de turma, Anderson, Ana Amélia, Ana Carolina, Cereno, Cristiano, Domenica, Fernando, Flávia, Flávio, Paulo e Kneipp.

Às minhas amigas de sempre, Beth, July e Liliane. Obrigada pelas conversas, incentivo e pela amizade verdadeira.

Aos colegas e amigos do mestrado, Aldo, Dendê, Eleonardo, Fabíola, Fernanda, Flávio Ferreira, Gegê, Jeanne, José Maria, Juninho, Luciana, Lucas, Luiz Heleno, Marcelo, Mário, Paulo Anderson, Rodrigo, Silvana, Tatianna e Wagner. Obrigada pelos bons momentos do mestrado e pelas boas risadas.

Ao CNPq pelo apoio financeiro.

A todos aqueles que contribuíram de alguma forma para a realização deste trabalho.

RESUMO

A eficiência do Método dos Elementos de Contorno na análise de problemas de engenharia depende fundamentalmente da eficiência dos algoritmos de integração e dos *solvers* utilizados. Nas últimas décadas, várias estratégias eficientes para a avaliação de integrais singulares e *quasi*-singulares foram desenvolvidas, e várias pesquisas têm evidenciado vantagens de *solvers* iterativos de Krylov em relação a *solvers* diretos. Com o objetivo de melhorar a eficiência das rotinas de montagem da matriz do sistema, propõe-se neste trabalho um processo baseado na combinação da transformação de coordenadas polares triangulares e da transformação não-linear de Telles. Além disso, o *solver* de gradiente biconjugado pré-condicionado de Jacobi (J-BiCG) é utilizado para o desenvolvimento de um algoritmo genérico de subestruturação. Vale ressaltar que elementos de contorno descontínuos são aplicados de modo a simplificar a introdução das condições de acoplamento entre um número genérico de subregiões, as quais eventualmente apresentam arestas e cantos. Desse modo, tanto o tempo de resolução do sistema quanto a memória são otimizados, uma vez que eliminam-se, completamente, o armazenamento e a manipulação de blocos de coeficientes nulos.

Por fim, nas aplicações, uma série de exemplos numéricos é apresentada nos quais a performance das estratégias propostas é observada em termos de precisão das respostas e tempo de CPU para a montagem e resolução do sistema de equações algébricas resultante.

ABSTRACT

The efficiency of Boundary Element Methods in the analysis of engineering problems strongly depends on the efficiency of integration algorithms and *solvers* used. In the last decades several efficient strategies for the evaluation of singular and *quasi*-singular integrals have been developed, and several strategies have highlighted the advantages of iterative Krylov's *solvers* compared to the direct ones. Aiming at improving the efficiency of the matrix-assembly routines, an integration procedure based on the combination of the triangle-polar-coordinate transformation and the Telles' non-linear coordinate transformation is proposed in this work. Besides, the Jacobi-preconditioned biconjugate gradient (J-BiCG) is adopted for developing the generic substructuring algorithm. It is worth mentioning that discontinuous boundary elements are employed in order to simplify the introduction of coupling conditions between a generic number of subregions, possibly with edges and corners. Thus, not only the solving time but also the storage memory are optimized, since the manipulation of zero blocks is completely avoided.

A series of numerical examples is presented in which the performance of the proposed strategies is observed in terms of precision and CPU-time measurements for assembling and solving the resulting BE systems of algebraic equations.

SUMÁRIO

Resumo	VI
Abstract	VII
Lista de Figuras	X
Lista de Tabelas	XIV
Capítulo 1 – Introdução	1
Capítulo 2 – Formulação Direta do M.E.C.	5
2.1. Introdução	5
2.2. Formulações Integrais do M.E.C.....	6
2.3. Algebrização da Equação Integral de Contorno.....	11
2.4. Geração de Elementos de Contorno Descontínuos	22
Capítulo 3 – Estratégia de Decomposição Genérica de Domínio.....	27
3.1. Introdução	27
3.2. <i>Solvers</i> Iterativos	28
3.3. Acoplamento EC–EC Genérico.....	29
3.4. Interfaces entre Subregiões.....	33
3.5. Pesquisa de Acoplamento	35
3.6. Resolução do Sistema Acoplado	37
Capítulo 4 – Algoritmos de Integração.....	39
4.1. Introdução	39
4.2. Integrais Não–singulares e <i>quasi</i> –singulares.....	40
4.3. Integrais Singulares	42
4.4. Procedimentos Especiais de Integração.....	43
4.4.1. Transformação de Coordenadas Polares Triangulares	44
4.4.2. Transformação Cúbica.....	47
4.4.3. Processo Combinado – Transformação de Coordenadas Polares Triangulares Juntamente com a Transformação Cúbica.....	50

4.5. Estudos de Casos.....	53
4.6. Elementos Esbeltos	62
Capítulo 5 – Aplicações	63
5.1. Introdução	63
5.2. Exemplo 1: Problema de Transferência de Calor.....	64
5.3. Exemplo 2: Barragem com Cortinas de Estaca–Prancha.....	76
5.4. Exemplo 3: Casca Cilíndrica com Enrijecedores Internos e Fratura.....	81
5.5. Exemplo 4: Interação Solo–Estrutura.....	97
Capítulo 6 – Conclusões.....	110
6.1. Conclusões.....	110
6.2. Aspectos Futuros.....	114
Referências Bibliográficas	115

LISTA DE FIGURAS

CAPÍTULO 2

Figura 2.1 – Representação do problema de valor de contorno.....	7
Figura 2.2 – Elementos de contorno contínuos.....	12
Figura 2.3 – Elementos de contorno descontínuos.....	14
Figura 2.4 – Função de forma associada ao nó 1 do elemento.....	20
Figura 2.5 – Problema genérico.....	23
Figura 2.6 – Malha de elementos de contorno atualizada para $\tilde{\mathbf{U}}_1$	26

CAPÍTULO 3

Figura 3.1 – Algoritmo do <i>solver</i> de Gradiente Bi-Conjugado com matriz de precondicionamento \mathbf{Q}	29
Figura 3.2 – Condições de acoplamento e continuidade.....	31
Figura 3.3 – Subdomínios acoplados.....	32
Figura 3.4 – Matriz global do sistema acoplado.....	32
Figura 3.5 – Malha de elementos de contorno $\tilde{\mathbf{U}}_1$	34
Figura 3.6 – Código de acoplamento: elasticidade.....	34
Figura 3.7 – Nós acoplados.....	37
Figura 3.8 – Processo iterativo.....	38

CAPÍTULO 4

Figura 4.1 – Subelementos de integração.....	40
Figura 4.2 – Seqüência de mapeamentos – transformação de coordenadas polares triangulares.....	45
Figura 4.3 – Posição dos pontos de integração após a transformação de coordenadas polares triangulares.....	47
Figura 4.4 – Posição dos pontos de integração após a transformação não-linear.....	50
Figura 4.5 – Posição dos pontos de integração após a transformação combinada.....	52
Figura 4.6 – Erro referente à integração da função I.....	54
Figura 4.7 – Erro referente à integração da função II.....	54

Figura 4.8 – Erro referente à integração da função III.....	55
Figura 4.9 – Erro referente à integração da função IV	55
Figura 4.10 – Domínio de integração.....	56
Figura 4.11 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.02$	57
Figura 4.12 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.04$	57
Figura 4.13 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.1$	58
Figura 4.14 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.2$	58
Figura 4.15 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.02$	59
Figura 4.16 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.04$	60
Figura 4.17 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.1$	60
Figura 4.18 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.2$	61

CAPÍTULO 5

Figura 5.1 – Decomposição de domínio	64
Figura 5.2 – Malhas de elementos de contorno para cada subregião	64
Figura 5.3 – Temperatura calculada com diferentes métodos – modelo I.....	66
Figura 5.4 – Temperatura obtida para os diferentes valores de d – modelo I	66
Figura 5.5 – Temperatura calculada com diferentes métodos – modelo II.....	67
Figura 5.6 – Temperatura obtida para os diferentes valores de d – modelo II	67
Figura 5.7 – Erro na avaliação do potencial x npi ($d = 0.1$)	68
Figura 5.8 – Erro na avaliação do potencial x npi ($d = 0.05$).....	69
Figura 5.9 – Erro na avaliação do potencial x npi ($d = 0.01$).....	69
Figura 5.10 – Erro na avaliação do fluxo x npi ($d = 0.1$)	70
Figura 5.11 – Erro na avaliação do fluxo x npi ($d = 0.05$).....	70
Figura 5.12 – Erro na avaliação do fluxo x npi ($d = 0.01$).....	71
Figura 5.13 – Tempo de CPU dos <i>solvers</i> para o modelo I.....	72
Figura 5.14 – Tempo de CPU dos <i>solvers</i> para o modelo II.....	72
Figura 5.15 – Número de iterações/ordem do sistema: modelo I	73
Figura 5.16 – Número de iterações/ordem do sistema: modelo II	73
Figura 5.17 – Tempo de CPU DLSARG/Tempo de montagem: modelo I.....	74
Figura 5.18 – Tempo de CPU DLSARG/Tempo de montagem: modelo II.....	74
Figura 5.19 – Número de condicionamento estimado para o modelo I	75
Figura 5.20 – Número de condicionamento estimado para o modelo II	75

Figura 5.21 – Barragem com cortinas de estaca-prancha.....	76
Figura 5.22 – Subestruturação da barragem	77
Figura 5.23 – Linhas equipotenciais e distribuição de fluxo	78
Figura 5.24 – Pressão sob a base da barragem	78
Figura 5.25 – Fluxo no nó $P = (171.8, 0.0, 64.0)$	79
Figura 5.26 – Fluxo no nó $P = (175.0, 0.0, 64.0)$	79
Figura 5.27 – Fluxo no nó central de Ω_3	80
Figura 5.28 – Tempo de CPU dos <i>solvers</i>	81
Figura 5.29 – Casca cilíndrica	82
Figura 5.30 – Um oitavo da casca	82
Figura 5.31 – Vistas da casca.....	83
Figura 5.32 – Deslocamento u_y ao longo do eixo $z : h = 0,025a$; modelo 1	84
Figura 5.33 – Deslocamento u_y ao longo do eixo $z : h = 0,025a$; modelo 2	85
Figura 5.34 – Deslocamento u_y ao longo do eixo $z : h = 0,025a$; modelo 3	85
Figura 5.35 – Deslocamento u_y ao longo do eixo $z : h = 0,050a$; modelo 1	86
Figura 5.36 – Deslocamento u_y ao longo do eixo $z : h = 0,050a$; modelo 2.....	86
Figura 5.37 – Deslocamento u_y ao longo do eixo $z : h = 0,050a$; modelo 3.....	87
Figura 5.38 – Deslocamento u_y ao longo do eixo $z : h = 0,075a$; modelo 1	87
Figura 5.39 – Deslocamento u_y ao longo do eixo $z : h = 0,075a$; modelo 2.....	88
Figura 5.40 – Deslocamento u_y ao longo do eixo $z : h = 0,075a$; modelo 3.....	88
Figura 5.41 – Deslocamento u_y ao longo do eixo $z : h = 0,100a$; modelo 1	89
Figura 5.42 – Deslocamento u_y ao longo do eixo $z : h = 0,100a$; modelo 2	89
Figura 5.43 – Deslocamento u_y ao longo do eixo $z : h = 0,100a$; modelo 3	90
Figura 5.44 – DLSARG x J–BiCG: $h = 0,025a$; modelo 2.....	90
Figura 5.45 – DLSARG x J–BiCG: $h = 0,050a$; modelo 1	91
Figura 5.46 – DLSARG x J–BiCG: $h = 0,050a$; modelo 2.....	91
Figura 5.47 – DLSARG x J–BiCG: $h = 0,075a$; modelo 1	92
Figura 5.48 – DLSARG x J–BiCG: $h = 0,075a$; modelo 2.....	92
Figura 5.49 – DLSARG x J–BiCG: $h = 0,100a$; modelo 1	93

Figura 5.50 – DLSARG x J–BiCG: $h = 0,100a$; modelo 2	93
Figura 5.51 – Tempo de CPU dos <i>solvers</i> : $h = 0,025a$; modelo 2	94
Figura 5.52 – Tempo de CPU dos <i>solvers</i> : $h = 0,050a$; modelo 1	94
Figura 5.53 – Tempo de CPU dos <i>solvers</i> : $h = 0,050a$; modelo 2	95
Figura 5.54 – Tempo de CPU dos <i>solvers</i> : $h = 0,075a$; modelo 1	95
Figura 5.55 – Tempo de CPU dos <i>solvers</i> : $h = 0,075a$; modelo 2	96
Figura 5.56 – Tempo de CPU dos <i>solvers</i> : $h = 0,100a$; modelo 1	96
Figura 5.57 – Tempo de CPU dos <i>solvers</i> : $h = 0,100a$; modelo 2	97
Figura 5.58 – Interação solo–estrutura	98
Figura 5.59 – Vista superior da malha: 1 bloco de fundação	99
Figura 5.60 – Vista lateral da malha: 1 bloco de fundação.....	99
Figura 5.61 – Vista 3D da malha do bloco	100
Figura 5.62 – Bulbo de tensões: 1 bloco de fundação	100
Figura 5.63 – Perfil de deslocamento: 1 bloco de fundação.....	101
Figura 5.64 – Pontos da base do bloco	101
Figura 5.65 – Deslocamento vertical do ponto A.....	102
Figura 5.66 – Deslocamento vertical do ponto B.....	102
Figura 5.67 – Deslocamento vertical do ponto C	103
Figura 5.68 – Comparação com as soluções analíticas	103
Figura 5.69 – Tempo de CPU dos <i>solvers</i>	104
Figura 5.70 – Número de iterações/ordem do sistema: 1 bloco de fundação.....	104
Figura 5.71 – Tempo de CPU DLSARG/Tempo de Montagem: 1 bloco de fundação	105
Figura 5.72 – Vista superior da malha: 2 blocos de fundação.....	106
Figura 5.73 – Vista lateral da malha: 2 blocos de fundação.....	106
Figura 5.74 – Tempo de CPU J–BiCG/Tempo de montagem: 2 blocos de fundação	109
Figura 5.75 – Número de iterações/ordem do sistema: 2 blocos de fundação	109

LISTA DE TABELAS

CAPÍTULO 2

Tabela 2.1 – Variável <i>idiscel</i> para o problema genérico	23
Tabela 2.2 – Atualização da malha de elementos de contorno	24

CAPÍTULO 3

Tabela 3.1 – Variável <i>intel</i>	34
--	----

CAPÍTULO 5

Tabela 5.1 – Modelos de EC para a barragem	77
Tabela 5.2 – Dados das malhas para o Modelo 1	83
Tabela 5.3 – Dados das malhas para o Modelo 2	83
Tabela 5.4 – Dados das malhas para o Modelo 3	84

CAPÍTULO 1 – INTRODUÇÃO

Atualmente, os métodos numéricos constituem uma importante ferramenta na análise de problemas complexos de engenharia, usualmente envolvendo características não-lineares ou a reconstituição temporal de sua resposta. Não raro, simulações via modelos de ordem elevada (com milhares de graus de liberdade) tornam-se necessárias a fim de que precisão aceitável possa ser obtida. Sendo assim, a resolução eficiente de problemas de engenharia dentro das restrições de máquina (memória disponível e tempo de processamento) é que talvez seja uma das questões mais interessantes no contexto das pesquisas atuais. Este tema tem sido o objeto de investigação de muitos pesquisadores e constitui um dos pontos centrais deste trabalho.

Nas últimas décadas o Método dos Elementos de Contorno (M.E.C.) tem sido aplicado na resolução de diferentes tipos de problemas (Bonnet, 1999; Wrobel e Aliabadi, 2002). Como principais vantagens desse método citam-se a sua alta precisão, uma significativa redução do tamanho do sistema de equações algébricas resultante e uma maior simplicidade das malhas, uma vez que faz-se necessário discretizar apenas o contorno do problema. Além disso, as soluções fundamentais usadas na formulação do método satisfazem a condição de radiação, ou seja, para domínios abertos, como por exemplo, problemas de interação solo-estrutura, a resposta do problema depende apenas das fontes na região de análise (Bonnet, 1999; Araújo *et al.*, 2001; Araújo *et al.*, 2002; Araújo *et al.*, 2003;). O método também adequa-se para o modelamento de problemas de fratura nos quais há concentração de tensões (Gray *et al.*, 1990; Cruse, 1996). Por outro lado, a ausência de soluções fundamentais para todos os tipos de problemas, o fato do sistema de equações resultante ser não-simétrico e a presença de integrais singulares e *quasi*-singulares podem ser mencionados como desvantagens do método.

Embora as integrais fortemente singulares, as quais existem apenas no sentido do valor principal de Cauchy, demandem cuidado especial, a eficiência do método está

intimamente ligada à eficiência dos algoritmos de integração utilizados para avaliar as integrais fracamente singulares e *quasi*-singulares, que apesar de existirem no sentido ordinário, podem consumir um elevado tempo de processamento, uma vez que requerem um número elevado de pontos de integração para que resultados precisos sejam obtidos. Muitas pesquisas recentes têm se concentrado no estudo e desenvolvimento de algoritmos de integração eficientes, ou seja, que forneçam precisão aceitável sem que o tempo de processamento torne-se proibitivo. Uma das primeiras estratégias desenvolvidas baseia-se na subdivisão dos elementos de integração em vários subelementos e a aplicação da quadratura gaussiana sobre cada subelemento separadamente (Lachat e Watson, 1976; Jun *et al.*, 1985; Voutsinas e Bergeles, 1990). Todavia, tem-se que o grau do polinômio de maior ordem que pode ser exatamente integrado utilizando-se essa técnica é determinado pelo número de pontos de integração considerado em cada subelemento de integração e não pelo número total de pontos de integração por elemento.

Em Telles (1987), encontra-se um processo para a avaliação das integrais fracamente singulares e *quasi*-singulares baseado em uma transformação não-linear de coordenadas polinomial do 3º grau. Tem-se, nesse processo, que os pontos de integração são deslocados para a direção do ponto fonte (ξ) em função da distância do elemento de contorno ao ponto fonte (r), e para valores de r muito grandes este processo é automaticamente desativado (Telles e Oliveira, 1994). Outra estratégia que pode ser adotada para aumentar a eficiência dos algoritmos de integração é a transformação de coordenadas polares triangulares, a qual caracteriza-se pela subdivisão do elemento em subdomínios triangulares e por uma seqüência de mapeamentos de modo a reduzir a ordem da singularidade (Li *et al.*, 1985; Alberto, 2002).

Com o objetivo de aumentar a eficiência dos algoritmos de integração desenvolveu-se, neste trabalho, um processo baseado na combinação da transformação de coordenadas polares triangulares e da transformação não-linear de Telles.

Do ponto de vista de aplicação, o principal objetivo deste trabalho é a aplicação do M.E.C. a problemas escalares e vetoriais em que haja descontinuidades em seus domínios de definição ou envolvendo materiais não-homogêneos, bem como a

problemas de grande ordem. Para tal, aplicam-se técnicas de subestruturação, as quais consistem basicamente na subdivisão do domínio do problema em subregiões e na consideração de modelos de contorno para cada uma delas. A técnica usada neste trabalho fundamenta-se na aplicação de *solvers* iterativos, dentre os quais destacam-se os denominados *solvers* de Krylov (Araújo, 1989; Sonneveld, 1989; van der Vorst, 1992; Axelsson, 1994; Saad, 1996; van der Vorst, 2002; van der Vorst, 2003). Estes *solvers* têm se mostrado bastante eficientes principalmente na análise de problemas de grande porte. Uma característica importante desse tipo de *solver* é o fato de não se realizarem transformações da matriz durante o processo de resolução do sistema de equações. Deste modo, pode-se eliminar completamente o armazenamento e a manipulação de blocos de coeficientes nulos. Estes *solvers* também são de grande importância no desenvolvimento de programas computacionais com opção de processamento paralelo, uma vez que possibilitam a subdivisão do problema por meio de decomposição genérica de domínio (subestruturação) (Araújo *et al.*, 2001; Araújo *et al.*, 2002; Araújo *et al.*, 2003; Araújo *et al.*, 2004).

Em Martins (2000), iniciou-se o desenvolvimento da estratégia de acoplamento genérico EC-EC. Nesse trabalho as subregiões ainda não eram tratadas independentemente. Em Dors (2002), realizou-se a implementação de domínios numericamente desacoplados, mas com uso apenas de elementos de contorno contínuos. Tal fato implicava em uma maior dificuldade no acoplamento de nós comuns a mais de uma interface, devido à descontinuidade de fluxo ou força nestes nós. Para a resolução do sistema de equações fazia-se necessária a aplicação de condições de acoplamento e continuidade. Em Alberto (2002), estendeu-se a estratégia de acoplamento até então desenvolvida para problemas transientes.

Neste trabalho a estratégia de acoplamento genérico EC-EC apresentada em trabalhos anteriores é bastante simplificada mediante a introdução de elementos de contorno descontínuos. Deste modo, a simulação de descontinuidades de fluxo ou força em cantos e arestas em interfaces acopladas é feita através da aplicação de apenas condições de acoplamento entre os nós das várias subregiões que compõem o modelo. Essas subregiões são tratadas de forma independente, portanto não há a necessidade de montar o sistema matricial global acoplado. Durante a resolução do

problema trabalha-se apenas com os subsistemas matriciais associados a cada uma das subregiões como se os mesmos fossem desacoplados.

Este trabalho encontra-se estruturado da seguinte forma:

- No capítulo 2, apresentam-se a formulação do M.E.C. para a análise de problemas estacionários de potencial e elasticidade, e as estratégias de algebrização da equação integral de contorno;
- No capítulo 3, apresenta-se a formulação do *solver* iterativo adotado para a resolução do sistema, aspectos gerais da formulação de acoplamento genérico EC-EC e da resolução do sistema acoplado;
- No capítulo 4, descrevem-se as integrais existentes em formulações padrão do M.E.C. e os algoritmos de integração utilizados na avaliação destas;
- No capítulo 5, apresenta-se uma série de exemplos analisados via programa computacional desenvolvido (NAESY) visando validar e avaliar as estratégias implementadas;
- E finalmente, no capítulo 6, apresentam-se as conclusões da pesquisa realizada e sugerem-se alguns tópicos importantes para trabalhos futuros.

CAPÍTULO 2 – FORMULAÇÃO DIRETA DO M.E.C.

2.1. Introdução

Nas últimas décadas o Método dos Elementos de Contorno (M.E.C.) se estabeleceu como uma importante alternativa de análise dentre os vários métodos numéricos existentes, tendo sido aplicado aos mais diversos tipos de problemas de engenharia, lineares ou não-lineares, envolvendo fluidos e sólidos.

Uma vez estabelecida a equação ou o sistema de equações diferenciais parciais que descrevem o comportamento das variáveis de um dado problema físico, o método consiste basicamente na transformação deste sistema em um sistema de equações integrais envolvendo apenas valores de contorno. Sendo assim, diferentemente dos métodos de domínio, faz-se, naturalmente necessária a discretização apenas do contorno do domínio envolvido para a aproximação da resposta. De outro modo, uma formulação baseada em equações integrais de contorno satisfaz identicamente a resposta do problema no interior de seu domínio de definição, e, aproximadamente, no contorno. A avaliação da resposta no interior do domínio é realizada a partir dos valores de contorno.

Neste capítulo apresenta-se, de forma sucinta, a formulação do M.E.C. para a análise de problemas estacionários de potencial e elasticidade, aspectos gerais relacionados com as estratégias de algebrização da equação integral de contorno e uma descrição da biblioteca de elementos.

2.2. Formulações Integrais do M.E.C.

Partindo-se da equação ou do sistema de equações diferenciais parciais que governam um determinado problema físico e mediante a aplicação de algumas relações matemáticas é possível obter a representação integral de contorno deste problema (Brebbia *et al.*, 1984). Uma alternativa para isso é via a expressão geral de resíduos ponderados. Outra alternativa é via a aplicação direta de relações de reciprocidade que envolvam dois estados-solução do problema em questão, na região de análise (Banerjee, 1994). Note que, partindo-se da expressão geral de resíduos ponderados, faz-se necessário usar o teorema da divergência para a obtenção da formulação integral de contorno. Procedimento análogo também adota-se para a derivação da relação de reciprocidade. Em ambos os casos, funções de ponderação apropriadas serão empregadas para a obtenção da representação integral de contorno da resposta do problema. Particularmente para problemas de potencial, adota-se, normalmente, como relação de reciprocidade a 2ª identidade de Green. Em problemas elastostáticos, descritos pelas equações de Navier, usa-se o teorema de Betti-Rayleigh.

Tem-se, desta forma, para problemas estacionários de potencial, a seguinte representação integral de contorno:

$$c(\xi)u(\xi) + \int_{\Gamma} p^*(\mathbf{x}, \xi)u(\mathbf{x})d\Gamma = \int_{\Gamma} u^*(\mathbf{x}, \xi)p(\mathbf{x})d\Gamma - \int_{\Omega} u^*(\mathbf{x}, \xi)b(\mathbf{x})d\Omega . \quad (2.1)$$

Nesta equação \mathbf{x} e ξ são, respectivamente, o ponto de campo e o ponto fonte, conforme indica a Figura 2.1, $c(\xi)$ é o termo livre da integral, $b(\mathbf{x})$ é o vetor das forças de corpo, $u(\mathbf{x})$ e $p(\mathbf{x})$ são vetores contendo as variáveis de contorno, e, $u^*(\mathbf{x}, \xi)$ e $p^*(\mathbf{x}, \xi)$ são as soluções fundamentais do problema.

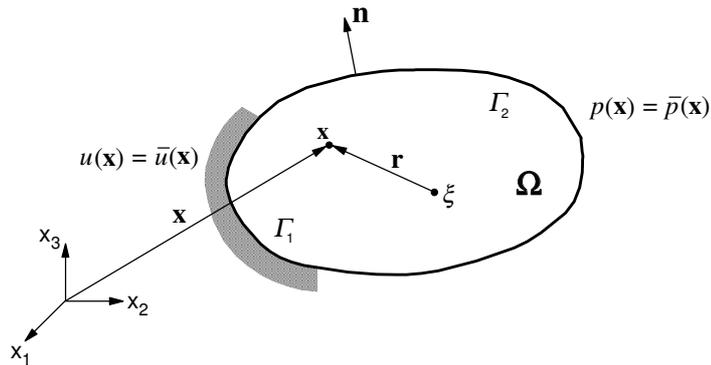


Figura 2.1 – Representação do problema de valor de contorno

Para o caso 3D, tem-se, respectivamente, as seguintes expressões para potenciais e fluxos fundamentais:

$$u^*(\mathbf{x}, \xi) = \frac{1}{4\pi k r} \text{ e } p^*(\mathbf{x}, \xi) = \frac{r_i n_i}{4\pi r^3} . \quad (2.2)$$

Em (2.2), r_i e n_i são, respectivamente, as componentes na direção i do vetor raio, \mathbf{r} , que liga o ponto de campo ao ponto fonte, e do vetor normal ao contorno que aponta para fora do domínio. r é o módulo do vetor raio e k , a constante física do problema, por exemplo, coeficiente de permeabilidade.

O termo livre da integral, $c(\xi)$, surge como conseqüência do comportamento fortemente singular do núcleo fundamental p^* quando ξ está sobre o contorno Γ . Sua obtenção é feita mediante a exclusão do ponto singular do caminho de integração através da definição de uma casca esférica de raio ε centrada em ξ e aplicando-se o limite para $\varepsilon \rightarrow 0$. Deste modo obtém-se:

$$c(\xi) = 1 + \lim_{\varepsilon \rightarrow 0} \int_{\Gamma_\varepsilon} p^*(\mathbf{x}, \xi) d\Gamma . \quad (2.3)$$

Para problemas elastostáticos, tem-se:

$$c_{ik}(\xi)u_i(\xi) + \int_{\Gamma} p_{ik}^*(\mathbf{x}, \xi)u_i(\mathbf{x})d\Gamma = \int_{\Gamma} u_{ik}^*(\mathbf{x}, \xi)p_i(\mathbf{x})d\Gamma + \int_{\Omega} u_{ik}^*(\mathbf{x}, \xi)b_i(\mathbf{x})d\Omega, \quad (2.4)$$

com, i e k variando de 1 a 3.

Na expressão anterior, \mathbf{x} e ξ são, respectivamente, o ponto de campo e o ponto fonte, $c_{ik}(\xi)$ é o termo livre da integral, $b_i(\mathbf{x})$ é o vetor das forças de volume, $u_i(\mathbf{x})$ e $p_i(\mathbf{x})$ são vetores contendo as variáveis de contorno, e, $u_{ik}^*(\mathbf{x}, \hat{\Gamma})$ e $p_{ik}^*(\mathbf{x}, \hat{\Gamma})$ são as soluções fundamentais do problema, as quais, para o caso tridimensional, são dadas por:

$$u_{ik}^*(\mathbf{x}, \xi) = \frac{(1+\nu)}{8\pi E(1-\nu)r} \left[(3-4\nu)\delta_{ik} + \frac{r_i r_k}{r^2} \right] \quad \text{e} \quad (2.5)$$

$$p_{ik}^*(\mathbf{x}, \xi) = -\frac{1}{8\pi(1-\nu)r^2} \left\{ (1-2\nu) \left(n_k \frac{r_i}{r} - n_i \frac{r_k}{r} \right) + \left[\frac{3r_i r_k}{r^2} + (1-2\nu)\delta_{ik} \right] \frac{r_k n_k}{r^2} \right\}.$$

Na expressão (2.5), E e ν são, respectivamente, o módulo de elasticidade e coeficiente de Poisson. r é o módulo do vetor raio, r_i e n_i são, respectivamente, as componentes na direção i do vetor raio e do vetor normal ao contorno que aponta para fora do domínio e δ_{ik} é a função delta de Dirac.

O termo livre da integral, $c_{ik}(\xi)$, é dado por:

$$c_{ik}(\xi) = \delta_{ik} - \frac{1}{8\pi(1-\nu)} \left\{ \lim_{\varepsilon \rightarrow 0} \int_{\Gamma_\varepsilon} \frac{1}{\varepsilon^2} [(1-2\nu)\delta_{ik} + 3\varepsilon_{,i} \varepsilon_{,k}] d\bar{\Gamma}_\varepsilon \right\}. \quad (2.6)$$

A avaliação da resposta no contorno, para os problemas citados anteriormente, é realizada mediante a resolução das equações integrais (2.1) e (2.4) sob a consideração das seguintes condições de contorno gerais:

– Problemas de potencial:

$$\begin{aligned} u(\mathbf{x}) &= \bar{u}(\mathbf{x}), \quad \mathbf{x} \text{ em } \Gamma_1, \text{ e} \\ p(\mathbf{x}) &= k_n \frac{\partial u(\mathbf{x})}{\partial n} = \bar{p}(\mathbf{x}), \quad \mathbf{x} \text{ em } \Gamma_2. \end{aligned} \quad (2.7)$$

– Problemas de elasticidade:

$$\begin{aligned} u_i(\mathbf{x}) &= \bar{u}_i(\mathbf{x}), \quad \mathbf{x} \text{ em } \Gamma_1, \text{ e} \\ p_i(\mathbf{x}) &= \sigma_{ij} n_j = \bar{p}_i(\mathbf{x}), \quad \mathbf{x} \text{ em } \Gamma_2. \end{aligned} \quad (2.8)$$

Uma vez estabelecida a resposta no contorno do domínio de definição do problema, pode-se, através destes valores, obter a resposta em qualquer ponto ξ , pertencente ao domínio de definição do mesmo.

Para pontos internos tem-se que $c(\xi)$ é igual a um, portanto o potencial é determinado a partir da expressão

$$u(\xi) = \int_{\Gamma} u^*(\mathbf{x}, \xi) p(\mathbf{x}) d\Gamma - \int_{\Gamma} p^*(\mathbf{x}, \xi) u(\mathbf{x}) d\Gamma - \int_{\Omega} u^*(\mathbf{x}, \xi) b(\mathbf{x}) d\Omega, \quad (2.9)$$

e o fluxo, $p(\xi)$, a partir de sua derivada em relação às coordenadas cartesianas de ξ . Tem-se:

$$p(\xi) = -k \frac{\partial u(\xi)}{\partial x_i(\xi)}, \quad (2.10)$$

com,

$$\frac{\partial u(\xi)}{\partial x_i(\xi)} = \frac{\partial u(\xi)}{\partial \xi_i} = \int_{\Gamma} \frac{\partial u^*(\mathbf{x}, \xi)}{\partial \xi_i} p(\mathbf{x}) d\Gamma - \int_{\Gamma} \frac{\partial p^*(\mathbf{x}, \xi)}{\partial \xi_i} u(\mathbf{x}) d\Gamma - \int_{\Omega} \frac{\partial u^*(\mathbf{x}, \xi)}{\partial \xi_i} b(\mathbf{x}) d\Omega, \quad (2.11)$$

e as expressões de $\frac{\partial u^*}{\partial \xi_i}$ e $\frac{\partial p^*}{\partial \xi_i}$ em Γ , dadas por:

$$\frac{\partial u^*(\mathbf{x}, \xi)}{\partial \xi_i} = \frac{1}{4\pi} \left(\frac{r_{,i}}{r^2} \right) \quad \text{e} \quad (2.12)$$

$$\frac{\partial p^*(\mathbf{x}, \xi)}{\partial \xi_i} = \frac{1}{4\pi r^3} (n_i - 3r_{,i} r_{,n}). \quad (2.13)$$

Para problemas de elasticidade tem-se que, para pontos internos ao domínio, o termo $c_{ik}(\xi)$ é igual a δ_{ik} . Portanto, a expressão integral de contorno de deslocamentos é dada por

$$u_i(\xi) = \int_{\Gamma} u_{ik}^*(\mathbf{x}, \xi) p_k(\mathbf{x}) d\Gamma - \int_{\Gamma} p_{ik}^*(\mathbf{x}, \xi) u_i(\mathbf{x}) d\Gamma + \int_{\Omega} u_{ik}^*(\mathbf{x}, \xi) b_i(\mathbf{x}) d\Omega, \quad \text{e} \quad (2.14)$$

o estado de tensão no ponto ξ pode ser obtido derivando-se esta equação em relação às suas coordenadas cartesianas, de modo a se obter o tensor de deformações, permitindo-se, assim, a aplicação da lei de Hooke.

Deste modo, obtém-se a seguinte expressão para cálculo do estado de tensão em pontos internos:

$$\sigma_{ij}(\xi) = \int_{\Gamma} u_{ij}^*(\mathbf{x}, \xi) p_l(\mathbf{x}) d\Gamma - \int_{\Gamma} p_{ij}^*(\mathbf{x}, \xi) u_l(\mathbf{x}) d\Gamma + \int_{\Omega} u_{ij}^*(\mathbf{x}, \xi) b_l(\mathbf{x}) d\Omega, \quad (2.15)$$

com,

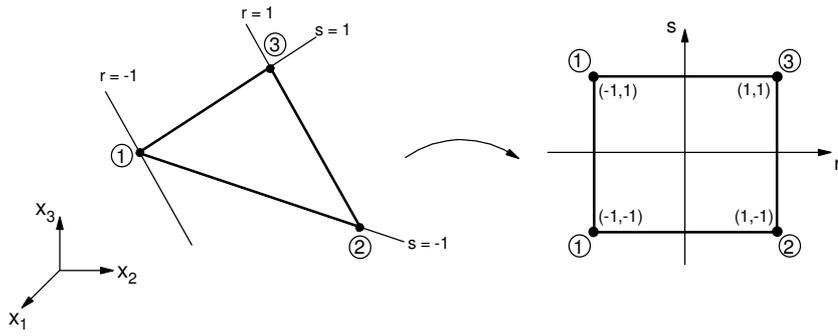
$$u_{ij}^*(\mathbf{x}, \xi) = \frac{1}{8\pi(1-\nu)r^2} [(1-2\nu)(\delta_{lj}r_i + \delta_{li}r_j - \delta_{ij}r_l) + 3r_i r_j r_l] \quad e \quad (2.16)$$

$$p_{ij}^*(\mathbf{x}, \xi) = \frac{G}{4\pi(1-\nu)r^3} \left\{ 3r_n [v(\delta_{li}r_j + \delta_{lj}r_i) + (1-2\nu)\delta_{ij}r_l - 5r_i r_j r_l] + \right. \\ \left. 3v(r_l r_j n_i + r_l r_i n_j) + (1-2\nu)(3r_i r_j n_l + \delta_{li}n_j + \delta_{lj}n_i) - (1-4\nu)\delta_{ij}n_l \right\}. \quad (2.17)$$

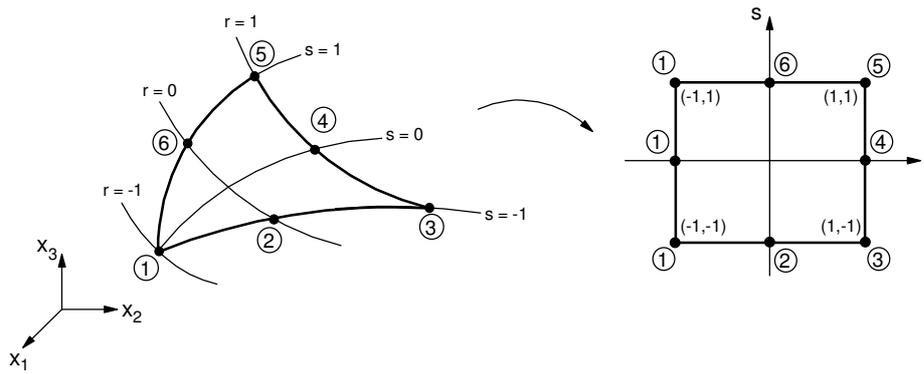
2.3. Algebrização da Equação Integral de Contorno

Devido à complexidade das equações integrais de contorno apresentadas em (2.1) e (2.4), que representam, respectivamente, problemas estacionários de potencial e elastostáticos, a obtenção da solução analítica, para problemas com as mais variadas geometrias e submetidos às mais diversas condições de contorno, torna-se praticamente impossível. Sendo assim, a alternativa para a resolução de tais equações é a aplicação de processos numéricos ou processos de discretização. Estes processos têm por objetivo transformar o sistema de equações integrais em um sistema de equações algébricas que descrevem aproximadamente a resposta dos problemas por meio de um número finito de graus de liberdade.

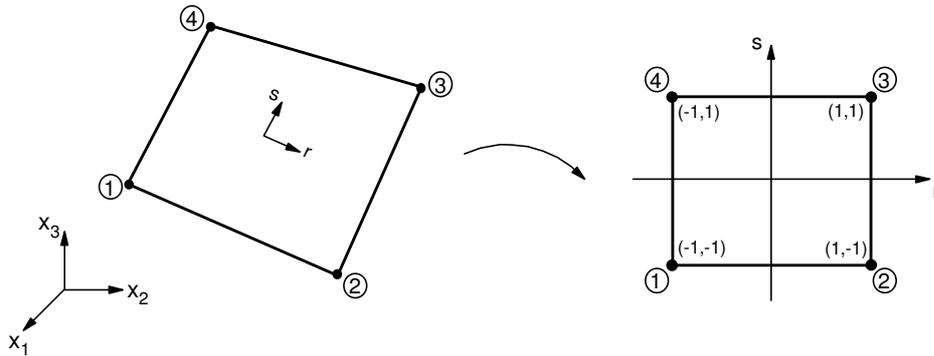
O primeiro passo para a obtenção do sistema de equações algébricas é a discretização do contorno em uma série de elementos finitos, nos quais as variáveis de campo e a geometria são interpoladas a partir de seus valores nodais. Neste trabalho, a discretização é realizada utilizando-se elementos isoparamétricos, para os quais as funções de forma necessárias à interpolação da geometria e das variáveis de campo no interior do elemento são as mesmas. A Figura 2.2 apresenta os elementos de contorno contínuos disponíveis na biblioteca de elementos do algoritmo de acoplamento EC-EC (Dors, 2002).



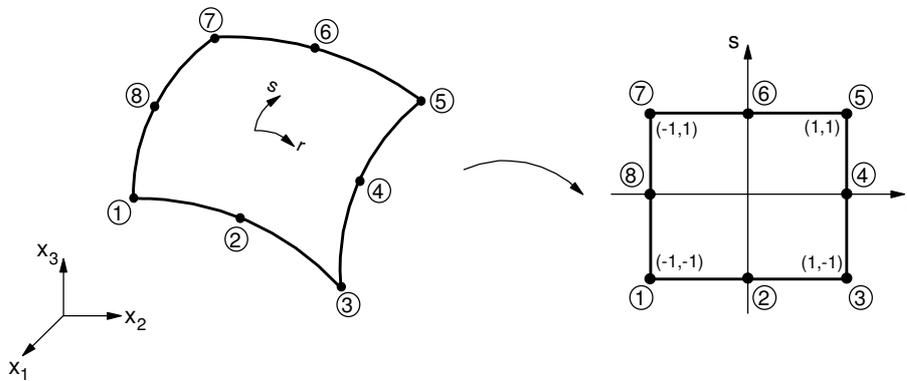
(a) elemento contínuo triangular de 3 nós – q3_3dbe



(b) elemento contínuo triangular de 6 nós – q6_3dbe



(c) elemento contínuo quadrangular de 4 nós – q4_3dbe

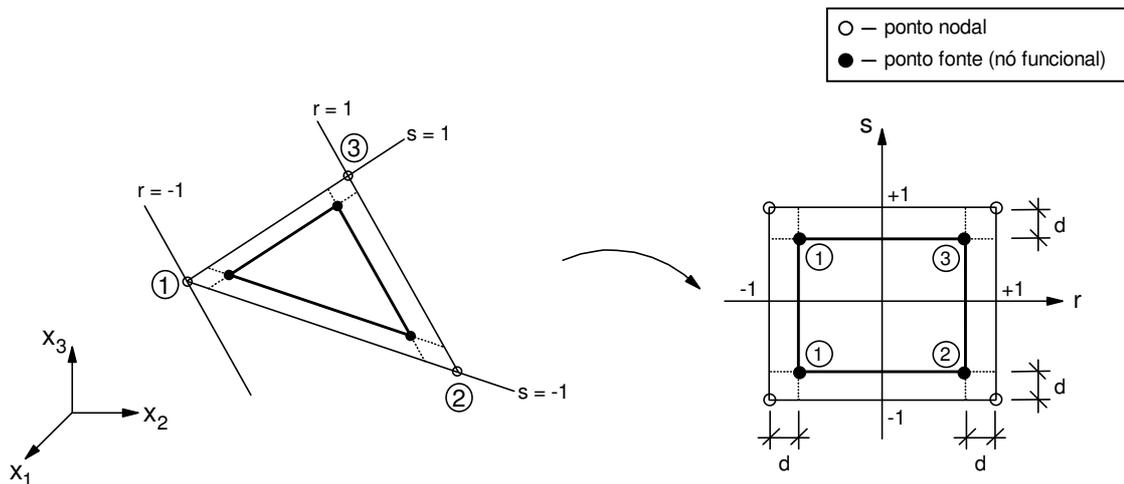


(d) elemento contínuo quadrangular de 8 nós – q8_3dbe

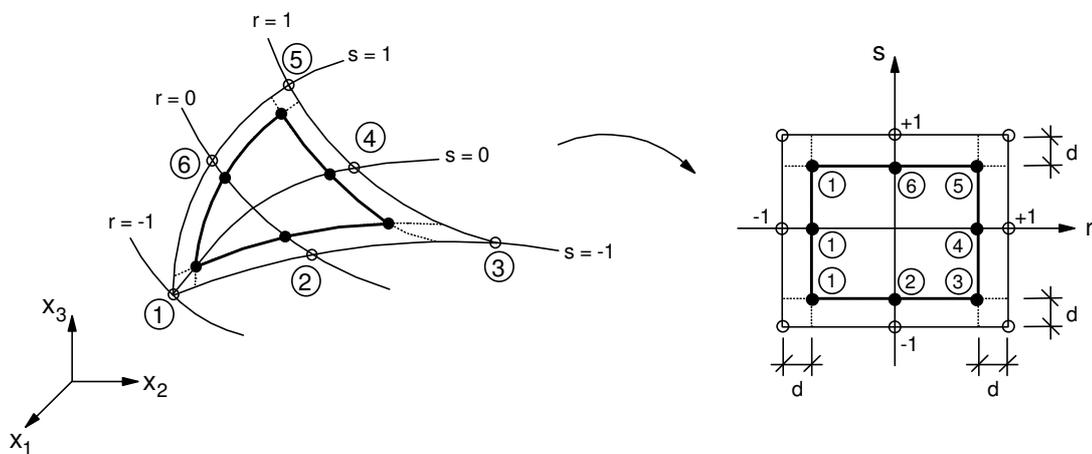
Figura 2.2 – Elementos de contorno contínuos

Acrescentaram-se também a esta biblioteca os elementos descontínuos mostrados na Figura 2.3, os quais fornecem outras opções de simulação de descontinuidade de fluxo ou de força. Eles serão, portanto, muito úteis no desenvolvimento da estratégia de acoplamento, já que facilitarão o acoplamento de nós de interface em que não haja continuidade de forças ou fluxo como ocorre em arestas e cantos.

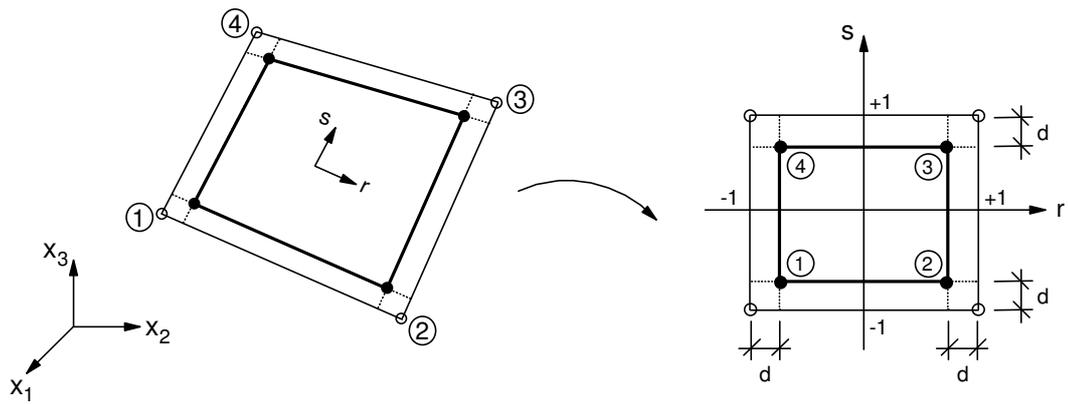
Estes elementos são gerados a partir dos elementos de contorno contínuos através do deslocamento de seus nós, no espaço de coordenadas naturais (r, s) , de um valor d , como mostra a Figura 2.3 (Danson, 1982). Nestes elementos as variáveis de campo (\mathbf{u} ou \mathbf{p}) são interpoladas utilizando-se as funções de forma obtidas para os nós deslocados (nós funcionais), enquanto a geometria do contorno é avaliada a partir das funções de forma do elemento contínuo (associadas aos nós não deslocados).



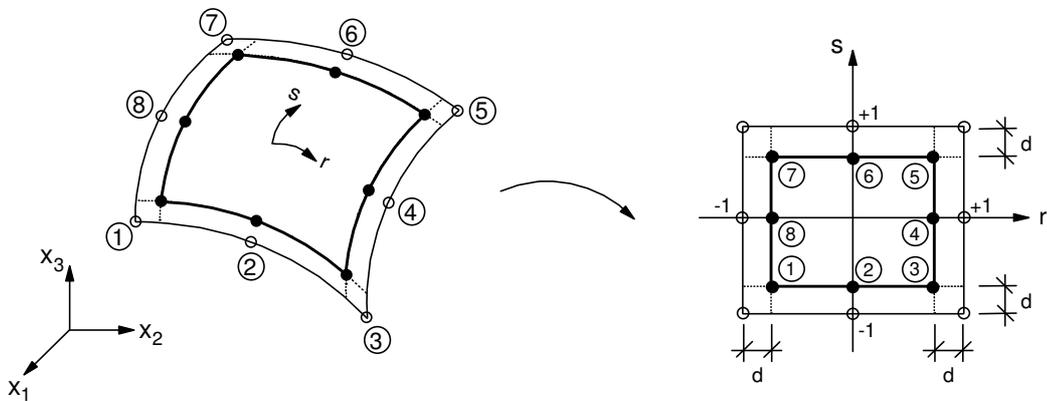
(a) elemento descontínuo triangular de 3 nós



(b) elemento descontínuo triangular de 6 nós



(c) elemento descontínuo quadrangular de 4 nós



(d) elemento descontínuo quadrangular de 8 nós

Figura 2.3 – Elementos de contorno descontínuos

Utilizando-se os elementos de contorno apresentados anteriormente, a geometria do problema e as variáveis de campo podem ser genericamente representadas em função de seus valores nodais por,

$$x_i(r, s) = \sum_{q=1}^{nnoel} h_q(r, s) x_{iq} \quad , \quad (2.18)$$

$$u_i[\mathbf{x}(r, s)] = \begin{cases} \sum_{q=1}^{nnoel} h_q(r, s) u_{iq}, & \text{para elemento de contorno contínuo} \\ \sum_{q=1}^{nnoel} h_q^d(r, s) u_{iq}, & \text{para elemento de contorno descontínuo} \end{cases} \quad \text{e} \quad (2.19)$$

$$p_i[\mathbf{x}(r,s)] = \begin{cases} \sum_{q=1}^{nnoel} h_q(r,s) p_{iq}, & \text{para elemento de contorno contínuo} \\ \sum_{q=1}^{nnoel} h_q^d(r,s) p_{iq}, & \text{para elemento de contorno descontínuo} \end{cases}, \quad (2.20)$$

com, $i = 1, 2, 3$ e $nnoel$ igual ao número de nós do elemento de contorno.

Nas expressões (2.18), (2.19) e (2.20), $x_i(r,s)$, $u_i(r,s)$ e $p_i(r,s)$ são, respectivamente, as coordenadas cartesianas, componentes de potencial/deslocamento e fluxo/força no interior do elemento, e x_{iq} , u_{iq} e p_{iq} são seus valores nodais; $h_q(r,s)$ e $h_q^d(r,s)$ são, respectivamente, a q -ésima função de forma e a q -ésima função de forma modificada do elemento, definidas em função das coordenadas naturais (r,s) .

As funções de forma, $h_q(r,s)$, para os diversos tipos de elemento são apresentadas a seguir.

– *Elemento quadrangular de 4 nós:*

$$h_1(r,s) = 0,25(1-r)(1-s)$$

$$h_2(r,s) = 0,25(1+r)(1-s)$$

$$h_3(r,s) = 0,25(1+r)(1+s)$$

$$h_4(r,s) = 0,25(1-r)(1+s)$$

– *Elemento quadrangular de 8 nós:*

$$h_1(r,s) = 0,25(1-r)(s-1)(r+s+1)$$

$$h_2(r,s) = 0,50(1-s)(1-r^2)$$

$$h_3(r,s) = 0,25(1+r)(s-1)(s-r+1)$$

$$h_4(r,s) = 0,50(1+r)(1-s^2)$$

$$h_5(r, s) = 0,25(1+r)(1+s)(r+s-1)$$

$$h_6(r, s) = 0,50(1+s)(1-r^2)$$

$$h_7(r, s) = 0,25(r-1)(s+1)(r-s+1)$$

$$h_8(r, s) = 0,50(1-r)(1-s^2)$$

Note que os elementos triangulares foram gerados a partir dos elementos quadrangulares colapsando alguns nós dos mesmos. No caso do elemento de 3 nós, colapsaram-se os nós 1 e 4 do elemento quadrangular de 4 nós, e no caso do elemento triangular de 6 nós, colapsaram-se os nós 1, 7 e 8 do elemento quadrangular de 8 nós. Portanto, as funções de forma para os elementos triangulares podem ser escritas em termos das funções de forma dos elementos quadrangulares da seguinte forma:

$$- \begin{cases} h_1(r, s) = h'_1(r, s) + h'_4(r, s) \\ h_2(r, s) = h'_2(r, s) \\ h_3(r, s) = h'_3(r, s) \end{cases}, \text{ para o elemento triangular de 3 nós, e}$$

$$- \begin{cases} h_1(r, s) = h'_1(r, s) + h'_7(r, s) + h'_8(r, s) \\ h_2(r, s) = h'_2(r, s) \\ h_3(r, s) = h'_3(r, s) \\ h_4(r, s) = h'_4(r, s) \\ h_5(r, s) = h'_5(r, s) \\ h_6(r, s) = h'_6(r, s) \end{cases}, \text{ para o elemento triangular de 6 nós,}$$

Nas expressões acima, $h'_q(r, s)$ denota a função de forma do elemento quadrangular em questão. Deste modo, obtém-se:

– *Elemento triangular de 3 nós:*

$$h_1(r, s) = 0,50(1-r)$$

$$h_2(r, s) = 0,25(1+r)(1-s)$$

$$h_3(r, s) = 0,25(1+r)(1+s)$$

– *Elemento triangular de 6 nós:*

$$h_1(r, s) = 0,50 r (r - 1)$$

$$h_2(r, s) = 0,50(1 - s)(1 - r^2)$$

$$h_3(r, s) = 0,25(1 + r)(s - 1)(s - r + 1)$$

$$h_4(r, s) = 0,50(1 + r)(1 - s^2)$$

$$h_5(r, s) = 0,25(1 + r)(1 + s)(r + s - 1)$$

$$h_6(r, s) = 0,50(1 + s)(1 - r^2)$$

As funções de forma modificadas, $h_q^d(r, s)$, para os diversos tipos de elemento são dadas por:

– *Elemento quadrangular de 4 nós:*

Considerando-se que o j -ésimo nó do elemento possui coordenadas naturais, (r_j, s_j) , $r_1 = r_4 = s_1 = s_2 = -1 + d$ e $r_2 = r_3 = s_3 = s_4 = 1 - d$ (Figura 2.3), chega-se a:

$$h_1^d(r, s) = \frac{(r - r_2)(s - r_3)}{(r_1 - r_2)(s_1 - s_3)}$$

$$h_2^d(r, s) = \frac{(r - r_1)(s - s_3)}{(r_2 - r_1)(s_2 - s_3)}$$

$$h_3^d(r, s) = \frac{(r - r_1)(s - s_1)}{(r_3 - r_1)(s_3 - s_1)}$$

$$h_4^d(r, s) = \frac{(r - r_2)(s - s_1)}{(r_4 - r_2)(s_4 - s_1)}$$

– *Elemento quadrangular de 8 nós:*

Considerando-se que o j -ésimo nó do elemento possui coordenadas naturais, (r_j, s_j) , $r_1 = r_7 = r_8 = s_1 = s_2 = s_3 = -1 + d$, $r_3 = r_4 = r_5 = s_5 = s_6 = s_7 = 1 - d$ e $r_2 = r_6 = s_4 = s_8 = 0$ (Figura 2.3), chega-se a:

$$h_2^d(r, s) = \frac{(r - r_1)(r - r_3)(s - s_5)}{r_1 r_3 (s_2 - s_5)}$$

$$h_4^d(r, s) = \frac{(r - r_1)(s - s_3)(s - s_5)}{s_3 s_5 (r_4 - r_1)}$$

$$h_6^d(r, s) = \frac{(s - s_1)(r - r_5)(r - r_7)}{r_5 r_7 (s_6 - s_1)}$$

$$h_8^d(r, s) = \frac{(r - r_3)(s - s_1)(s - s_7)}{s_1 s_7 (r_8 - r_3)}$$

$$h_1^d(r, s) = \frac{(r - r_3)(s - s_5)}{(r_1 - r_3)(s_1 - s_5)} - \frac{1}{2} [h_2^d(r, s) + h_8^d(r, s)]$$

$$h_3^d(r, s) = \frac{(r - r_1)(s - s_7)}{(r_3 - r_1)(s_3 - s_7)} - \frac{1}{2} [h_2^d(r, s) + h_4^d(r, s)]$$

$$h_5^d(r, s) = \frac{(r - r_7)(s - s_1)}{(r_5 - r_7)(s_5 - s_1)} - \frac{1}{2} [h_4^d(r, s) + h_6^d(r, s)]$$

$$h_7^d(r, s) = \frac{(r - r_5)(s - s_3)}{(r_7 - r_5)(s_7 - s_3)} - \frac{1}{2} [h_6^d(r, s) + h_8^d(r, s)]$$

E, adotando-se o mesmo procedimento descrito anteriormente, chega-se às seguintes funções de forma modificadas para os elementos triangulares:

– *Elemento triangular de 3 nós:*

$$h_1^d(r, s) = \frac{(r - r_2)(s - s_3)}{(r_1 - r_2)(s_1 - s_3)} + \frac{(r - r_2)(s - s_1)}{(r_4 - r_2)(s_4 - s_1)}$$

$$h_2^d(r, s) = \frac{(r - r_1)(s - s_3)}{(r_2 - r_1)(s_2 - s_3)}$$

$$h_3^d(r, s) = \frac{(r - r_1)(s - s_1)}{(r_3 - r_1)(s_3 - s_1)}$$

– Elemento triangular de 6 nós:

$$h_2^d(r, s) = \frac{(r - r_1)(r - r_3)(s - s_5)}{r_1 r_3 (s_2 - s_5)}$$

$$h_4^d(r, s) = \frac{(r - r_1)(s - s_3)(s - s_5)}{s_3 s_5 (r_4 - r_1)}$$

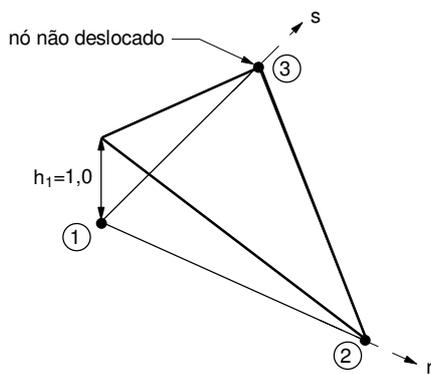
$$h_6^d(r, s) = \frac{(s - s_1)(r - r_5)(r - r_1)}{r_5 r_1 (s_6 - s_1)}$$

$$h_1^d(r, s) = \frac{(r - r_3)(s - s_5)}{(r_1 - r_3)(s_1 - s_5)} + \frac{(r - r_5)(s - s_3)}{(r_1 - r_5)(s_1 - s_3)} - \frac{1}{2} [h_2^d(r, s) + h_6^d(r, s)]$$

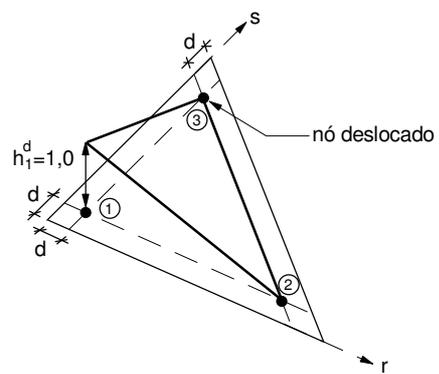
$$h_3^d(r, s) = \frac{(r - r_1)(s - s_1)}{(r_3 - r_1)(s_3 - s_1)} - \frac{1}{2} [h_2^d(r, s) + h_4^d(r, s)]$$

$$h_5^d(r, s) = \frac{(r - r_1)(s - s_1)}{(r_5 - r_1)(s_5 - s_1)} - \frac{1}{2} [h_4^d(r, s) + h_6^d(r, s)]$$

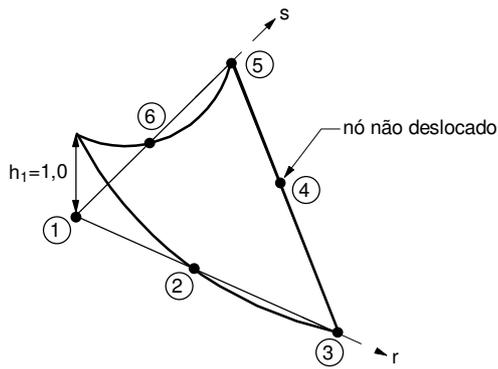
Na Figura 2.4 são apresentadas, a título de ilustração, as funções de forma $h_1(r, s)$ e $h_1^d(r, s)$ associadas ao nó 1 dos vários elementos.



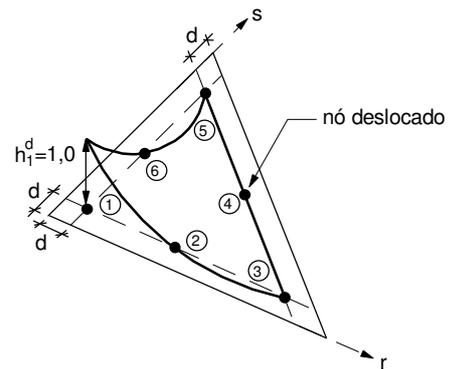
(a) h_1 – elemento contínuo triangular de 3 nós



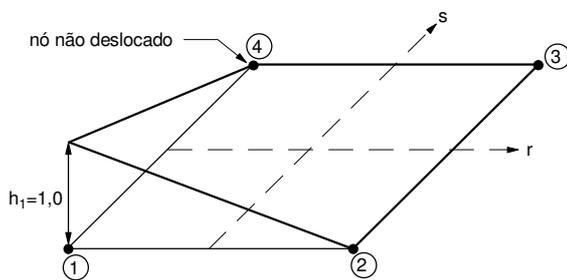
(b) h_1^d – elemento descontínuo triangular de 3 nós



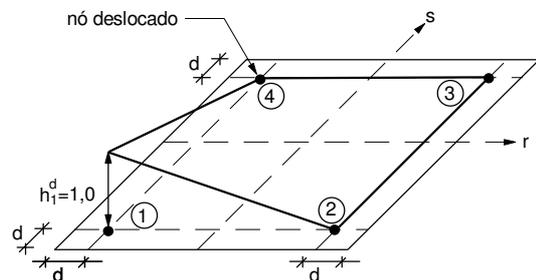
(c) h_1 – elemento contínuo triangular de 6 nós



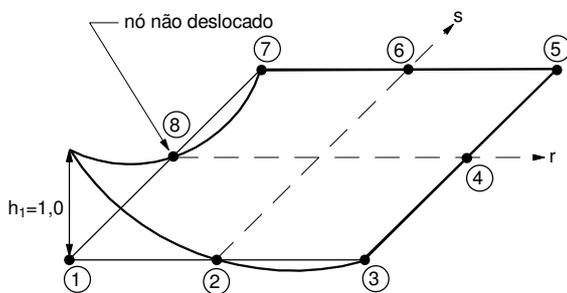
(d) h_1^d – elemento descontínuo triangular de 6 nós



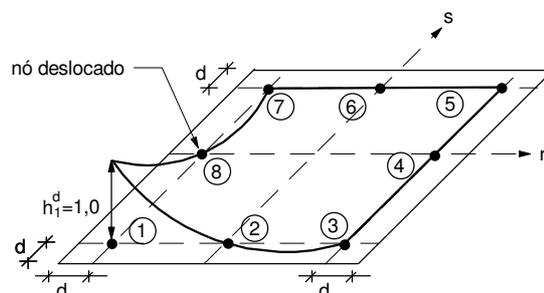
(e) h_1 – elemento contínuo quadrangular de 4 nós



h_1^d – elemento descontínuo quadrangular de 4 nós



(g) h_1 – elemento contínuo quadrangular de 8 nós



(h) h_1^d – elemento descontínuo quadrangular de 8 nós

Figura 2.4 – Função de forma associada ao nó 1 do elemento

Menciona-se, todavia, que nas análises posteriormente apresentadas neste trabalho serão considerados apenas os elementos quadrangulares de 8 nós (q8_3dbe).

A equação integral de contorno discretizada é obtida substituindo-se as expressões (2.18), (2.19) e (2.20) na equação (2.1) ou (2.4). Desta forma obtém-se:

$$c(\xi)_{ik} u_i(\xi) + \sum_{j=1}^{ne} \left(\int_{\Gamma_j} p_{ik}^* [\mathbf{x}(r,s), \xi] \sum_{q=1}^{noel} h_q(r,s) d\Gamma [\mathbf{x}(r,s)] \right) u_{iq}^{(j)} = \sum_{j=1}^{ne} \left(\int_{\Gamma_j} u_{ik}^* [\mathbf{x}(r,s), \xi] \sum_{q=1}^{noel} h_q(r,s) d\Gamma [\mathbf{x}(r,s)] \right) p_{iq}^{(j)} , \quad (2.21)$$

com, ne igual ao número total de elementos, os subscritos i e k variam de 1 ao número de graus de liberdade do problema ($ndofn$), e Γ_j é uma superfície tridimensional.

Quando a integração for realizada sobre um elemento descontínuo, as funções de forma h_q , na expressão (2.21), devem ser convenientemente substituídas pelas correspondentes funções de forma do elemento de contorno descontínuo (h_q^d).

Por fim, o sistema de equações algébricas é obtido escrevendo-se a equação (2.21) para cada um dos nós dos elementos de contorno. O sistema resultante pode ser escrito da seguinte forma matricial:

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{p} . \quad (2.22)$$

Neste sistema, \mathbf{H} e \mathbf{G} são, respectivamente, as matrizes dos coeficientes associadas aos núcleos fundamentais p^* e u^* , \mathbf{u} é o vetor contendo os valores de contorno de potencial ou deslocamento e \mathbf{p} , os de fluxo ou forças de contorno.

Para se resolver o sistema de equações algébricas obtido, deve-se aplicar as condições de contorno dadas nas expressões (2.7) ou (2.8) e realizar trocas de colunas entre as matrizes \mathbf{H} e \mathbf{G} de modo que, ao final, todas as incógnitas estejam do lado esquerdo do sistema de equações. Obtém-se deste modo um sistema da forma,

$$\mathbf{A}\mathbf{x} = \mathbf{b} , \quad (2.23)$$

no qual, \mathbf{A} é a matriz cheia e não-simétrica dos coeficientes de influência, obtida a partir de \mathbf{H} e \mathbf{G} . \mathbf{x} é o vetor das variáveis desconhecidas e \mathbf{b} é o vetor do lado direito obtido multiplicando-se os valores de contorno prescritos pelas respectivas colunas das matrizes de influência (de \mathbf{H} ou \mathbf{G});

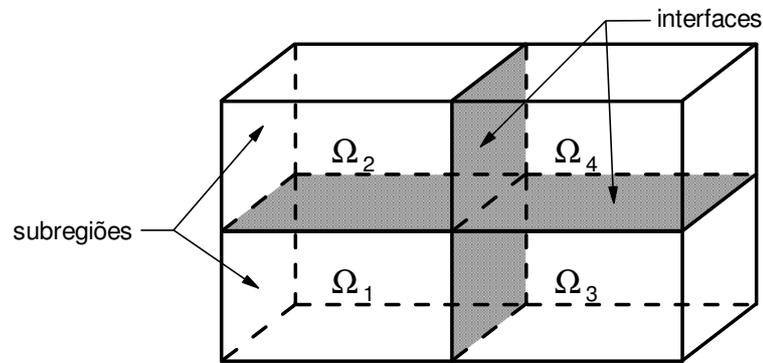
A partir da resolução do sistema de equações (2.23), obtêm-se todos os valores de contorno desconhecidos do problema, com os quais é então possível avaliar a resposta em pontos internos do domínio.

2.4. Geração de Elementos de Contorno Descontínuos

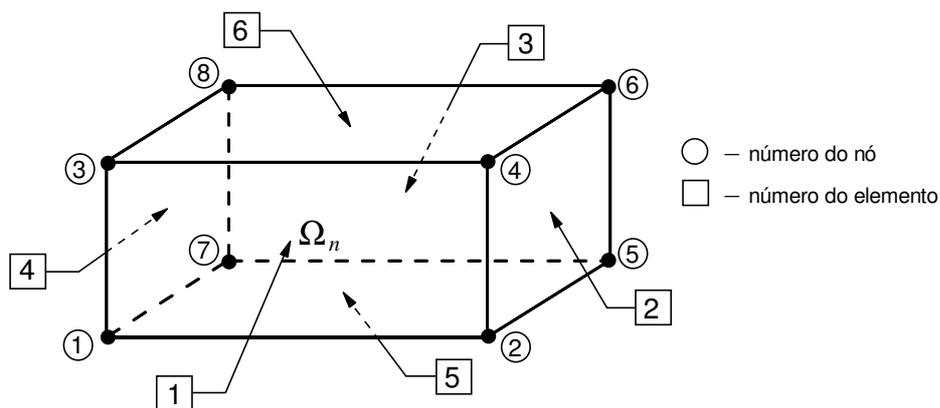
Durante a geração da malha de elementos de contorno do problema, deve-se definir quais elementos da malha serão contínuos e quais descontínuos. Para tal, considera-se a variável *idiscel*, definida, tanto para problemas de potencial quanto de elasticidade, da seguinte forma:

$$idiscel(k) = \begin{cases} 0, & \text{para elemento contínuo} \\ 1, & \text{para elemento descontínuo} \end{cases}.$$

A fim de que a descrição do processo de geração dos elementos descontínuos se torne mais clara, utilizar-se-á, como exemplo, o problema genérico (escalar) apresentado na Figura 2.5. Este problema apresenta um domínio composto por 4 subregiões e utiliza-se um elemento quadrangular de 4 nós (linear) por face de cada subregião para a discretização do modelo.



(a) Domínio do problema



(b) Detalhe da numeração local dos nós e elementos da n -ésima subregião

Figura 2.5 – Problema genérico

Considerando-se, por exemplo, que todos os elementos de interface entre as subregiões sejam descontínuos e que os demais, contínuos, tem-se os seguintes valores para a variável *idiscel* :

Tabela 2.1 – Variável *idiscel* para o problema genérico

Elemento (<i>ie</i>)	<i>idiscel</i> (<i>ie</i>)			
	Subregião 1	Subregião 2	Subregião 3	Subregião 4
1	0	0	0	0
2	1	1	0	0
3	0	0	0	0
4	0	0	1	1
5	0	1	0	1
6	1	0	1	0

Menciona-se que a variável *idiscel* é a única do processo de geração de elementos descontínuos que o usuário precisa fornecer. Após o estabelecimento desta variável, o processo é totalmente automatizado. A rotina *discelems* responsabiliza-se pela geração da malha de elementos de contorno descontínuos a partir da malha elementos de contorno contínuos fornecida originalmente, a qual será agora definida pelas seguintes variáveis:

- *nnofunc* : número de nós funcionais;
- *incde* : armazena a incidência dos elementos após a atualização da malha;

- *coordde* : armazena as coordenadas cartesianas dos pontos nodais da malha atualizada;
- *nmultde(i)* : armazena o número de nós múltiplos do *i*-ésimo nó;
- *imultde(i, nmultde(i))* : armazena o número dos nós múltiplos do *i*-ésimo nó;

De forma resumida, a rotina *discelems* divide-se em 2 partes. A primeira parte consiste nos seguintes passos genéricos:

- Inicialmente, faz-se *nnofunc* igual a *nno* (número de nós da malha inicial de elementos de contorno da subregião em questão);
- Em seguida, identificam-se, via variável *idiscel*, descrita anteriormente, os elementos contínuos e descontínuos e definem-se as novas incidências, *incde*, conforme a seguinte estratégia:
 - a) Se o elemento for contínuo atribuem-se à variável *incde* os valores de incidência originalmente fornecidos pelo usuário (variável *inc*);
 - b) Se o elemento for descontínuo, adicionam-se novos nós ao modelo através do incremento da variável *nnofunc* e a variável *incde* receberá o número dos nós acrescentados;
- Por fim, eliminam-se nós desnecessariamente gerados.

Considerando-se, por exemplo, a subregião 1 do problema apresentado na Figura 2.5 tem-se os seguintes valores para as incidências antes (*inc*) e após (*incde*) a execução da rotina *discelems*:

Tabela 2.2 – Atualização da malha de elementos de contorno

<i>nno</i> = 8					<i>nnofunc</i> = 16				
elemento	<i>inc</i> (<i>ne, nnoel</i>)				elemento	<i>incde</i> (<i>ne, nnoel</i>)			
1	1	2	4	3	1	1	2	4	3
2	2	5	6	4	2	9	10	11	12
3	5	7	8	6	3	5	7	8	6
4	1	3	8	7	4	1	3	8	7
5	1	7	5	2	5	1	7	5	2
6	3	4	6	8	6	13	14	15	16

Na Tabela 2.2 *ne* e *nnoel* são, respectivamente, o número de elementos da subregião e o número de nós por elemento.

A segunda parte da rotina corresponde à geração das coordenadas cartesianas dos pontos nodais dos elementos descontínuos (ver Figura 2.3), que serão armazenadas na variável *coordde*, e das variáveis *nmultde* e *imultde* como descrito a seguir:

- Se *nnofunc* for igual a *nno* tem-se que a malha atualizada é idêntica à malha original, portanto as variáveis *coordde*, *nmultde* e *imultde* receberão os mesmos valores das variáveis *coord*, *nmult* e *imult*;
- Caso contrário *coordde* é inicializada com os valores de *coord* (coordenadas dos pontos nodais do elemento fornecidas pelo usuário) e as variáveis *nmultde* e *imultde* com o valor zero, e procede-se da seguinte forma:
 - a) Faz-se uma pesquisa sobre o número de nós da subregião buscando-se identificar nós com coordenadas cartesianas iguais. Se dois nós *i* e *j* possuírem coordenadas iguais, as variáveis *nmultde* e *imultde* são atualizadas segundo as expressões:

$$\begin{cases} nmultde(i) = nmultde(i) + 1 \\ imultde(i, nmultde(i)) = j \\ nmultde(j) = nmultde(j) + 1 \\ imultde(j, nmultde(j)) = i \end{cases} .$$

- b) Identificam-se os elementos descontínuos (*idiscel* = 1) e as coordenadas dos pontos nodais destes elementos são calculadas a partir da expressão:

$$x_i = \sum_{q=1}^{moel} h'_q \hat{x}_{iq} , \quad i = 1, 2, 3 , \quad (2.24)$$

em que, x_i é a componente na direção *i* do vetor de coordenadas nodais do elemento, \hat{x}_{iq} é a componente na direção *i* do vetor de coordenadas

nodais do q -ésimo nó do elemento e h'_q é a função de forma para o elemento de contorno contínuo avaliada nos nós deslocados do elemento.

Portanto, tem-se, após a execução da rotina *discelems*, a seguinte malha de elementos de contorno para a subregião 1 do problema genérico:

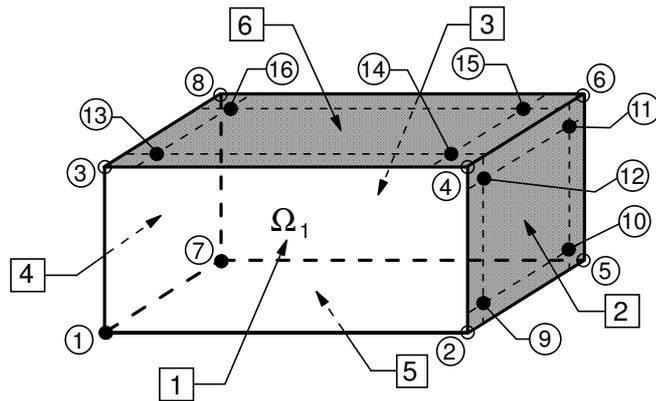


Figura 2.6 – Malha de elementos de contorno atualizada para $\hat{\mathbf{U}}_1$

CAPÍTULO 3 – ESTRATÉGIA DE DECOMPOSIÇÃO GENÉRICA DE DOMÍNIO

3.1. Introdução

Estratégias de decomposição genérica de domínio ou subestruturação são necessárias na análise de problemas cujos domínios sejam formados por materiais diferentes (não-homogêneos) e/ou apresentem descontinuidades de domínio, como por exemplo, problemas de acoplamento solo/fundação e de mecânica da fratura. Essas também são de grande importância na resolução de problemas de grande porte utilizando-se técnicas de processamento paralelo (Hsiao *et al.*, 2000; Dors, 2002; Alberto, 2002).

Tais estratégias consistem basicamente na subdivisão do domínio do problema em um conjunto de subregiões e na aplicação de modelos de elementos de contorno para cada uma delas separadamente. A solução do sistema de equações resultante é avaliada mediante a consideração, além das condições de contorno, de condições a serem impostas nas interfaces entre subregiões (acoplamento).

A estratégia de acoplamento genérico EC-EC (estratégia de subestruturação) adotada neste trabalho baseia-se na utilização de elementos de contorno descontínuos e de *solvers* iterativos, os quais permitirão que a solução do sistema global seja avaliada sem que o mesmo seja explicitamente montado. Sendo assim os blocos de coeficientes nulos da matriz serão completamente excluídos da etapa de resolução do sistema de equações.

Neste capítulo apresenta-se o algoritmo do *solver* iterativo, a formulação de acoplamento genérico EC-EC, o esquema de montagem do sistema matricial acoplado (pesquisa e variáveis de acoplamento) e a estratégia de resolução do sistema.

3.2. Solvers Iterativos

Por serem eficientes e possibilitarem a resolução de sistemas algébricos sem a transformação da respectiva matriz, os *solvers* iterativos têm sido cada vez mais utilizados para a resolução de sistemas algébricos em problemas lineares e não-lineares. Sua importância tem se evidenciado principalmente na análise de problemas de grande porte (com algumas centenas de milhares de graus de liberdade) uma vez que a aplicação destes reduz, significativamente, o tempo de resolução de sistemas de ordens elevadas, bem como, o espaço de memória para armazenamento das matrizes destes sistemas.

Dentre os diversos *solvers* iterativos existentes destacam-se os denominados *solvers* iterativos de Krylov (Hageman e Young, 1981; Araújo, 1989; Kane, 1990; Mansur *et al.*, 1992; Barra *et al.*, 1992; Barra *et al.*, 1993). Estes *solvers* baseiam-se no subespaço de Krylov e suas versões preconditionadas apresentam excelentes índices de convergência. Como exemplo deste *solvers* citam-se os algoritmos de Lanczos, de Gradiente Bi-conjugado (Bi-CG) e de Resíduo Mínimo Generalizado (GMRES).

Neste trabalho adotou-se, como *solver* iterativo, apenas o algoritmo de Gradiente Bi-conjugado. Para se derivar este algoritmo, parte-se inicialmente do algoritmo de tridiagonalização de Lanczos (Wilkinson, 1965), através do qual deriva-se diretamente o algoritmo de Lanczos, e a partir deste, pode-se obter o algoritmo de Gradiente Bi-conjugado. Na Figura 3.1 apresenta-se, sucintamente, o algoritmo do *solver* iterativo J-BiCG. Maiores detalhes podem ser encontrados em Araújo (1989), Mansur *et al.* (1992), Saad (1996), Araújo e Martins (2001), Araújo *et al.* (2001) e Araújo *et al.* (2004).

O processo iterativo apresentado abaixo é geral e pode ser aplicado a quaisquer tipos de sistemas não singulares sendo que a convergência geralmente é garantida em no máximo n iterações, com n menor ou igual à ordem da matriz do sistema (N). Entretanto, devido a erros de truncamento que podem ocorrer durante a execução das operações a convergência pode ser obtida para $n \geq N$.

Para evitar que tais problemas ocorram, bem como, melhorar a convergência destes *solvers*, utilizam-se estratégias de preconditionamento (\mathbf{Q}). Neste trabalho utiliza-se a matriz de preconditionamento de Jacobi, que é definida como a matriz dos elementos diagonais da matriz de coeficientes (\mathbf{A}).

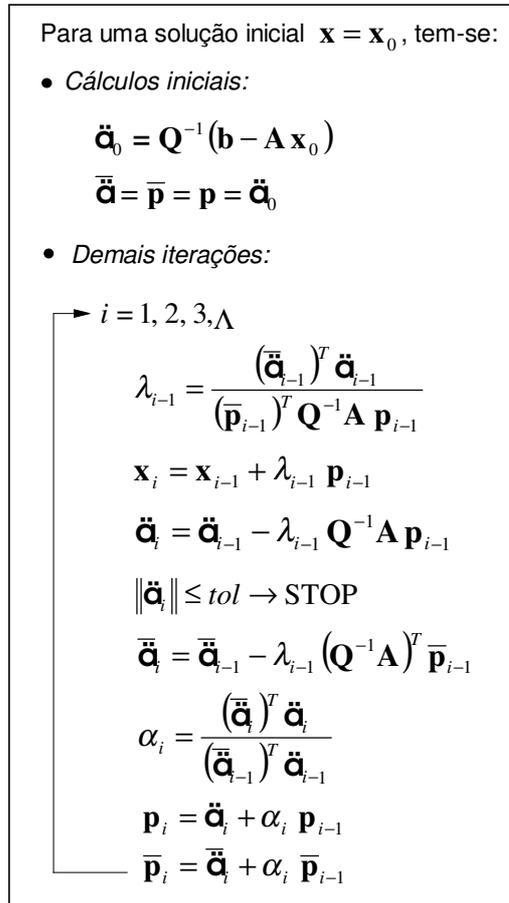


Figura 3.1 – Algoritmo do *solver* de Gradiente Bi-Conjugado com matriz de preconditionamento \mathbf{Q}

3.3. Acoplamento EC-EC Genérico

O algoritmo de acoplamento EC-EC genérico é desenvolvido escrevendo-se a expressão (2.22) para cada uma das subregiões que compõem o domínio. Assim, resulta:

$$\mathbf{H}^k \mathbf{u}^k = \mathbf{G}^k \mathbf{p}^k, \text{ com } k = 1, \Lambda, ns, \quad (3.1)$$

onde ns é igual ao número total de subregiões.

Gera-se, portanto, um conjunto de ns sistemas de equações, o qual deve ser organizado de modo a obter-se, no termo do lado direito, os valores de contorno prescritos e os fluxos ou forças de interface (incógnitas), e, no termo do lado esquerdo, os valores incógnitos de contorno e os potenciais ou deslocamentos de interface (incógnitas). Sendo assim, resulta:

$$\begin{bmatrix} \mathbf{A}_b^k & \mathbf{H}_i^k \end{bmatrix} \begin{Bmatrix} \mathbf{x}_b^k \\ \mathbf{u}_i^k \end{Bmatrix} = \begin{bmatrix} \mathbf{B}_b^k & \mathbf{G}_i^k \end{bmatrix} \begin{Bmatrix} \mathbf{y}_b^k \\ \mathbf{p}_i^k \end{Bmatrix}, \text{ com } k = 1, \Lambda, ns, \quad (3.2)$$

No sistema acima, as matrizes \mathbf{A}_b^k e \mathbf{B}_b^k , e os vetores, \mathbf{x}_b^k e \mathbf{y}_b^k , resultam da introdução das condições de contorno do problema e são obtidos a partir das seguintes expressões:

$$\mathbf{A}_b^k = \begin{bmatrix} \mathbf{H}_{c(u)}^k & -\mathbf{G}_{c(p)}^k \end{bmatrix}, \quad \mathbf{x}_b^k = \begin{Bmatrix} \mathbf{u}_b^k \\ \mathbf{p}_b^k \end{Bmatrix}, \quad (3.3)$$

$$\mathbf{B}_b^k = \begin{bmatrix} -\mathbf{H}_{c(u)}^k & -\mathbf{G}_{c(p)}^k \end{bmatrix}, \quad \mathbf{y}_b^k = \begin{Bmatrix} \bar{\mathbf{u}}_b^k \\ \bar{\mathbf{p}}_b^k \end{Bmatrix}. \quad (3.4)$$

Em (3.2), i indica superfície de interface (com potencial/deslocamento e fluxo/força desconhecidos), e, \mathbf{x}_b^k e \mathbf{y}_b^k são, respectivamente, os vetores com os valores de contorno desconhecidos e prescritos da subregião k . \mathbf{u}_i^k e \mathbf{p}_i^k são, respectivamente, os vetores de potencial/deslocamento e de fluxo/força de interface da subregião k . Nas expressões (3.3) e (3.4), o índice $c(u)$ define a parte do contorno da k -ésima subregião com \mathbf{u} incógnito e o índice $c(p)$, a parte do contorno da k -ésima subregião com \mathbf{p} incógnito.

Para que o sistema em (3.2) seja resolvido deve-se, naturalmente, aplicar condições de acoplamento entre as interfaces de subregiões comuns. Estas são aplicadas a nós acoplados, ou seja, a nós pertencentes a duas subregiões diferentes, que tenham mesma posição geométrica, e vetores normais unitários externos correspondentes com a mesma direção, mas sentidos opostos, e são dadas por

$$\mathbf{u}_i = \mathbf{u}_i \quad (\text{compatibilidade}) \quad e, \quad (3.5)$$

$$\mathbf{p}_i = -\mathbf{p}_i \quad (\text{equilíbrio}) \quad , \quad (3.6)$$

em que, o subscrito i denota interface.

Nos trabalhos anteriores, como consideravam-se apenas elementos de contorno contínuos no algoritmo de acoplamento EC-EC além da aplicação das condições de acoplamento, fazia-se necessário introduzir um número adicional de equações para que o número de incógnitas e de equações do problema fosse igualado. Este conjunto adicional de equações era obtido mediante a aplicação de condições de continuidade entre os nós de interface comum, ou seja, nós pertencentes a mais de uma interface como mostra a Figura 3.2 (Dors, 2002; Alberto, 2002). Estes nós apresentam descontinuidades de fluxo ou força e geralmente ocorrem quando da simulação de arestas ou cantos em interfaces acopladas. A aplicação das condições de continuidade tinha como objetivo eliminar incógnitas desnecessárias na análise.

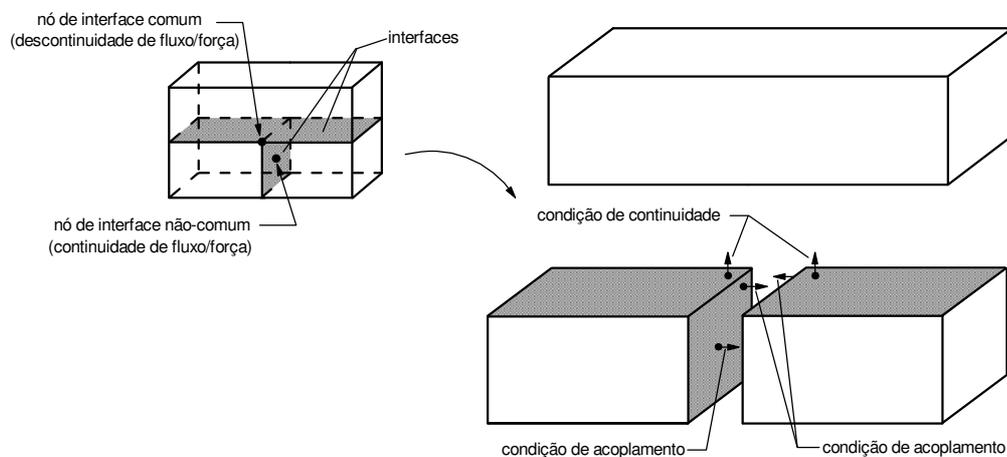


Figura 3.2 – Condições de acoplamento e continuidade

Na presente estratégia, que considera elementos descontínuos, o algoritmo de acoplamento é bastante simplificado. Tem-se que um nó de uma determinada subregião poderá estar acoplado a apenas um outro nó. Portanto, existirão sempre duas equações para cada par \mathbf{u} e \mathbf{p} a ser determinado, deste modo a aplicação de condições de continuidade torna-se desnecessária.

Sendo assim, através da aplicação apenas das condições de acoplamento entre os nós das várias subregiões e da introdução das condições de contorno do problema pode-se obter o sistema matricial acoplado na forma $\mathbf{A} \mathbf{x} = \mathbf{b}$.

Considere, por exemplo, o domínio $\hat{\mathbf{U}}$ subdividido em três subregiões apresentado na figura abaixo.

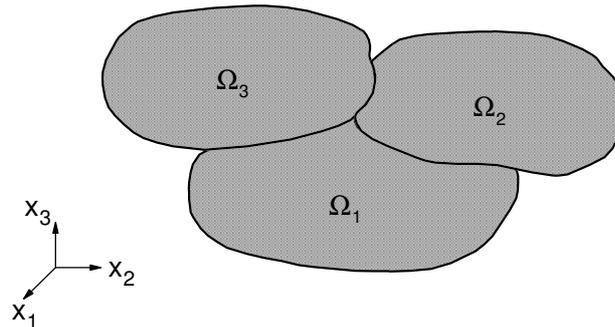


Figura 3.3 – Subdomínios acoplados

Tem-se para este modelo, após a aplicação das condições de acoplamento e da introdução das condições de contorno, um sistema matricial acoplado resultante com o seguinte aspecto:

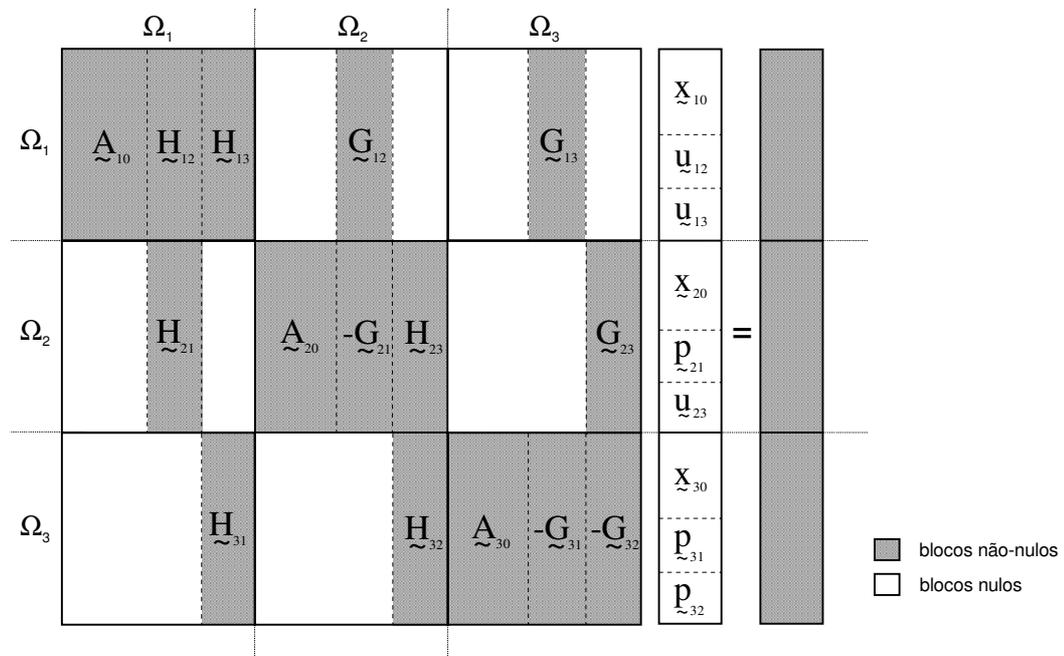


Figura 3.4 – Matriz global do sistema acoplado

Observa-se que a matriz global do sistema apresenta muitos blocos nulos, o que implicaria na utilização de grande espaço de memória para o armazenamento do sistema matricial global acoplado. Todavia, a estratégia de acoplamento genérico adotada baseia-se na utilização de *solvers* iterativos, os quais permitirão que a resolução do sistema seja realizada sem que o sistema global seja explicitamente montado.

A resolução do sistema mediante a utilização de *solvers* iterativos funcionará basicamente da seguinte forma: trabalhar-se-á com as matrizes de cada uma das subregiões, realizando-se os produtos matriz-vetor e vetor-vetor como se os sistemas fossem desacoplados. Na verdade, a cada iteração do *solver*, será realizada a transferência dos valores de acoplamento entre subregiões. Ao final do processo iterativo os valores de contorno serão atualizados.

3.4. Interfaces entre Subregiões

Os elementos de contorno pertencentes às interfaces entre as várias subregiões são identificados, tanto para problemas de potencial quanto de elasticidade, pela variável *intel*. Todavia, como no primeiro caso temos um problema escalar e no segundo um problema vetorial, esta variável é definida de forma diferente para estes dois tipos de problema. Para problemas de potencial tem-se:

$$intel = \begin{cases} 0, & \text{indica que o } k - \text{ésimo elemento não é de interface} \\ 1, & \text{indica que o } k - \text{ésimo elemento é de interface} \end{cases} .$$

Considerando-se, por exemplo, a subregião 1 do problema genérico apresentado na Figura 2.5, tem-se os seguintes valores para a variável *intel*:

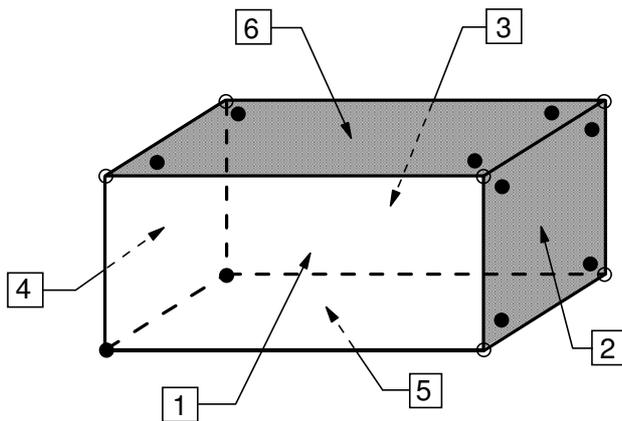


Figura 3.5 – Malha de elementos de contorno $\hat{\mathbf{U}}_1$

Tabela 3.1 – Variável *intel*

Elemento (<i>ie</i>)	<i>intel</i> (<i>ie</i>)
1	0
2	1
3	0
4	0
5	0
6	1

Note que utilizam-se elementos descontínuos nas interfaces entre subregiões, pois trata-se de uma aresta (descontinuidades).

Por outro lado, para problemas de elasticidade tem-se que a variável *intel* além de identificar os elementos de interface também fornecerá o código de acoplamento entre estes elementos. Este é definido por direção (graus de liberdade) como mostra a expressão a seguir:

$intel(k) = ijk$; 'ijk' é o código de acoplamento do *k*-ésimo elemento ;

se, por exemplo, $i = 0$, tem-se que o *k*-ésimo elemento não está acoplado na direção *i*, por outro lado, se $i = 1$, o *k*-ésimo elemento está acoplado na direção *i*.

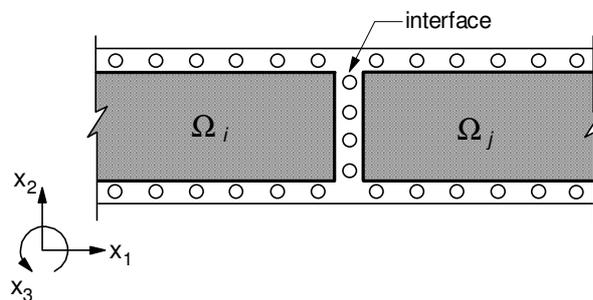


Figura 3.6 – Código de acoplamento: elasticidade

Para o problema de elasticidade apresentado acima tem-se que o deslocamento na direção x_2 na interface entre as subregiões não está impedido, ou seja, haverá um deslizamento relativo entre as subregiões nesta direção. Sendo assim, o código de acoplamento para os elementos de interface será " $intel(ie)=100$ ", ou seja, os elementos estão acoplados na direção x_1 , e desacoplados nas direções x_2 e x_3 .

3.5. Pesquisa de Acoplamento

Inicia-se a pesquisa de acoplamento com a geração da variável lógica $icprob$, a qual indica a existência ou não de domínios acoplados, como segue:

$$icprob = \begin{cases} \cdot false, & \text{não existem domínios acoplados} \\ \cdot true, & \text{existem domínios acoplados} \end{cases} .$$

Esta variável é inicializada como valor " $\cdot false \cdot$ ". Em seguida, busca-se identificar a existência de elementos de interface, via variável $intel$; caso seja detectado algum elemento de interface atribui-se o valor " $\cdot true \cdot$ " à variável $icprob$. Este procedimento é realizado pela rotina $coup_prob$ responsável pela identificação de problemas de acoplamento.

Se ao final do procedimento anterior a variável $icprob$ for igual a " $\cdot true \cdot$ ", passa-se à realização da pesquisa de acoplamento propriamente dita. Para tal, introduziram-se as rotinas $research_coupl$ (problemas escalares) e $research_coupl3D$ (problemas vetoriais), responsáveis pela identificação de nós acoplados entre duas subregiões e que utilizam as seguintes variáveis:

- $icoupli(ino)$: armazena o número do nó da região $jsub$ acoplado ao nó ino da subregião $isub$;
- $icoupsi(ino)$: armazena o número da subregião à qual o nó acoplado ao nó ino pertence;
- $icoupnj(jno)$: armazena o número do nó da subregião $isub$ acoplado ao nó jno da subregião $jsub$;

– $icoupsj(jno)$: armazena o número da subregião à qual o nó acoplado ao nó jno pertence.

De forma geral estas rotinas seguem os seguintes passos:

- Faz-se uma pesquisa sobre o número de subregiões do modelo comparando-se as coordenadas dos nós da subregião $isub$ com as coordenadas dos nós das demais subregiões;
- Caso sejam identificados nós com as mesmas coordenadas, respeitando-se uma tolerância previamente estabelecida, geram-se as variáveis $icoupni$, $icoupsi$, $icoupnj$ e $icoupsj$. Por exemplo, se os nós ino e jno pertencentes, respectivamente, às subregiões $isub$ e $jsub$ são acoplados, tem-se:

$$\left\{ \begin{array}{l} icoupni(ino) = jno \\ icoupsi(ino) = jsub \\ icoupnj(jno) = ino \\ icoupsj(jno) = isub \end{array} \right. ;$$

– Atribui-se o valor zero às variáveis $icoupni$, $icoupsi$, $icoupnj$ e $icoupsj$ referentes aos nós dos elementos das subregiões $isub$ e $jsub$ que não pertencem a elementos de interface. Para isto utiliza-se a variável $ifip$, que já existia no algoritmo e era utilizada para indicar as condições de contorno do problema. Na atual versão acrescentou-se a ela a opção de identificar nós pertencentes a elementos de interface, por exemplo, se $ifip(ino) = 2$ tem-se que o nó ino pertence a um elemento de interface.

Para o problema genérico apresentado no capítulo 2 (Figura 2.5) tem-se que as condições de acoplamento entre os nós 12 e 10 das subregiões 1 e 3, respectivamente, são dadas por $icoupni(12) = 10$, $icoupsi(12) = 3$, $icoupnj(10) = 12$ e $icoupsj(10) = 1$.

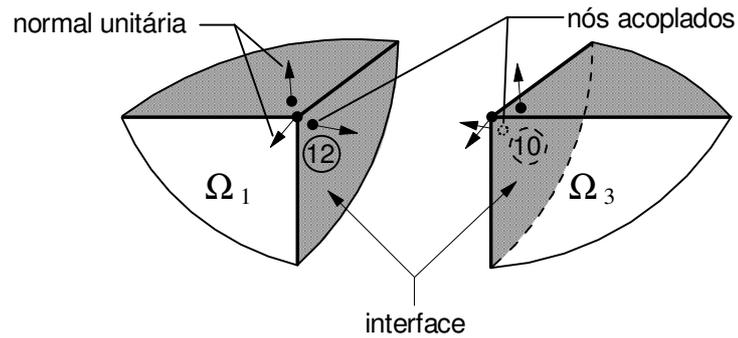


Figura 3.7 – Nós acoplados

3.6. Resolução do Sistema Acoplado

A resolução do sistema matricial global acoplado será realizada através da utilização dos *solvers* iterativos, os quais realizam, ao longo do processo iterativo, basicamente, operações de multiplicação matriz–vetor e vetor–vetor. As iterações são realizadas até que se encontre a solução do problema respeitada uma tolerância preestabelecida (*tol*).

Como mencionado anteriormente, deseja-se resolver o sistema matricial global de equações sem que o mesmo seja explicitamente montado. Sendo assim, as operações citadas acima serão realizadas de forma independente para cada uma das subregiões. Para isso, faz-se portanto apenas necessário acessar, quando do cálculo da contribuição de uma dada subregião no produto matriz–vetor em questão, as correspondentes variáveis dessa subregião. Em geral, as variáveis de nós de interface serão atualizadas ao longo do processo iterativo. No programa computacional, essa atualização ou transferência de dados (valores de **u** e **p**) entre duas subregiões é realizada pela rotina *transfer_sol*.

De maneira resumida tem-se, dentro do processo iterativo, um “*loop*” sobre o número de subregiões, onde, para cada subregião, computa-se a correspondente contribuição no produto matriz–vetor associado à matriz global do sistema acoplado. No início de cada nova iteração, os valores das variáveis acopladas são atualizados, como mostra a figura abaixo.

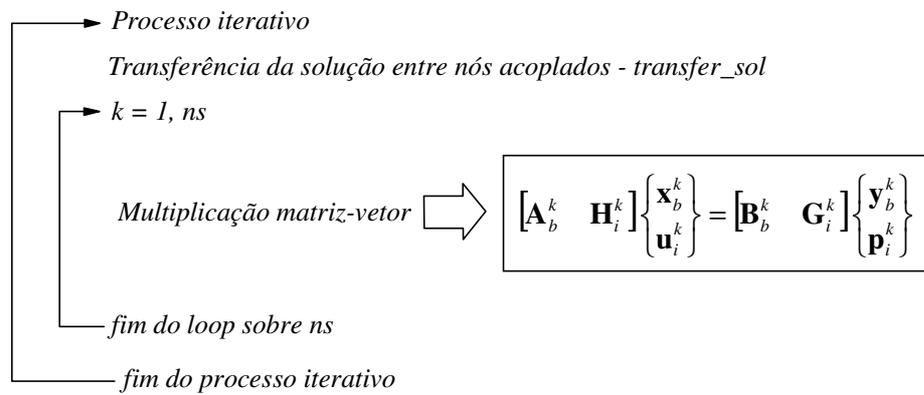


Figura 3.8 – Processo iterativo

Por processo similar ao descrito acima, realiza-se o produto $\mathbf{A}^T \times \mathbf{v}$, que também ocorre em esquemas iterativos de Krylov para sistemas não-hermitianos.

CAPÍTULO 4 – ALGORITMOS DE INTEGRAÇÃO

4.1. Introdução

Como visto no capítulo 2, a equação integral de contorno na forma discretizada envolve integrais cuja solução de forma analítica é praticamente impossível. Sendo assim faz-se necessário a utilização de processos numéricos de integração. Neste trabalho, adota-se o processo quadratura de Gauss–Legendre (Bathe, 1996), que é um dos mais eficientes processos de integração e mais comumente adotado.

Na formulação padrão do método dos elementos de contorno surgem, basicamente, três tipos de integrais: integrais não–singulares, *quasi*–singulares e singulares. Como avaliar tais integrais de forma eficiente em termos de precisão das respostas e tempo de processamento é um ponto fundamental em análises via M.E.C.

Neste capítulo descrevem-se aspectos gerais relacionados a essas integrais e aos processos numéricos utilizados para a avaliação das mesmas.

4.2. Integrais Não-singulares e *quasi*-singulares

As integrais não-singulares ocorrem quando o ponto fonte, ξ , não pertence ao elemento sobre o qual se integra. Nestes casos a avaliação das integrais é feita aplicando-se diretamente o processo de quadratura de Gauss sem nenhuma consideração especial (Banerjee, 1994), e a convergência ocorre através do aumento do número de pontos de integração.

No entanto, tendo em vista o aumento da eficiência da integração numérica, pode-se, após mapear os elementos no sistema de coordenadas naturais (r, s) , subdividi-los em vários subelementos de integração. A Figura 4.1 mostra esta subdivisão para o elemento de contorno quadrangular de 8 nós, mas vale ressaltar que procedimento análogo pode ser aplicado aos demais elementos.

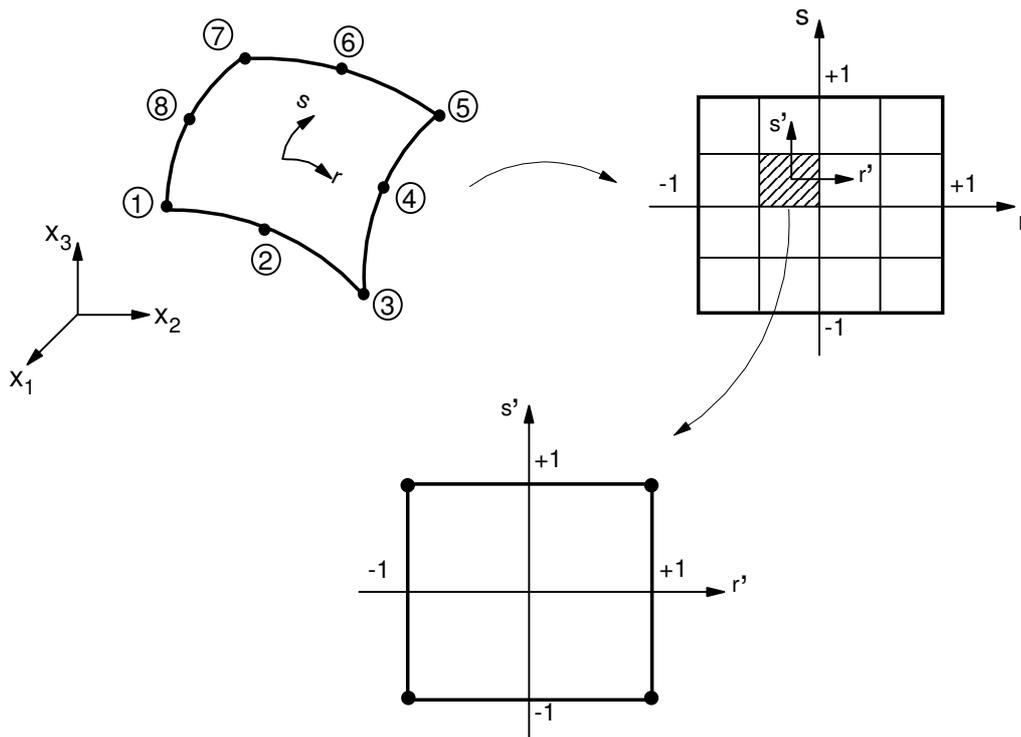


Figura 4.1 – Subelementos de integração

Portanto, tem-se as seguintes expressões para a avaliação numérica das integrais não singulares envolvidas na expressão (2.21):

$$\begin{aligned}
\int_{\Gamma_e} p^*[\mathbf{x}(r,s),\xi]h_q(r,s)d\tilde{A}[\mathbf{x}(r,s)] &= \int_{-1}^{+1} \int_{-1}^{+1} p^*(r,s)h_q(r,s)|\mathbf{J}_\Gamma^e(r,s)|drds = \\
&= \sum_{i=1}^{nse} \int_{-1}^{+1} \int_{-1}^{+1} p^*(r',s')h_q(r',s')|\mathbf{J}_\Gamma^e(r',s')||\mathbf{J}_\Gamma^{(i)}(r',s')|dr'ds' = \\
&= \sum_{i=1}^{nse} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} p^*(r'_k,s'_l)h_q(r'_k,s'_l)|\mathbf{J}_\Gamma^e(r'_k,s'_l)||\mathbf{J}_\Gamma^{(i)}(r'_k,s'_l)|w_k w_l,
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
\int_{\Gamma_e} u^*[\mathbf{x}(r,s),\xi]h_q(r,s)d\tilde{A}[\mathbf{x}(r,s)] &= \int_{-1}^{+1} \int_{-1}^{+1} u^*(r,s)h_q(r,s)|\mathbf{J}_\Gamma^e(r,s)|drds = \\
&= \sum_{i=1}^{nse} \int_{-1}^{+1} \int_{-1}^{+1} u^*(r',s')h_q(r',s')|\mathbf{J}_\Gamma^e(r',s')||\mathbf{J}_\Gamma^{(i)}(r',s')|dr'ds' = \\
&= \sum_{i=1}^{nse} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} u^*(r'_k,s'_l)h_q(r'_k,s'_l)|\mathbf{J}_\Gamma^e(r'_k,s'_l)||\mathbf{J}_\Gamma^{(i)}(r'_k,s'_l)|w_k w_l.
\end{aligned} \tag{4.2}$$

Em (4.1) e (4.2), nse é o número de subelementos considerado por elemento de contorno Γ_e ; ngr e ngs são, respectivamente, o número de pontos de integração nas direções r e s ; w_k e w_l são os pesos de integração nas direções r e s , e $|\mathbf{J}_\Gamma(r,s)|$ é o determinante do jacobiano da transformação dado por

$$\begin{aligned}
|\mathbf{J}_\Gamma(r,s)| &= \left\| \frac{\partial \mathbf{x}}{\partial r} \times \frac{\partial \mathbf{x}}{\partial s} \right\| = \left[\left(\frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} - \frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} \right)^2 + \right. \\
&\quad \left. \left(\frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} - \frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} \right)^2 + \left(\frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} - \frac{\partial x_2}{\partial r} \frac{\partial x_3}{\partial r} \right)^2 \right].
\end{aligned} \tag{4.3}$$

Note que quando a integração for realizada sobre um elemento descontínuo, as funções de forma h_q , nas expressões (4.1) e (4.2), devem ser convenientemente substituídas pelas correspondentes funções de forma do elemento de contorno descontínuo (h_q^d). Essa observação também é válida para os procedimentos especiais de integração apresentados no item 4.4.

As integrais *quasi*-singulares ocorrem quando r tende a zero, mas não torna-se nulo, como acontece tipicamente na avaliação de grandezas em pontos internos via expressões integrais de contorno, no modelamento de domínios esbeltos e também em modelos de elementos de contorno com a presença de elementos descontínuos. Nesses casos, cuidados especiais no cálculo dos coeficientes do sistema de equações algébricas (expressão 2.22) precisam ser tomados. Uma possível solução para este problema seria o aumento do número de pontos de integração da quadratura gaussiana, eventualmente com a utilização de subelementos de integração. No entanto, tal procedimento elevaria consideravelmente o custo computacional requerido para a montagem do sistema de equações com precisão aceitável. Uma alternativa mais eficiente é a utilização de processos especiais de integração baseado em transformações de coordenadas. Neste trabalho, adotam-se processos especiais de integração baseados na transformação de coordenadas polares triangulares e em transformações não-lineares, as quais serão apresentadas em seções subseqüentes (Mang *et al.*, 1985; Telles, 1987; Telles e Oliveira, 1994; Araújo, 1994; Alberto, 2002).

4.3. Integrais Singulares

Quando o ponto fonte, ξ , pertence ao elemento sobre o qual se integra, ou seja, r é igual a zero, surgem as integrais singulares ou impróprias, sendo que, estas podem ser fracamente singulares ou fortemente singulares. Tais integrais resultam do comportamento singular das soluções fundamentais quando $r \rightarrow 0$. Para o caso tridimensional tem-se singularidades da ordem $O(1/r^2)$ e $O(1/r)$ para, respectivamente, p^* e u^* (ver expressões 2.2 e 2.5).

As integrais fracamente singulares estão presentes na integração dos potenciais ou deslocamentos fundamentais (u^*). Nestes casos elas são convergentes no sentido ordinário e, portanto, pode-se aplicar a quadratura de Gauss padrão. Entretanto, a precisão da resposta, bem como, a eficiência do processo pode ser aumentada se processos especiais de integração forem adotados para tratar os núcleos singulares. Aqui também adotam-se processos especiais de integração baseados na transformação de coordenadas polares triangulares e em transformações não-lineares para a avaliação direta de integrais fracamente singulares.

Todavia esses processos não se aplicam no cálculo direto de integrais fortemente singulares, existentes apenas no sentido do valor principal de Cauchy. Tais integrais são aquelas associadas aos fluxos ou forças fundamentais (p^*). Um modo de se evitar a avaliação direta dessas integrais é aplicando-se o critério de potencial constante para problemas estacionários de potencial ou de deslocamento de corpo rígido para problemas de elasticidade. No primeiro caso tem-se que na ausência de forças de volume, para uma distribuição constante de potencial em $\Omega \cup \Gamma$, os fluxos correspondentes em Γ são identicamente nulos. Deste modo obtém-se

$$\mathbf{c}(\xi) + \mathbf{H}_{\xi\xi} = \begin{cases} -\sum_{\substack{k=1 \\ k \neq \hat{i}}}^N \mathbf{H}_{\xi_k}, & \text{para regiões finitas} \\ 1 - \sum_{\substack{k=1 \\ k \neq \hat{i}}}^N \mathbf{H}_{\xi_k}, & \text{para regiões infinitas} \end{cases}, \quad (4.4)$$

sendo todos os coeficientes \mathbf{H}_{ξ_k} no lado direito da equação (4.4) obtidos de integrais não-singulares ou fracamente singulares. Assim, a soma do termo livre $\mathbf{c}(\xi)$ e do termo $\mathbf{H}_{\xi\xi}$ pode ser facilmente determinada. Expressões análogas são obtidas para problemas de elasticidade, contudo em potencial tem-se que o termo livre refere-se a um bloco de matriz de ordem 1, e em elasticidade, a um bloco de ordem 3.

4.4. Procedimentos Especiais de Integração

Objetivando-se aumentar a precisão dos resultados, bem como a eficiência do algoritmo de integração, introduziram-se no programa alguns procedimentos especiais. Estes serão de grande importância principalmente na avaliação das integrais *quasi*-singulares e singulares.

A escolha do processo de integração é feita a partir da variável *icpt*, a qual poderá assumir os seguintes valores:

$$icpt = \begin{cases} 0 \rightarrow \text{quadratura de Gauss padrão} \\ 1 \rightarrow \text{transformação de coordenadas polares triangulares} \\ 2 \rightarrow \text{transformação não-linear} \\ 3 \rightarrow \text{processo combinado} \end{cases}$$

Durante a integração sobre um determinado elemento os processos especiais ($icpt = 1, 2$ ou 3) poderão ser ativados ou desativados dependendo do valor da variável r_{icpt} . Esta variável fornece um intervalo dentro do qual os processos especiais para avaliação de integrais *quasi*-singulares ou singulares devem ser ativados. Nos próximos itens descreve-se, de forma sucinta, estes procedimentos.

4.4.1. Transformação de Coordenadas Polares Triangulares

Este procedimento caracteriza-se por uma seqüência de mapeamentos de coordenadas com o objetivo de reduzir a ordem da singularidade das integrais. Inicialmente o elemento de contorno já mapeado no espaço de coordenadas naturais (r, s) é subdividido em dois ou três subdomínios triangulares, dependendo, respectivamente, se o ponto singular está situado no canto ou no meio da aresta do elemento (ver Figura 4.2 (a)). Em seguida, cada triângulo é mapeado em um quadrado de lado unitário utilizando-se a expressão

$$r_i(\zeta_1, \zeta_2) = \sum_{q=1}^3 h_q^* \hat{r}_{iq}, \quad (4.5)$$

em que, r_i são as coordenadas naturais (r, s) dos pontos nodais do triângulo com, $i = 1, 2$, $r_1 = r$ e $r_2 = s$; \hat{r}_{iq} é a componente i do vetor de coordenadas do q -ésimo nó do elemento. ζ_1 e ζ_2 são as coordenadas polares triangulares, e h_q^* são as funções de forma associadas à transformação, dadas por:

$$\begin{aligned} h_1^*(\zeta_1, \zeta_2) &= (1 - \zeta_1) \\ h_2^*(\zeta_1, \zeta_2) &= \zeta_1(1 - \zeta_2) \\ h_3^*(\zeta_1, \zeta_2) &= \zeta_1\zeta_2 \end{aligned} \quad (4.6)$$

Finalmente, este quadrado unitário é novamente mapeado em um espaço de coordenadas naturais (η_1, η_2) , com $-1 \leq \eta_1, \eta_2 \leq +1$ (ver Figura 4.2 (c), (d) e (e)), permitindo-se deste modo que o processo de quadratura de Gauss padrão seja diretamente aplicado (Alberto, 2002).

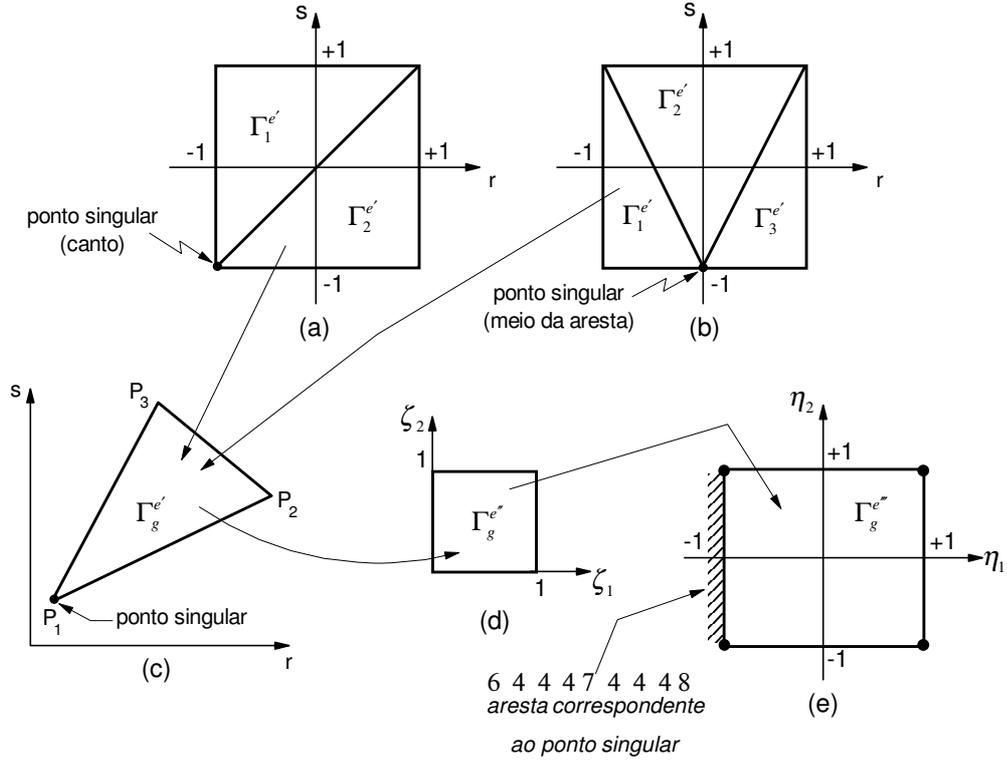


Figura 4.2 – Seqüência de mapeamentos – transformação de coordenadas polares triangulares

Sendo assim as integrais em (2.21) podem ser avaliadas a partir das seguintes expressões:

$$\begin{aligned}
 \int_{\Gamma_e} p^*[\mathbf{x}(r, s), \xi] h_q(r, s) d\tilde{\mathbf{A}}[\mathbf{x}(r, s)] &= \int_{-1}^{+1} \int_{-1}^{+1} p^*(r, s) h_q(r, s) |\mathbf{J}_\Gamma^e(r, s)| dr ds = \\
 &= \sum_{g=1}^{nsdi} \int_0^{+1+1} \int_0^{+1+1} p^*(\zeta_1, \zeta_2) h_q(\zeta_1, \zeta_2) |\mathbf{J}_\Gamma^e(\zeta_1, \zeta_2)| |\mathbf{J}_\Gamma(\tilde{\mathbf{A}}_g^{e'})| d\zeta_1 d\zeta_2 = \\
 &= \sum_{g=1}^{nsdi} \int_{-1}^{+1+1} \int_{-1}^{+1+1} p^*(\eta_1, \eta_2) h_q(\eta_1, \eta_2) |\mathbf{J}_\Gamma^e(\eta_1, \eta_2)| |\mathbf{J}_\Gamma(\tilde{\mathbf{A}}_g^{e'})| |\mathbf{J}_\Gamma(\tilde{\mathbf{A}}_g^{e''})| d\eta_1 d\eta_2 = \\
 &= \sum_{g=1}^{nsdi} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} p^*(\eta_{1k}, \eta_{2l}) h_q(\eta_{1k}, \eta_{2l}) |\mathbf{J}_\Gamma^e(\eta_{1k}, \eta_{2l})| |\mathbf{J}_\Gamma(\Gamma_g^{e'})| |\mathbf{J}_\Gamma(\Gamma_g^{e''})| w_k w_l \quad \mathbf{e},
 \end{aligned} \tag{4.7}$$

$$\begin{aligned}
& \int_{\Gamma_e} u^*[\mathbf{x}(r,s), \xi] h_q(r,s) d\tilde{A}[\mathbf{x}(r,s)] = \int_{-1}^{+1} \int_{-1}^{+1} u^*(r,s) h_q(r,s) |\mathbf{J}_{\Gamma}^e(r,s)| dr ds = \\
& = \sum_{g=1}^{nsdi} \int_0^{+1} \int_0^{+1} u^*(\zeta_1, \zeta_2) h_q(\zeta_1, \zeta_2) |\mathbf{J}_{\Gamma}^e(\zeta_1, \zeta_2)| |\mathbf{J}_{\Gamma}^{(i)}(\tilde{A}_g^{e'})| d\zeta_1 d\zeta_2 = \\
& = \sum_{g=1}^{nsdi} \int_{-1}^{+1} \int_{-1}^{+1} u^*(\eta_1, \eta_2) h_q(\eta_1, \eta_2) |\mathbf{J}_{\Gamma}^e(\eta_1, \eta_2)| |\mathbf{J}_{\Gamma}(\tilde{A}_g^{e'})| |\mathbf{J}_{\Gamma}(\tilde{A}_g^{e'})| d\eta_1 d\eta_2 = \\
& = \sum_{g=1}^{nsdi} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} u^*(\eta_{1k}, \eta_{2l}) h_q(\eta_{1k}, \eta_{2l}) |\mathbf{J}_{\Gamma}^e(\eta_{1k}, \eta_{2l})| |\mathbf{J}_{\Gamma}(\Gamma_g^{e'})| |\mathbf{J}_{\Gamma}(\Gamma_g^{e'})| w_k w_l,
\end{aligned} \tag{4.8}$$

onde, $nsdi$ é o número de subdomínios triangulares (igual a 2 ou 3), e o jacobiano envolvido nas várias transformações de coordenadas é dado por:

$$|\mathbf{J}_{\Gamma}(\Gamma_g^{e'})| |\mathbf{J}_{\Gamma}(\Gamma_g^{e'})| = \frac{1}{4} (1 + \eta_1) A_{\Gamma_g}. \tag{4.9}$$

Em (4.9), A_{Γ_g} é a área do subdomínio triangular, a qual poderá assumir os valores 1 ou 2 conforme a subdivisão realizada.

Menciona-se que após o mapeamento no espaço de coordenadas (η_1, η_2) , pode-se ainda, conforme a necessidade, subdividir os elementos em subelementos de integração como apresentado no item 4.2.

Considerando-se, por exemplo, uma integração de ordem 4x4, obtém-se aplicando-se a transformação de coordenadas polares triangulares, para o ponto singular coincidente com os nós 1 e 2 do elemento quadrangular de 8 nós, as seguintes posições dos pontos de integração:

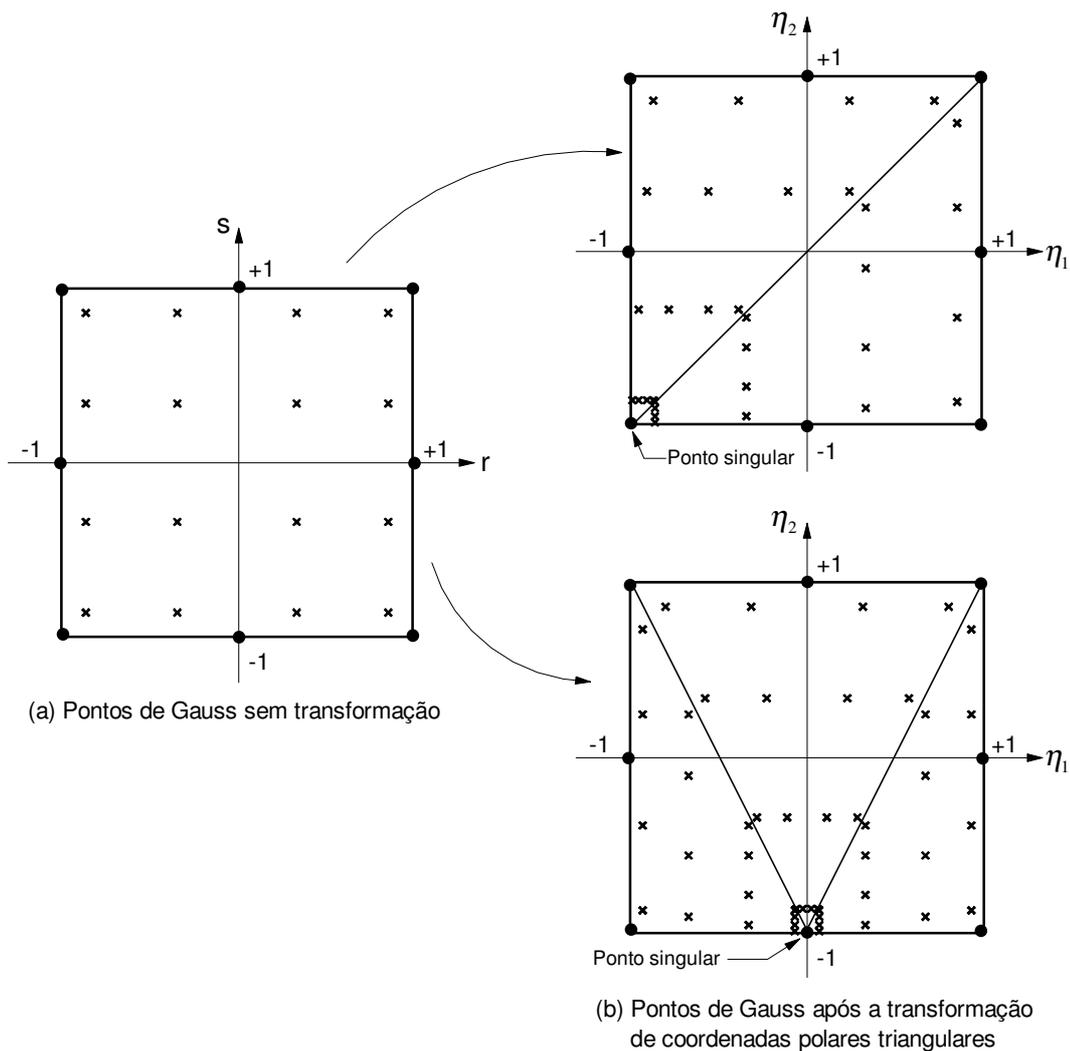


Figura 4.3 – Posição dos pontos de integração após a transformação de coordenadas polares triangulares

4.4.2. Transformação Cúbica

Adota-se neste trabalho a transformação não-linear desenvolvida por Telles (1987), a qual baseia-se em uma transformação polinomial do 3º grau e visa o aumento da eficiência do processo de quadratura de Gauss na avaliação de integrais singulares ou *quasi*-singulares através do deslocamento dos pontos de integração para as proximidades do ponto singular (Telles, 1987; Telles e Oliveira, 1994).

Para a derivação desta transformação, considere a integral da função $f(\eta)$, singular em $\bar{\eta}$, bem como o polinômio do 3º grau apresentados nas expressões (4.10) e (4.11):

$$I = \int_{-1}^{+1} f(\eta) d\eta \quad \text{e} \quad (4.10)$$

$$\eta(\gamma) = a\gamma^3 + b\gamma^2 + c\gamma + d \quad , \quad (4.11)$$

A determinação dos coeficientes a , b , c e d do polinômio é feita impondo-se as seguintes condições:

$$\begin{cases} \left. \frac{d\eta}{d\gamma} \right|_{\bar{\eta}} = 0 \\ \left. \frac{d^2\eta}{d\gamma^2} \right|_{\bar{\eta}} = 0 \\ \eta(1) = 1 \\ \eta(-1) = -1 \end{cases} \quad . \quad (4.12)$$

Sendo assim, obtém-se:

$$\begin{aligned} a &= 1/Q, \\ b &= -3\bar{\gamma}/Q, \\ c &= 3\bar{\gamma}^2/Q \quad \text{e} \\ d &= -b, \end{aligned} \quad (4.13)$$

em que $Q = 1 + 3\bar{\gamma}^2$ e $\bar{\gamma}$ é o valor de γ para o qual tem-se $\eta(\bar{\gamma}) = \bar{\eta}$, o qual pode ser calculado a partir da expressão

$$\bar{\gamma} = \sqrt[3]{(\bar{\eta}\eta^* + |\eta^*|)} + \sqrt[3]{(\bar{\eta}\eta^* - |\eta^*|)} + \bar{\eta} \quad , \quad \text{com } \eta^* = \bar{\eta}^2 - 1. \quad (4.14)$$

Portanto, a expressão (4.10) pode ser reescrita da seguinte forma,

$$I = \int_{-1}^{+1} f \left[\frac{(\gamma - \bar{\gamma})^3 + \bar{\gamma}(\bar{\gamma}^2 + 3)}{(1 + 3\bar{\gamma}^2)} \right] \frac{(\gamma - \bar{\gamma})^2}{(1 + 3\bar{\gamma}^2)} d\gamma. \quad (4.15)$$

No algoritmo de integração desenvolvido tem-se que a função $f(\eta)$ que se deseja integrar é igual aos núcleos fundamentais p^* e u^* , e γ representa a posição original dos pontos de integração, ou seja, as coordenadas (r, s) dos pontos de integração fornecidas na quadratura de Gauss padrão. As novas posições destes pontos são obtidas mediante a aplicação da expressão (4.11) em função da posição do ponto singular.

Portanto, as integrais em (2.21) são agora avaliadas a partir das expressões abaixo:

$$\begin{aligned}
\int_{\Gamma_e} p^*[\mathbf{x}(r, s), \xi] h_q(r, s) d\tilde{A}[\mathbf{x}(r, s)] &= \int_{-1}^{+1} \int_{-1}^{+1} p^*(r, s) h_q(r, s) |\mathbf{J}_\Gamma^e(r, s)| dr ds = \\
&= \int_{-1}^{+1} \int_{-1}^{+1} p^*(r(\gamma), s(\gamma)) h_q(r(\gamma), s(\gamma)) |\mathbf{J}_\Gamma^e(r(\gamma), s(\gamma))| |\mathbf{J}_{nl}(\gamma)| dr(\gamma) ds(\gamma) = \\
&= \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} p^*(r(\gamma_k), s(\gamma_l)) h_q(r(\gamma_k), s(\gamma_l)) |\mathbf{J}_\Gamma^e(r(\gamma_k), s(\gamma_l))| |\mathbf{J}_{nl}(\gamma_k)| |\mathbf{J}_{nl}(\gamma_l)| w_k w_l,
\end{aligned} \tag{4.16}$$

$$\begin{aligned}
\int_{\Gamma_e} u^*[\mathbf{x}(r, s), \xi] h_q(r, s) d\tilde{A}[\mathbf{x}(r, s)] &= \int_{-1}^{+1} \int_{-1}^{+1} u^*(r, s) h_q(r, s) |\mathbf{J}_\Gamma^e(r, s)| dr ds = \\
&= \int_{-1}^{+1} \int_{-1}^{+1} u^*(r(\gamma), s(\gamma)) h_q(r(\gamma), s(\gamma)) |\mathbf{J}_\Gamma^e(r(\gamma), s(\gamma))| |\mathbf{J}_{nl}(\gamma)| dr(\gamma) ds(\gamma) = \\
&= \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} u^*(r(\gamma_k), s(\gamma_l)) h_q(r(\gamma_k), s(\gamma_l)) |\mathbf{J}_\Gamma^e(r(\gamma_k), s(\gamma_l))| |\mathbf{J}_{nl}(\gamma_k)| |\mathbf{J}_{nl}(\gamma_l)| w_k w_l.
\end{aligned} \tag{4.17}$$

Em (4.16) e (4.17) $|\mathbf{J}_{nl}(\gamma)|$ é o determinante do jacobiano da transformação não-linear, o qual é dado por:

$$|\mathbf{J}_{nl}(\gamma)| = 3 \frac{(\gamma - \bar{\gamma})^2}{(1 + 3\bar{\gamma}^2)}. \tag{4.18}$$

Esta transformação, de maneira análoga à transformação apresentada anteriormente, também pode ser aplicada associada à estratégia de subdivisão dos elementos de integração.

Considerando-se novamente uma integração de ordem 4x4 sobre um elemento quadrangular de 8 nós obtém-se, após a aplicação da transformação não-linear (transformação de Telles), as posições para os pontos de integração apresentadas na Figura 4.4.

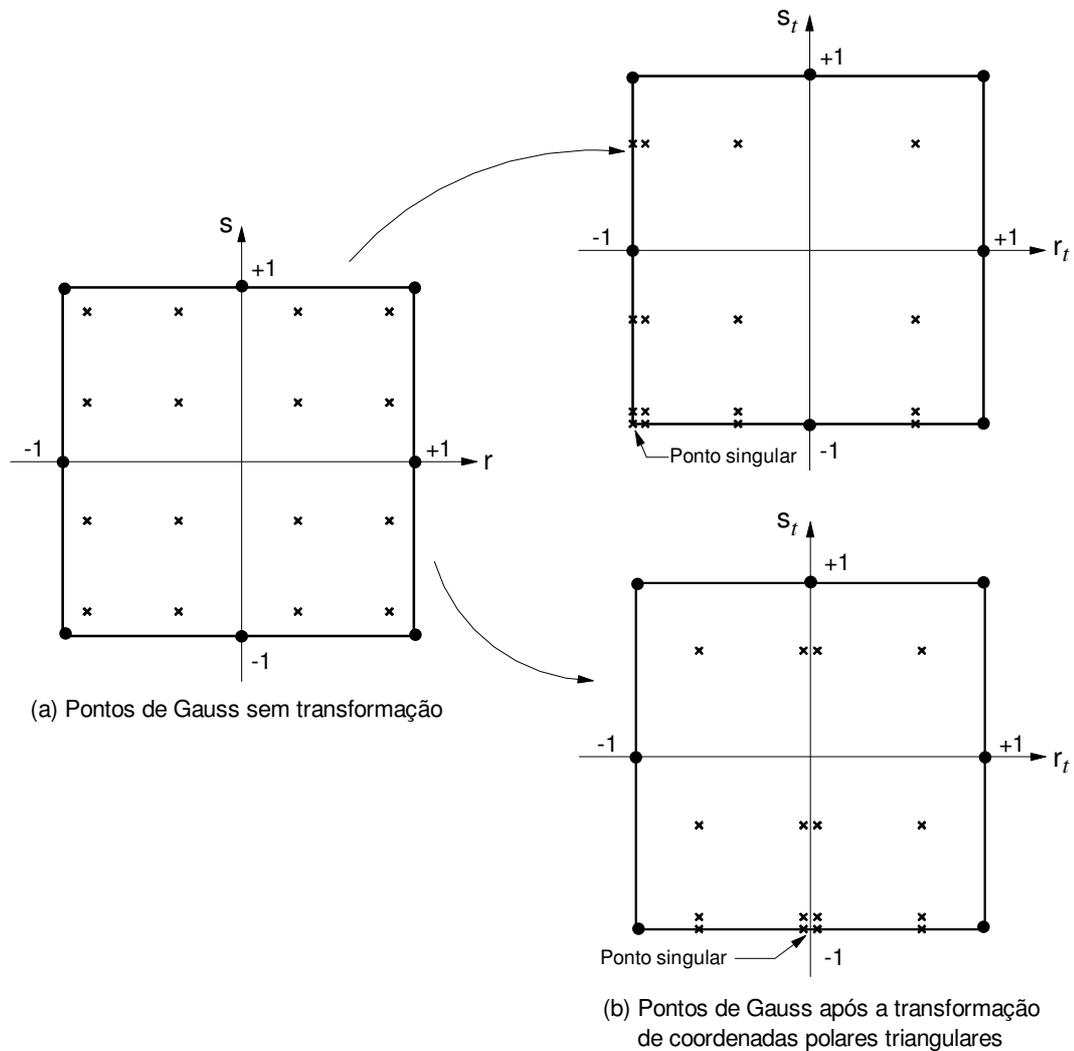


Figura 4.4 – Posição dos pontos de integração após a transformação não-linear

4.4.3. Processo Combinado – Transformação de Coordenadas Polares Triangulares Juntamente com a Transformação Cúbica

Este processo de integração baseia-se na combinação da transformação de coordenadas polares triangulares e da transformação de Telles. Inicialmente realiza-se a seqüência de mapeamentos referentes à transformação de coordenadas polares

triangulares definindo-se a posição dos pontos de integração no domínio $-1 \leq \eta_1, \eta_2 \leq +1$ (ver Figura 4.2). Após estes mapeamentos aplica-se a transformação não-linear de coordenadas segundo η_1 , de modo a deslocar os pontos de integração na direção do ponto singular. Deste modo, resulta para as integrais em (2.21):

$$\begin{aligned}
& \int_{\Gamma_e} p^*[\mathbf{x}(r,s), \xi] h_q(r,s) d\tilde{A}[\mathbf{x}(r,s)] = \int_{-1}^{+1} \int_{-1}^{+1} p^*(r,s) h_q(r,s) |\mathbf{J}_\Gamma^e(r,s)| dr ds = \\
& = \sum_{g=1}^{nsdi} \int_0^{+1} \int_0^{+1} p^*(\zeta_1, \zeta_2) h_q(\zeta_1, \zeta_2) |\mathbf{J}_\Gamma^e(\zeta_1, \zeta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| d\zeta_1 d\zeta_2 = \\
& = \sum_{g=1}^{nsdi} \int_{-1}^{+1} \int_{-1}^{+1} p^*(\eta_1, \eta_2) h_q(\eta_1, \eta_2) |\mathbf{J}_\Gamma^e(\eta_1, \eta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| |\mathbf{J}_\Gamma(\tilde{A}_g^{e''})| d\eta_1 d\eta_2 = \\
& = \sum_{g=1}^{nsdi} \int_{-1}^{+1} \int_{-1}^{+1} \left\{ p^*(\eta_1(\gamma), \eta_2) h_q(\eta_1(\gamma), \eta_2) |\mathbf{J}_\Gamma^e(\eta_1(\gamma), \eta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| \right. \\
& \quad \left. |\mathbf{J}_\Gamma(\tilde{A}_g^{e''})| |\mathbf{J}_{nl}(\gamma)| d\eta_1(\gamma) d\eta_2 \right\} = \\
& = \sum_{g=1}^{nsdi} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} \left\{ p^*(\eta_1(\gamma_k), \eta_{2l}) h_q(\eta_1(\gamma_k), \eta_{2l}) |\mathbf{J}_\Gamma^e(\eta_1(\gamma_k), \eta_{2l})| \right. \\
& \quad \left. |\mathbf{J}_\Gamma(\Gamma_g^{e'})| |\mathbf{J}_\Gamma(\Gamma_g^{e''})| |\mathbf{J}_{nl}(\gamma_k)| w_k w_l \right\} , e
\end{aligned} \tag{4.19}$$

$$\begin{aligned}
& \int_{\Gamma_e} u^*[\mathbf{x}(r,s), \xi] h_q(r,s) d\tilde{A}[\mathbf{x}(r,s)] = \int_{-1}^{+1} \int_{-1}^{+1} u^*(r,s) h_q(r,s) |\mathbf{J}_\Gamma^e(r,s)| dr ds = \\
& = \sum_{g=1}^{nsdi} \int_0^{+1} \int_0^{+1} u^*(\zeta_1, \zeta_2) h_q(\zeta_1, \zeta_2) |\mathbf{J}_\Gamma^e(\zeta_1, \zeta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| d\zeta_1 d\zeta_2 = \\
& = \sum_{g=1}^{nsdi} \int_{-1}^{+1} \int_{-1}^{+1} u^*(\eta_1, \eta_2) h_q(\eta_1, \eta_2) |\mathbf{J}_\Gamma^e(\eta_1, \eta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| |\mathbf{J}_\Gamma(\tilde{A}_g^{e''})| d\eta_1 d\eta_2 = \\
& = \sum_{g=1}^{nsdi} \int_{-1}^{+1} \int_{-1}^{+1} \left\{ u^*(\eta_1(\gamma), \eta_2) h_q(\eta_1(\gamma), \eta_2) |\mathbf{J}_\Gamma^e(\eta_1(\gamma), \eta_2)| |\mathbf{J}_\Gamma(\tilde{A}_g^{e'})| \right. \\
& \quad \left. |\mathbf{J}_\Gamma(\tilde{A}_g^{e''})| |\mathbf{J}_{nl}(\gamma)| d\eta_1(\gamma) d\eta_2 \right\} = \\
& = \sum_{g=1}^{nsdi} \sum_{k=1}^{ngr} \sum_{l=1}^{ngs} \left\{ u^*(\eta_1(\gamma_k), \eta_{2l}) h_q(\eta_1(\gamma_k), \eta_{2l}) |\mathbf{J}_\Gamma^e(\eta_1(\gamma_k), \eta_{2l})| \right. \\
& \quad \left. |\mathbf{J}_\Gamma(\Gamma_g^{e'})| |\mathbf{J}_\Gamma(\Gamma_g^{e''})| |\mathbf{J}_{nl}(\gamma_k)| w_k w_l \right\} .
\end{aligned} \tag{4.20}$$

Observa-se que não faz sentido aplicar a transformação não-linear segundo a direção η_2 , uma vez que não há singularidade nesta direção (ver Figura 4.2).

Como nos casos anteriores esta estratégia pode ser aplicada com ou sem subelementos de integração. Na figura a seguir, apresenta-se a posição dos pontos de integração (ordem 4x4) após a aplicação do processo combinado.

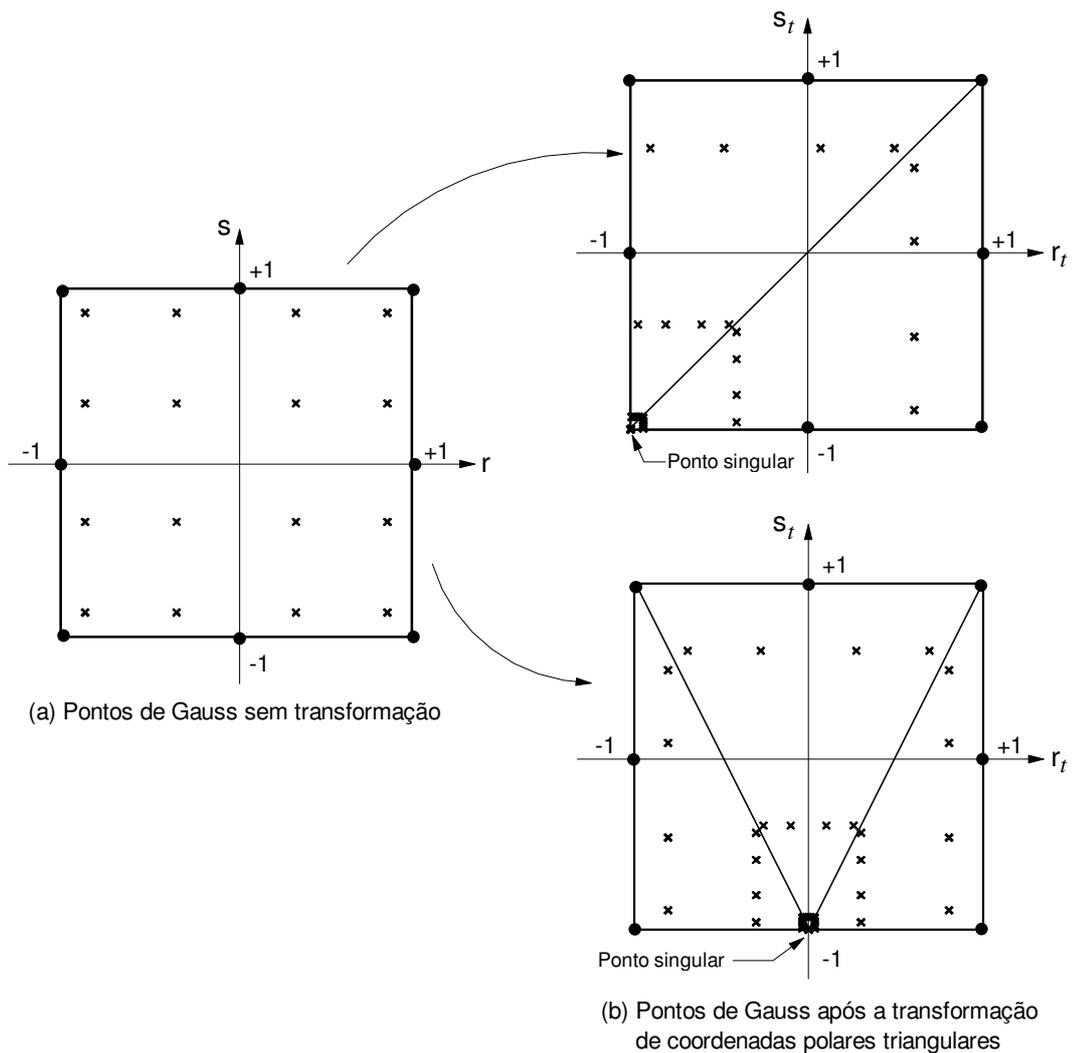


Figura 4.5 – Posição dos pontos de integração após a transformação combinada

4.5. Estudos de Casos

Para validar os algoritmos de integração introduzidos no programa, bem como realizar um estudo da eficiência dos mesmos (performance), apresenta-se nesta seção a integração das funções teste a seguir:

– Função I:
$$\int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{\sqrt{(1.004 - \eta_1)^2 + (1.004 - \eta_2)^2}} d\eta_1 d\eta_2 \Rightarrow PS(1.004, 1.004) ;$$

– Função II:
$$\int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{\sqrt{(\eta_1 + 1)^2 + (\eta_2 + 1)^2}} d\eta_1 d\eta_2 \Rightarrow PS(-1, -1) ;$$

– Função III:
$$\int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{\sqrt{(\eta_1)^2 + (\eta_2 + 1)^2}} d\eta_1 d\eta_2 \Rightarrow PS(0, -1) ;$$

– Função IV:
$$\int_{-1}^{+1} \int_{-1}^{+1} \frac{1}{\sqrt{(1.0001 - \eta_1)^2 + (1.0001 - \eta_2)^2}} d\eta_1 d\eta_2 \Rightarrow PS(1.0001, 1.0001) .$$

A função I também foi analisada por Telles (1987); através das outras funções procura-se fazer um estudo da resposta da integração quando se tem o ponto singular (*PS*) coincidente com um nó de canto (funções II e IV) e um nó de meio de aresta (função III) do elemento sobre o qual se integra. Nos gráficos abaixo apresentam-se os resultados obtidos utilizando-se os processos descritos anteriormente.

PS = (1.004, 1.004)

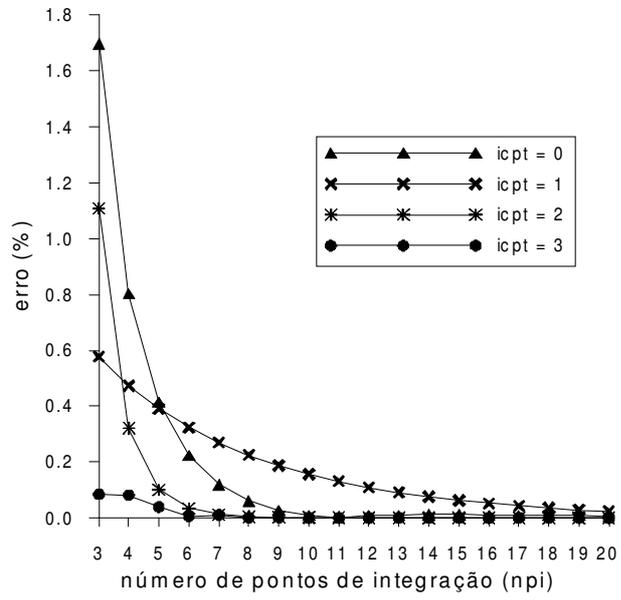


Figura 4.6 – Erro referente à integração da função I

PS = (-1, -1)

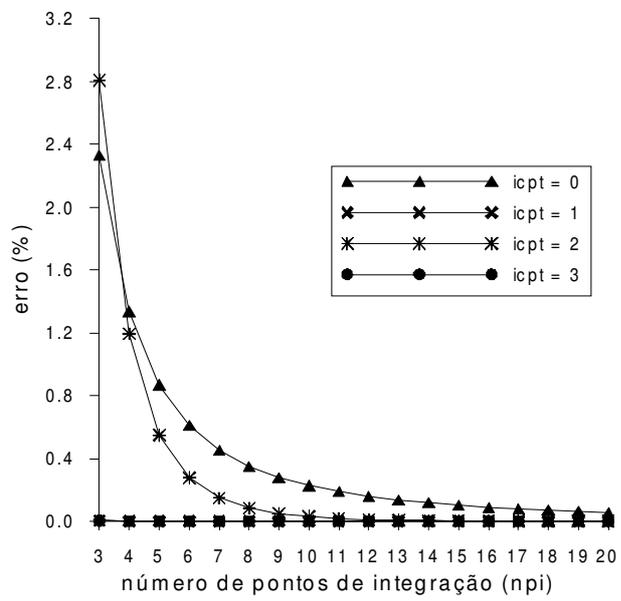


Figura 4.7 – Erro referente à integração da função II

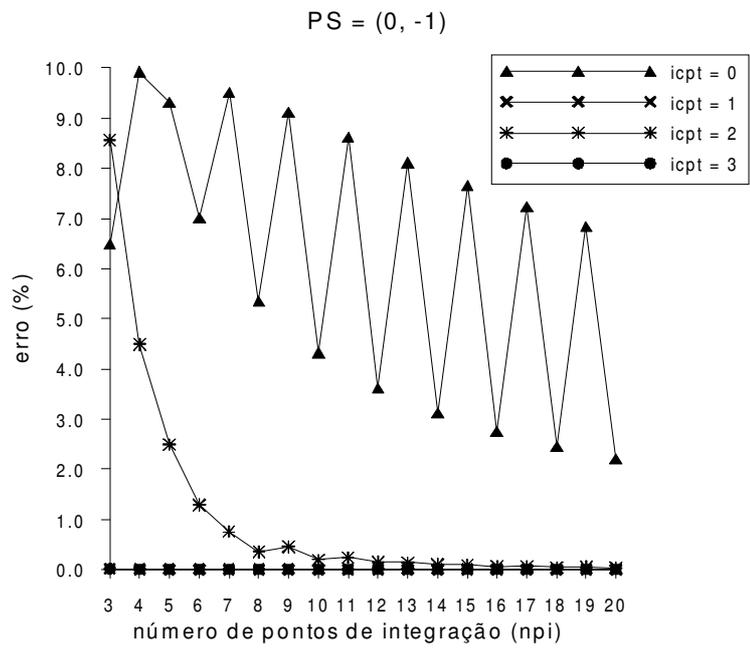


Figura 4.8 – Erro referente à integração da função III

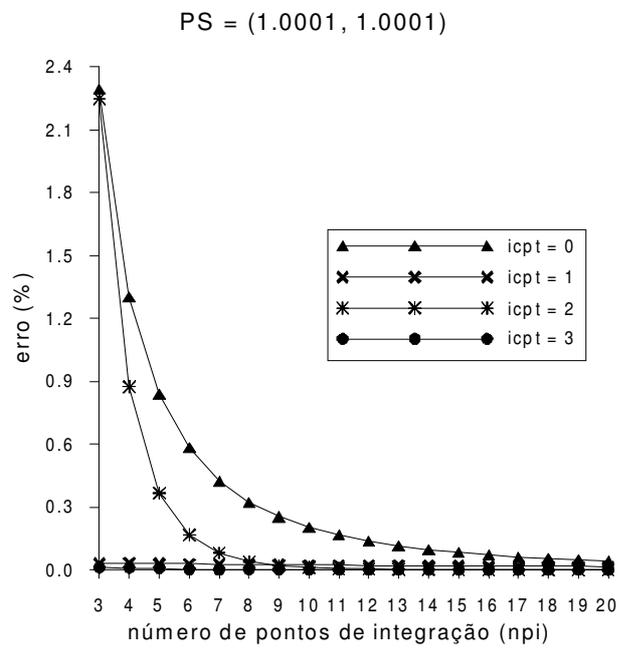


Figura 4.9 – Erro referente à integração da função IV

Menciona-se que calcularam-se os erros em relação à solução analítica das integrais, e que, npi denota o número de pontos de integração por direção, ou seja, se $npi = 3$ significa dizer que uma integração de ordem 3×3 foi adotada.

A partir dos gráficos acima observa-se que os processos de integração 1 e 3 mostraram-se bastante eficientes. Estes, em geral, apresentaram erros muito satisfatórios mesmo para ordens de integração baixas ($npi = 3$). Apenas na avaliação da função I nota-se uma pequena desvantagem do processo $icpt = 1$ em relação aos demais procedimentos. Para $PS = (1.004, 1.004)$ e $npi \geq 6$ nota-se um melhor desempenho do processo de integração baseado na transformação de Telles ($icpt = 2$). Todavia, pode-se observar que para ordens de integração menores os erros referentes a este processo aumentam rapidamente. De modo geral, o processo 3 apresenta o melhor desempenho.

Realizou-se, também, um estudo do desempenho dos processos descritos acima na avaliação dos termos $h(3,3)$ e $g(3,3)$. Para tal, considerou-se o domínio esbelto discretizado utilizando-se o elemento de oito nós apresentado na Figura 4.10.

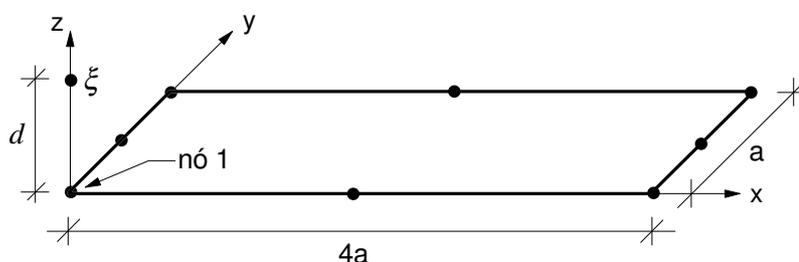


Figura 4.10 – Domínio de integração

As integrais foram avaliadas para diferentes relações de d/a , sendo d a distância do ponto fonte ao mais próximo nó do elemento, o qual corresponde ao nó 1.

Os gráficos a seguir apresentam os resultados, sendo que os erros da integração foram calculados em relação à resposta obtida para uma integração de ordem 20×20 e $nse = 6$, em função do número de pontos de integração para diferentes valores de d .

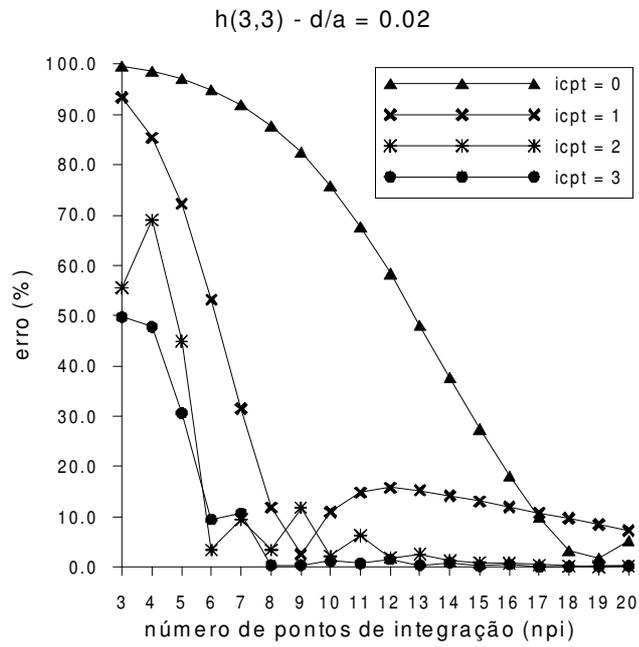


Figura 4.11 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.02$

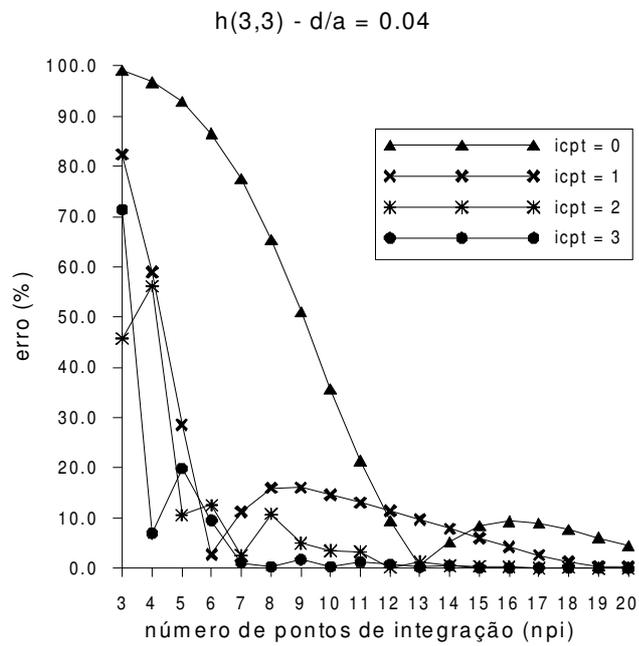


Figura 4.12 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.04$

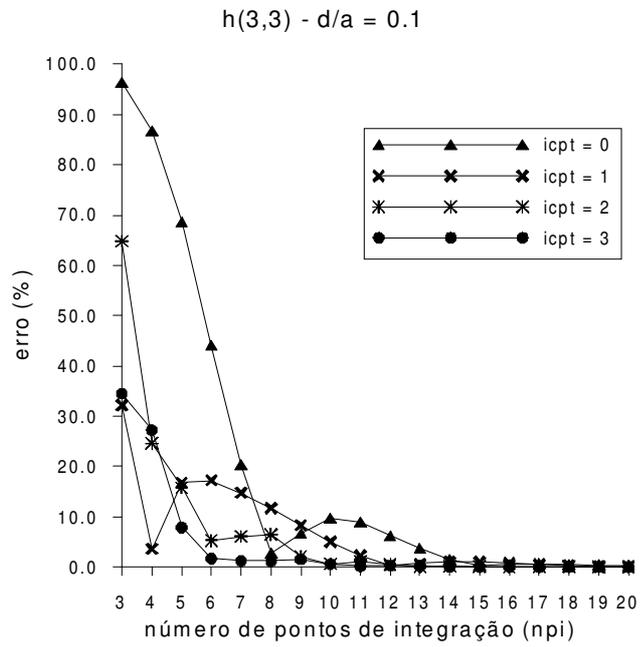


Figura 4.13 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.1$

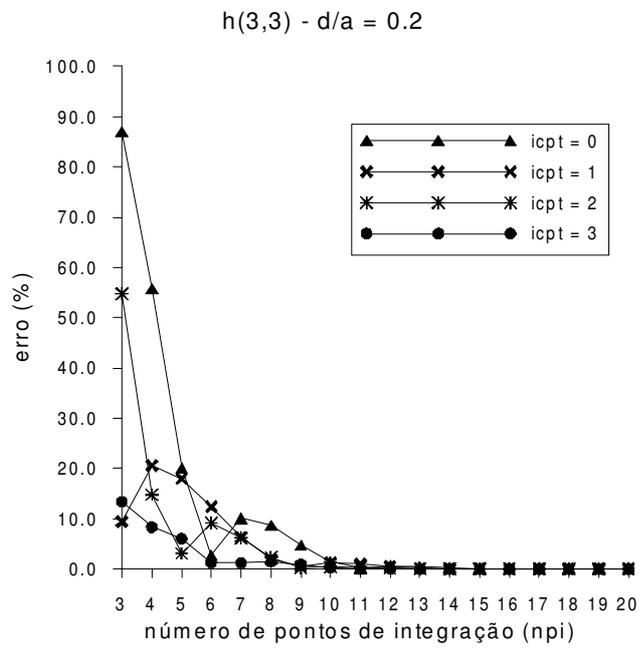


Figura 4.14 – Erro na avaliação do termo $h(3,3)$ para $d/a = 0.2$

Observa-se a partir dos gráficos acima, que os processos de integração 2 e 3 apresentaram melhor desempenho na avaliação da matriz \mathbf{H} , a qual envolve integrais fortemente singulares. Para $icpt = 3$ e $d/a = 0,02$ ou $0,04$ obtiveram-se resultados precisos para $npi \geq 7$, por outro lado, considerando-se $icpt = 2$ deve-se ter $npi \geq 8$ e $npi \geq 7$ para, respectivamente, $d/a = 0,02$ e $d/a = 0,04$. Para estes valores de d/a há muito pouca diferença entre as ordens de integração utilizando-se um ou outro processo, contudo para o processo 2 observa-se uma oscilação na resposta, a qual se estabiliza apenas para $npi \geq 12$. Aumentando-se a distância d ($d/a = 0,1$ e $0,2$) e considerando-se $icpt = 3$ obtiveram-se resultados precisos para $npi \geq 6$; para o $icpt = 2$, erros da mesma ordem de grandeza aos obtidos utilizando-se o processo $icpt = 3$ foram alcançados para $npi = 9$ e $npi = 8$ para, respectivamente, $d/a = 0,1$ e $d/a = 0,2$. Nota-se também que para esses valores de d o processo 2 apresentou muito pouca oscilação. Os processos 0 e 1 de modo geral apresentaram desempenho pior que os processos 2 e 3, sendo que o processo $icpt = 3$ se mostrou o mais eficiente.

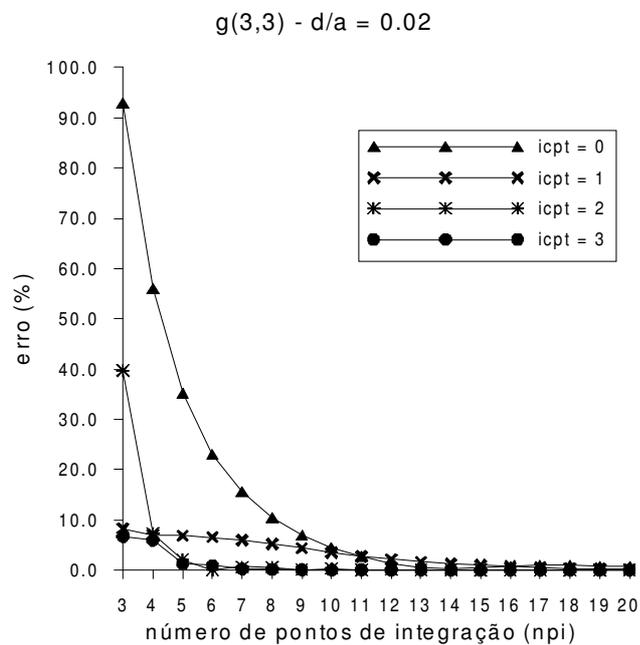


Figura 4.15 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.02$

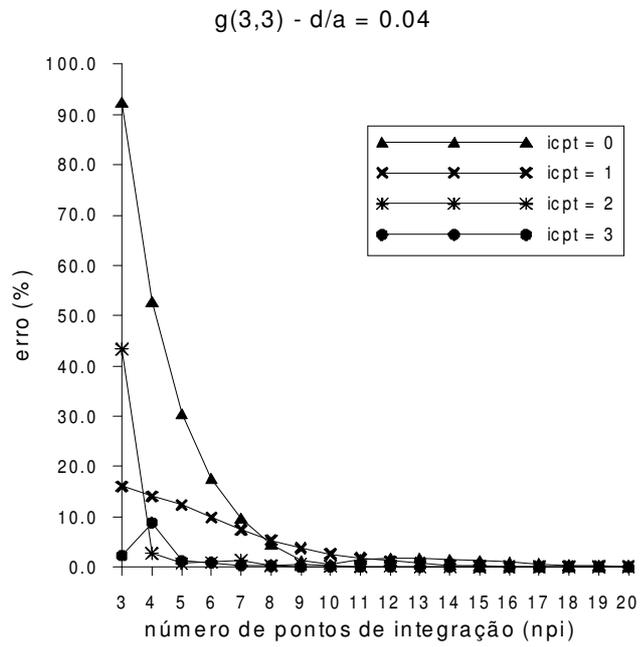


Figura 4.16 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.04$

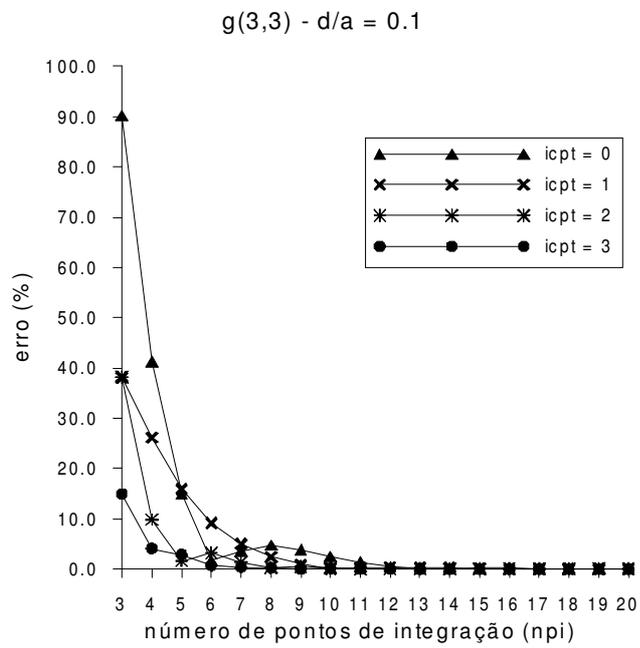


Figura 4.17 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.1$

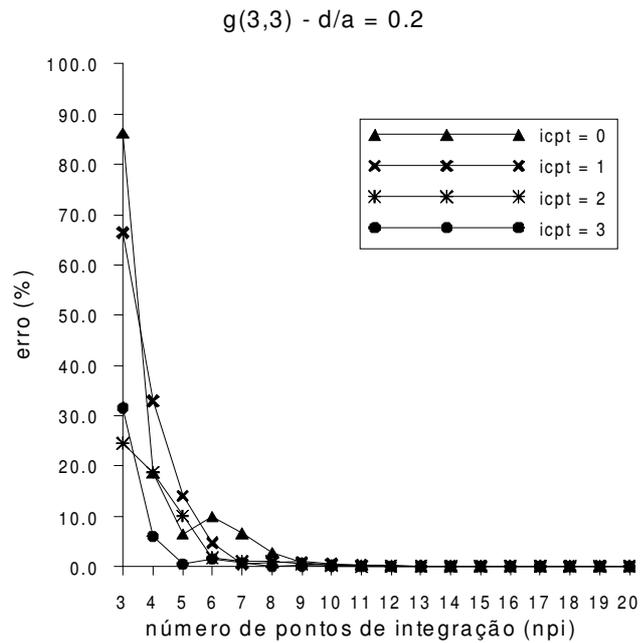


Figura 4.18 – Erro na avaliação do termo $g(3,3)$ para $d/a = 0.2$

A avaliação da matriz \mathbf{G} envolve o cálculo de integrais fracamente singulares. Para $d/a = 0.02$ e 0.04 tem-se que os processos de integração 2 e 3 apresentaram melhor desempenho que os outros processos, sendo o número mínimo de pontos de integração, em ambos os casos, igual a 5. Com o aumento valor de d observa-se uma melhora dos processos de integração 0 e 1. Bons resultados são obtidos para $npg \geq 6$ e $npg \geq 7$ para, respectivamente, $icpt = 2,3$ e $icpt = 1$, considerando-se $d/a = 0,1$ e $d/a = 0,2$. Novamente tem-se que o processo 3 mostrou-se mais eficiente; com erros da ordem de 5% mesmo para ordens de integração relativamente baixas ($npg \leq 5$).

Pode-se dizer que o processo $icpt = 0$ é inaceitável no que diz respeito ao tempo de processamento, uma vez que este demanda muitos pontos de integração para que resultados precisos sejam obtidos.

4.6. Elementos Esbeltos

Domínios esbeltos são definidos por duas superfícies principais localizadas a uma distância muito pequena uma da outra. No caso do programa computacional desenvolvido, as malhas de elementos de contorno em cada uma destas superfícies são geometricamente iguais, ou seja, cada nó de uma superfície está conectado a um nó correspondente da outra superfície por um vetor de módulo e normal à superfície mediana do domínio, sendo e igual à espessura do domínio esbelto.

A consideração destes tipos de domínios na análise implica necessariamente na resolução de integrais *quasi*-singulares devido à proximidade das superfícies principais. Sendo assim, introduziu-se no programa a rotina *intvar_nears*. Esta rotina responsabiliza-se pela geração das variáveis *inears* e *ising*, as quais são utilizadas para identificação de elementos *quasi*-singulares e singulares, respectivamente, em relação a um certo ponto fonte. Uma vez identificados os elementos *quasi*-singulares, os processos especiais de integração ($icpt = 1, 2$ e 3) são então ativados.

Também como consequência da proximidade entre nós da malha, os sistemas algébricos resultantes tornam-se *quasi*-singulares, portanto, mal-condicionados. Este fato em geral dificulta a convergência do *solver* iterativo ou torna a análise de tais sistemas via *solvers* diretos não confiável.

CAPÍTULO 5 – APLICAÇÕES

5.1. Introdução

Neste capítulo apresentam-se vários exemplos através dos quais buscou-se validar a estratégia de acoplamento e realizar um estudo da eficiência de estratégias implementadas no âmbito do programa computacional NAESY.

Comparações são feitas, quando possível, com as soluções analíticas dos problemas ou com as respostas obtidas utilizando-se o pacote comercial ANSYS 6.0. Em todas as análises utilizaram-se elementos quadrangulares de 8 nós (q8_3dbe) para discretização dos modelos e considerou-se apenas o *solver* iterativo J-BiCG, sendo a tolerância adotada igual a $tol = 10^{-5}$. Menciona-se que para a avaliação de eficiência desse *solver* utilizou-se o *solver* direto de alta precisão, DLSARG, disponível na biblioteca do FORTRAN (rotina IMSL). O computador utilizado nas análises possui um processador Intel de 1GHz e 768 MB de memória RAM.

5.2. Exemplo 1: Problema de transferência de calor

Este exemplo trata da análise de um problema de condução de calor 3D e tem como objetivo principal validar e estudar a performance do algoritmo de acoplamento genérico EC–EC desenvolvido, bem como, avaliar a eficiência do *solver* iterativo em função do condicionamento do sistema.

O domínio do problema é discretizado em oito subregiões e tem-se que a temperatura é prescrita nas duas extremidades do domínio e o fluxo, no restante do contorno, como mostra a Figura 5.1.

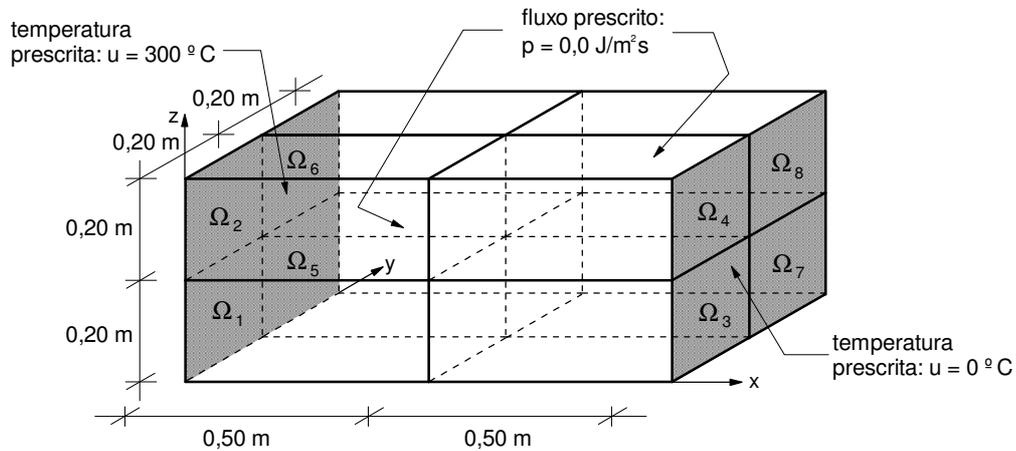


Figura 5.1 – Decomposição de domínio

Para a discretização de cada uma das subregiões utilizaram-se dois modelos de contorno diferentes com 54 e 96 elementos (ver Figura 5.2), resultando em sistemas com, respectivamente, 1312 e 2320 equações.

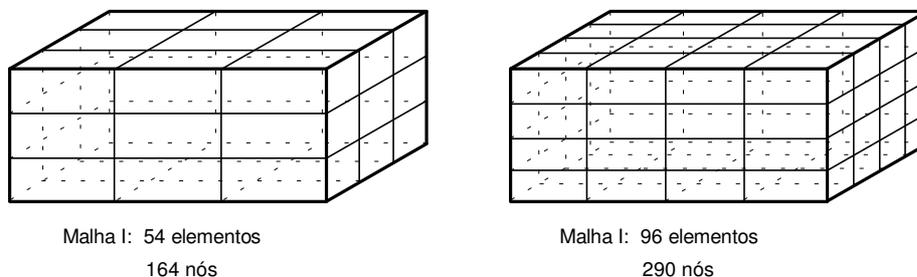


Figura 5.2 – Malhas de elementos de contorno para cada subregião

Na análise, adotaram-se os seguintes valores de condutividade térmica relativa para as subregiões:

$$k_1 = k, k_1 = 10^3 k, k_1 = 10^6 k, k_1 = 10^9 k, k_1 = 10^{10} k, k_1 = 10^{11} k, k_1 = 10^{12} k \text{ e}$$

$$k_2 = k_3 = k_4 = k_5 = k_6 = k_7 = k_8 = k = 1,0;$$

em que, k_i denota a condutividade do domínio Ω_i . Note que adotaram-se valores de condutividade absurdamente elevados de modo a gerar sistemas mal-condicionados.

Mediu-se a temperatura e o fluxo no ponto central de cada uma das subregiões. As respostas obtidas foram comparadas com a solução analítica do problema ($k_1 = k$) e com as soluções obtidas via ANSYS.

Este problema apesar de apresentar solução analítica simples é complexo no ponto de vista de simulação computacional. Considerando-se, por exemplo, a condutividade relativa $k_1 = k$, tem-se como resposta um perfil linear de distribuição de temperatura e fluxo constante ao longo de x . Entretanto, a subdivisão do domínio em várias subregiões implica na existência de vários cantos e arestas nas interfaces entre subregiões, conseqüentemente haverá descontinuidade de fluxo nestes nós. Sendo assim, é conveniente discretizar as interfaces utilizando-se elementos de contorno descontínuos. Os valores de deslocamento, d , adotados para a geração desses elementos foram $d = 0,01$, $d = 0,05$ e $d = 0,1$.

Nos gráficos das Figuras 5.3 e 5.5 apresentam-se as temperaturas no ponto central da subregião 5 calculadas utilizando-se o *solvers* direto (DLSARG) e iterativo (J-BiCG), para $d = 0,05$, em função das condutividades consideradas e dos vários processos integração para, respectivamente, os modelos I e II. Nas Figuras 5.4 e 5.6 analisa-se a variação dessa resposta em função dos valores de d . Menciona-se que estas respostas foram obtidas para $npi = 8 \times 8$. Comparam-se os resultados com os valores obtidos via ANSYS, sendo a malha de elementos finitos composta por 512 elementos do tipo SOLID90 (vinte nós) resultando em um modelo com 2673 nós.

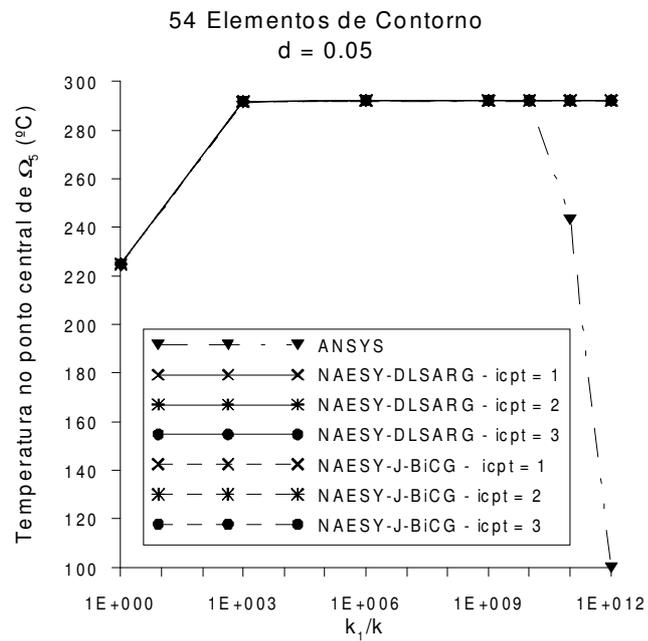


Figura 5.3 – Temperatura calculada com diferentes métodos – modelo I

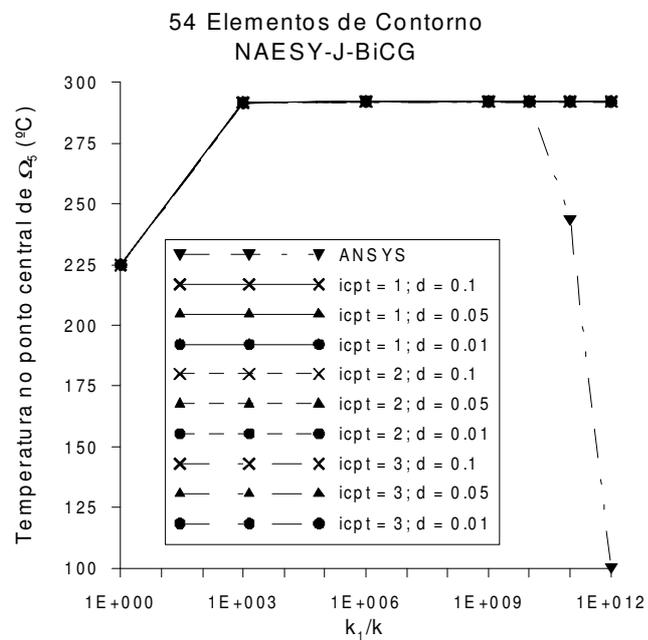


Figura 5.4 – Temperatura obtida para os diferentes valores de d – modelo I

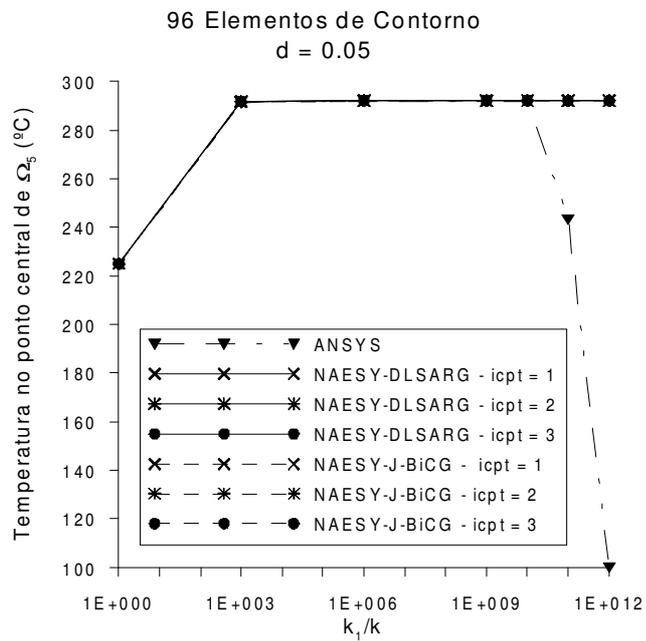


Figura 5.5 – Temperatura calculada com diferentes métodos – modelo II

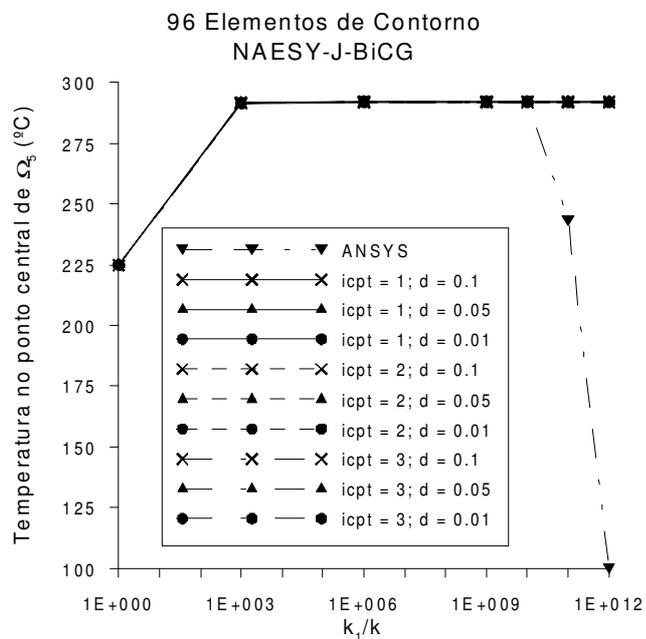


Figura 5.6 – Temperatura obtida para os diferentes valores de d – modelo II

Observa-se a partir das curvas de temperatura que as respostas obtidas utilizando-se o *solver* NAESY-J-BiCG é praticamente coincidente com as obtidas utilizando-se o *solver* de alta precisão NAESY-DLSARG mesmo para sistemas muito mal-condicionados ($k_1 > 10^{10} k$ e $d = 0,01$). Nota-se, também, que para valores de condutividade maiores que $10^{10} k$ não foi possível resolver o problema utilizando-se o ANSYS. Considerando-se os gráficos das Figuras 5.4 e 5.6 vê-se que a resposta do *solver* J-BiCG não varia em função dos valores de d considerados na análise.

Com o objetivo de avaliar a eficiência dos algoritmos de integração, realizou-se um estudo da variação da resposta do potencial e do fluxo no ponto central da subregião 5 para o modelo I e condutividade térmica relativa $k_1 = k$. Neste estudo, a resposta foi medida considerando-se diferentes ordens de integração e calcularam-se os erros com relação à solução analítica do problema, a saber: $u = 225^\circ C$ e $p = 300 J / m^2 s$. Os resultados são apresentados nos gráficos das Figuras de 5.7 a 5.12 para os vários processos de integração e valores de d .

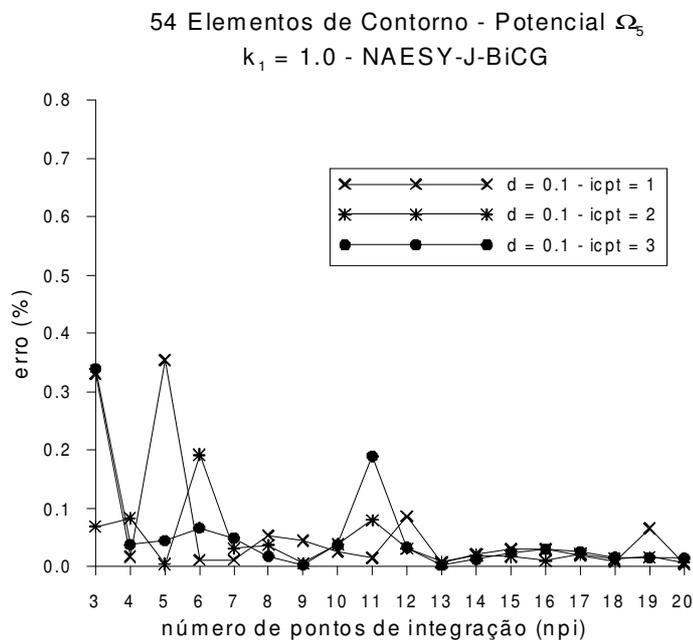


Figura 5.7 – Erro na avaliação do potencial x npi ($d = 0.1$)

54 Elementos de Contorno - Potencial Ω_5
 $k_1 = 1.0$ - NAESY-J-BiCG

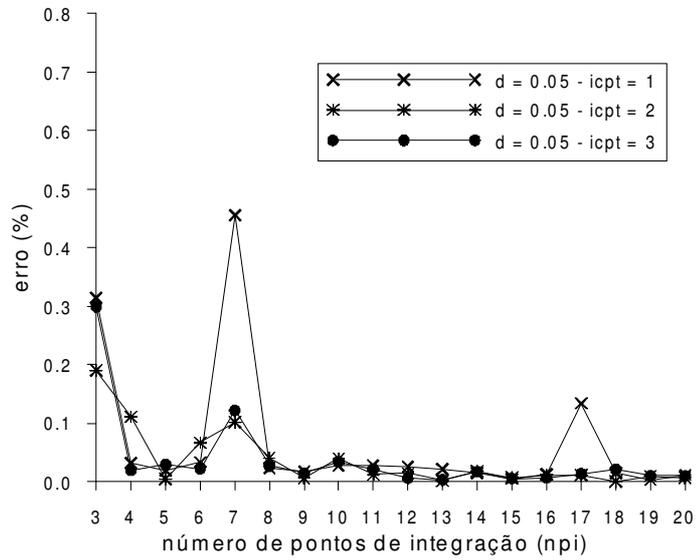


Figura 5.8 – Erro na avaliação do potencial x npi ($d = 0.05$)

54 Elementos de Contorno - Potencial Ω_5
 $k_1 = 1.0$ - NAESY-J-BiCG

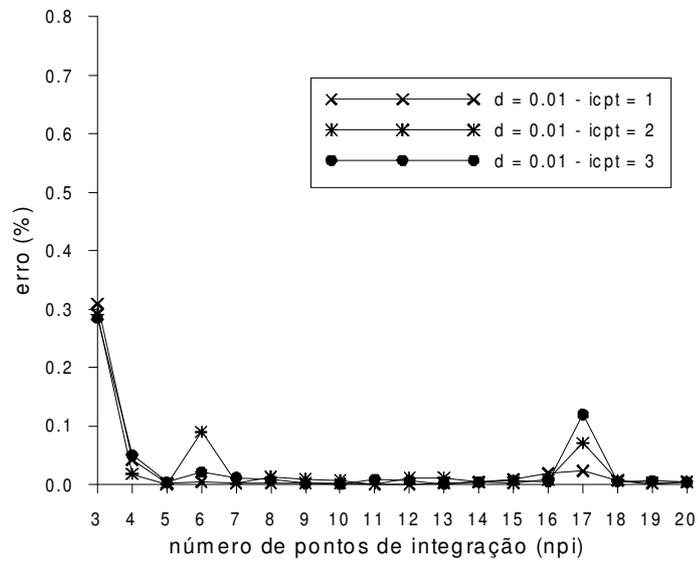


Figura 5.9 – Erro na avaliação do potencial x npi ($d = 0.01$)

54 Elementos de Contorno - Fluxo Ω_5
 $k_1 = 1.0$ - NAESY-J-BiCG

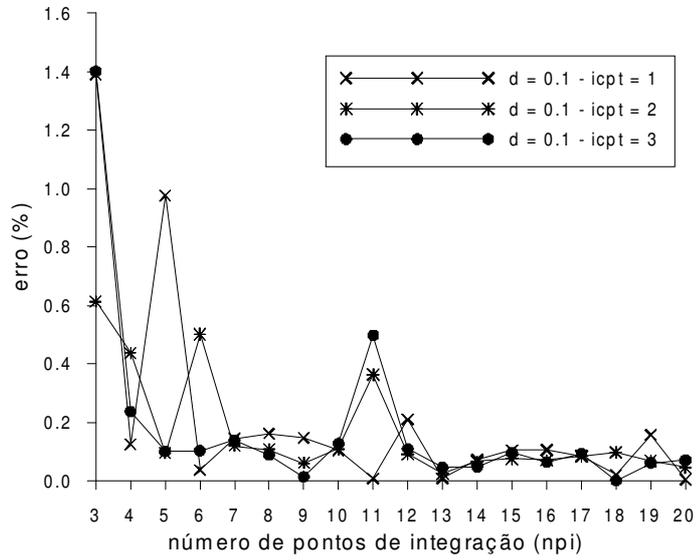


Figura 5.10 – Erro na avaliação do fluxo x npi ($d = 0.1$)

54 Elementos de Contorno - Fluxo Ω_5
 $k_1 = 1.0$ - NAESY-J-BiCG

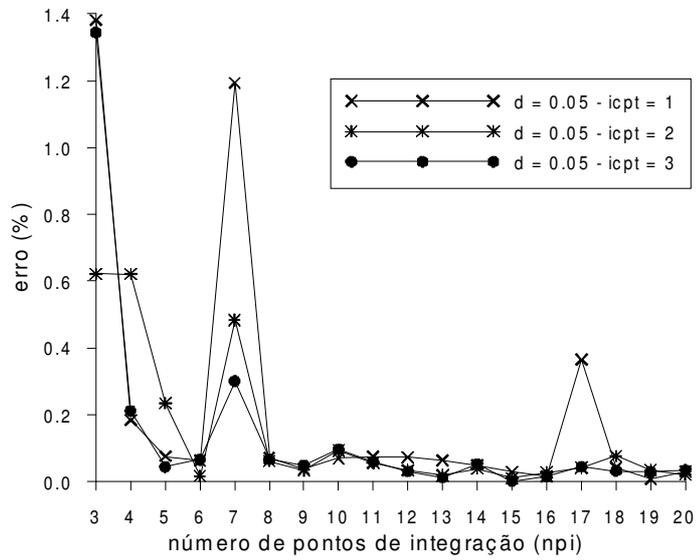


Figura 5.11 – Erro na avaliação do fluxo x npi ($d = 0.05$)

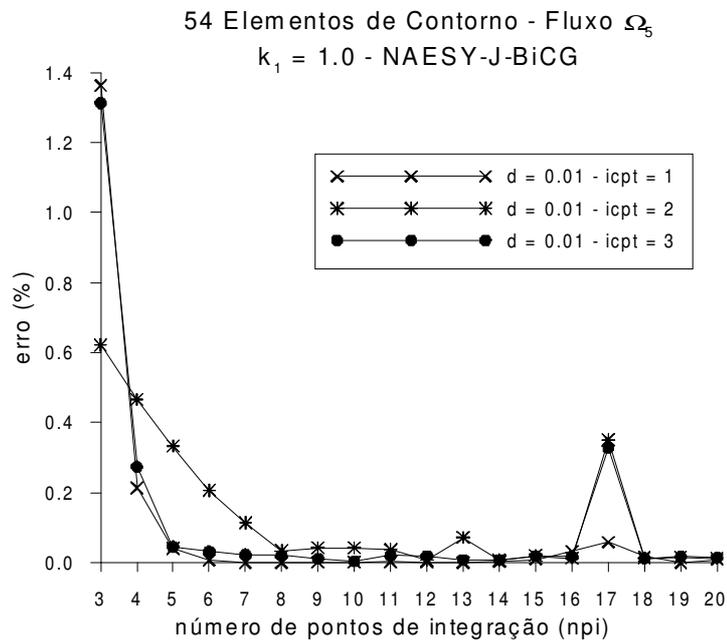


Figura 5.12 – Erro na avaliação do fluxo x npi ($d = 0.01$)

A performance do algoritmo de acoplamento foi avaliada com base na relação entre os tempos de processamento dos *solvers* J-BiCG e DLSARG, e do número de iterações (*nit*) em função da ordem do sistema de equações (n). Os resultados obtidos, para os vários processos de integração e valores de d , são apresentados nos gráficos das Figuras de 5.13 a 5.16 para os modelos I e II, respectivamente. Nas Figuras 5.17 e 5.18 plota-se o tempo de CPU do *solver* DLSARG escalonado pelo tempo de montagem da matriz global do sistema acoplado em função da condutividade relativa, para $d = 0,1$. A ordem de integração adotada foi $n_{pi} = 8 \times 8$. Observa-se que o *solver* J-BiCG, em geral, apresentou melhor desempenho que o *solver* direto, e que não ocorreram grandes alterações no tempo de resolução do sistema, para um mesmo valor de d , em função do processo de integração adotado.

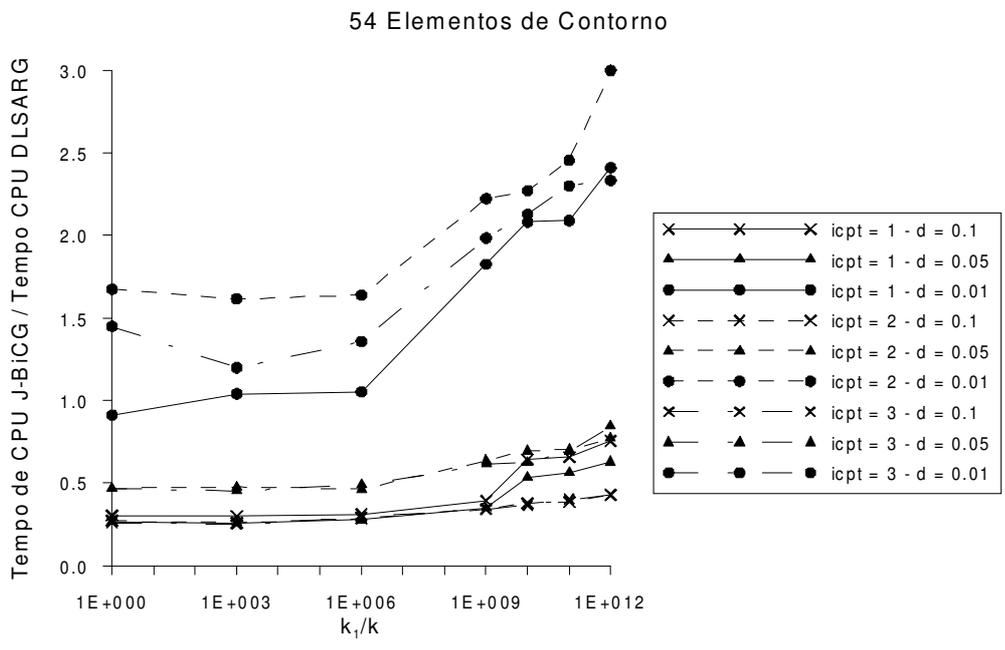


Figura 5.13 – Tempo de CPU dos solvers para o modelo I

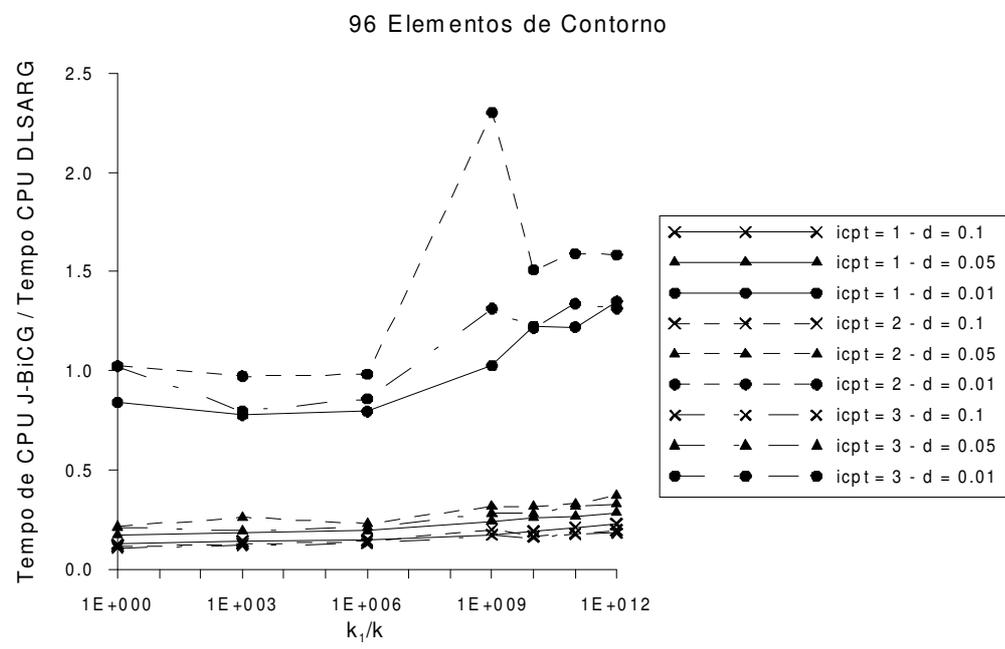


Figura 5.14 – Tempo de CPU dos solvers para o modelo II

54 Elementos de Contorno

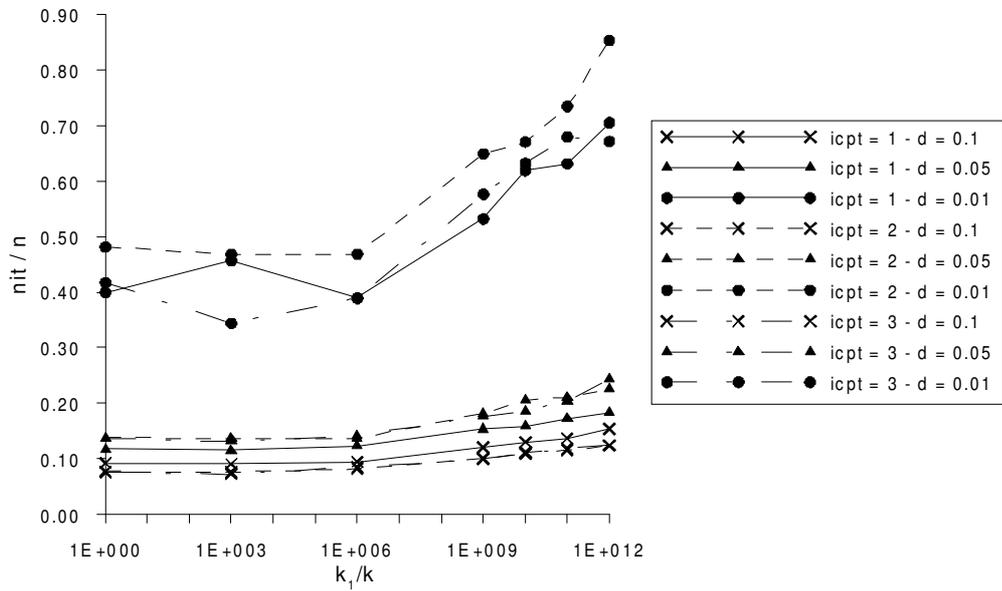


Figura 5.15 – Número de iterações/ordem do sistema: modelo I

96 Elementos de Contorno

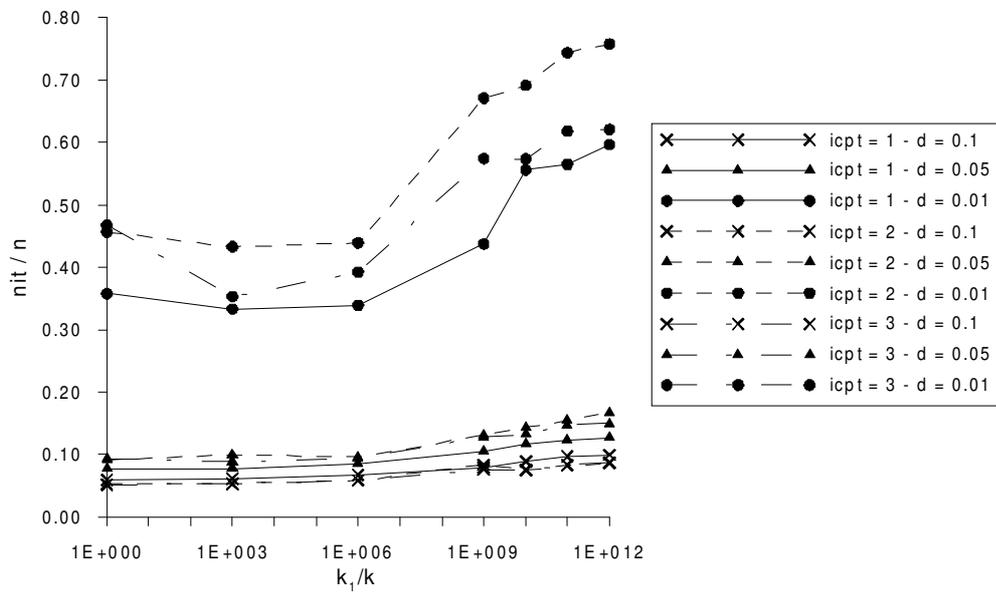


Figura 5.16 – Número de iterações/ordem do sistema: modelo II

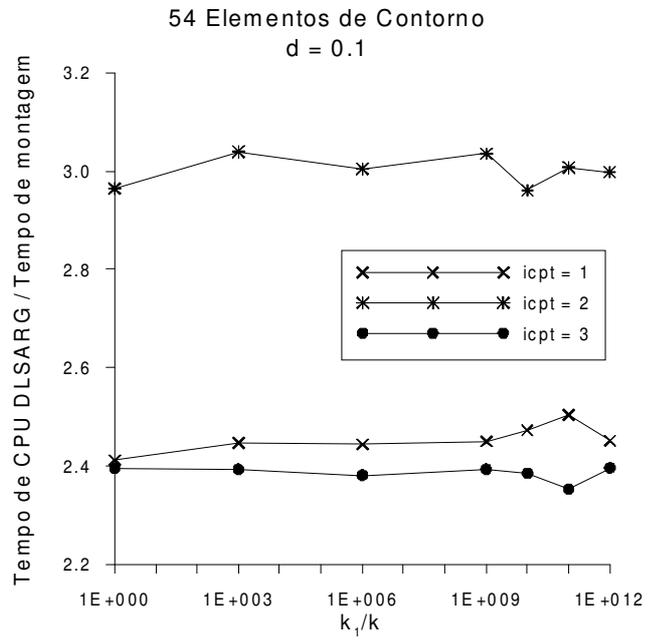


Figura 5.17 – Tempo de CPU DLSARG/Tempo de montagem: modelo I

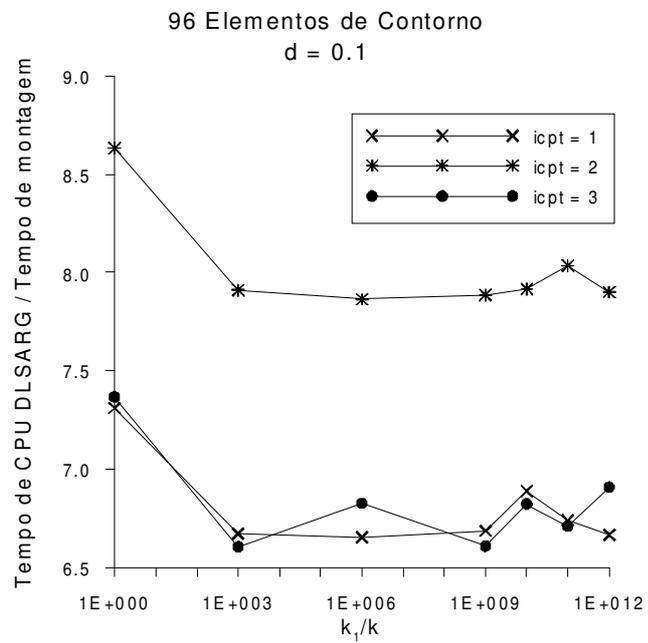


Figura 5.18 – Tempo de CPU DLSARG/Tempo de montagem: modelo II

A partir da rotina IMSL, DLFCRG, calcularam-se os valores estimados do número de condicionamento espectral do sistema de equações, apresentados nos gráficos das Figuras 5.19 e 5.20 em função do valores de d e de condutividade térmica adotados.

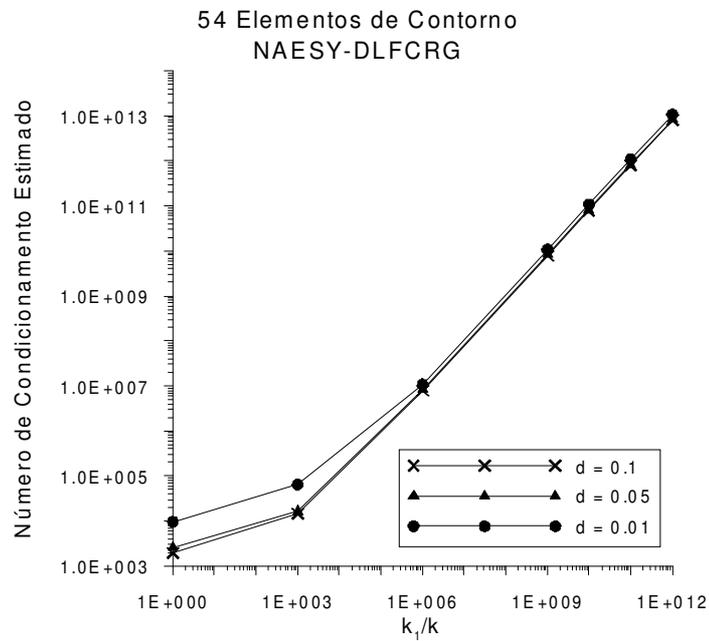


Figura 5.19 – Número de condicionamento estimado para o modelo I

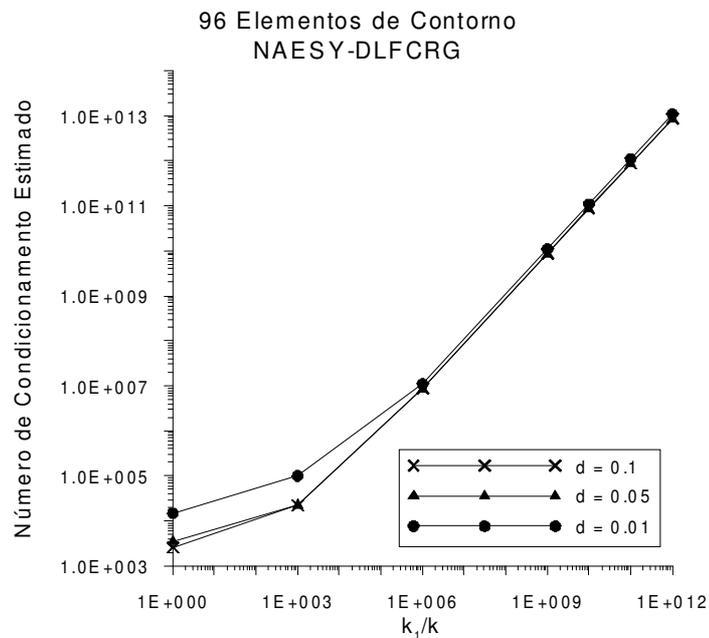


Figura 5.20 – Número de condicionamento estimado para o modelo II

5.3. Exemplo 2: Barragem com cortinas de estaca-prancha

Neste exemplo analisa-se a barragem impermeável com cortinas de estaca-prancha sobre solo isotrópico permeável (Figura 5.21). A superfície lateral e a rocha abaixo da massa de solo são consideradas impermeáveis. Tem-se que a constante de permeabilidade do solo vale $k = 5,08 \times 10^{-4} \text{ m/s}$.

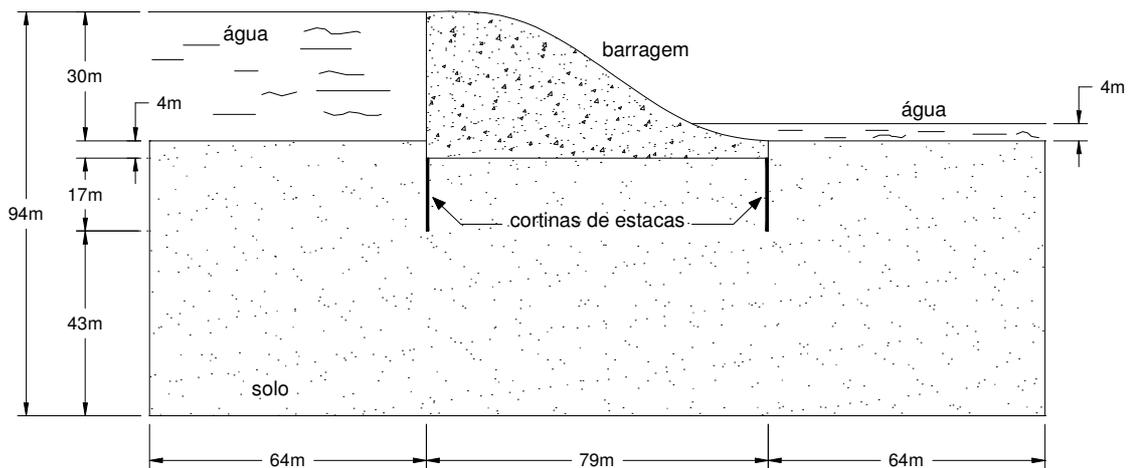


Figura 5.21 – Barragem com cortinas de estaca-prancha

Como condições de contorno tem-se os potenciais de 30m e 4m de coluna d'água aplicados, respectivamente, nas faces superiores das subregiões 1 e 3, e fluxo nulo prescrito nas demais faces do domínio, com exceção das interfaces entre subregiões.

A solução analítica deste problema é fornecida por Lambe e Whitman (1969) e a solução numérica obtida utilizando-se modelos de elementos finitos e de contorno 2D são apresentadas em Brebbia *et al.* (1984).

Devido à presença das cortinas de estaca-prancha a estratégia de subestruturação pode ser usada na análise deste problema. Para isso, subdividiu-se o domínio em três subregiões (ver Figura 5.22). Cada uma destas subregiões foi discretizada utilizando-se uma série de elementos de contorno resultando em um total de dez modelos, através dos quais procurou-se observar a precisão da resposta bem como, avaliar a eficiência do *solver*. A Tabela 5.1 fornece o número de elementos de contorno e o número total de nós para cada um dos modelos adotados.

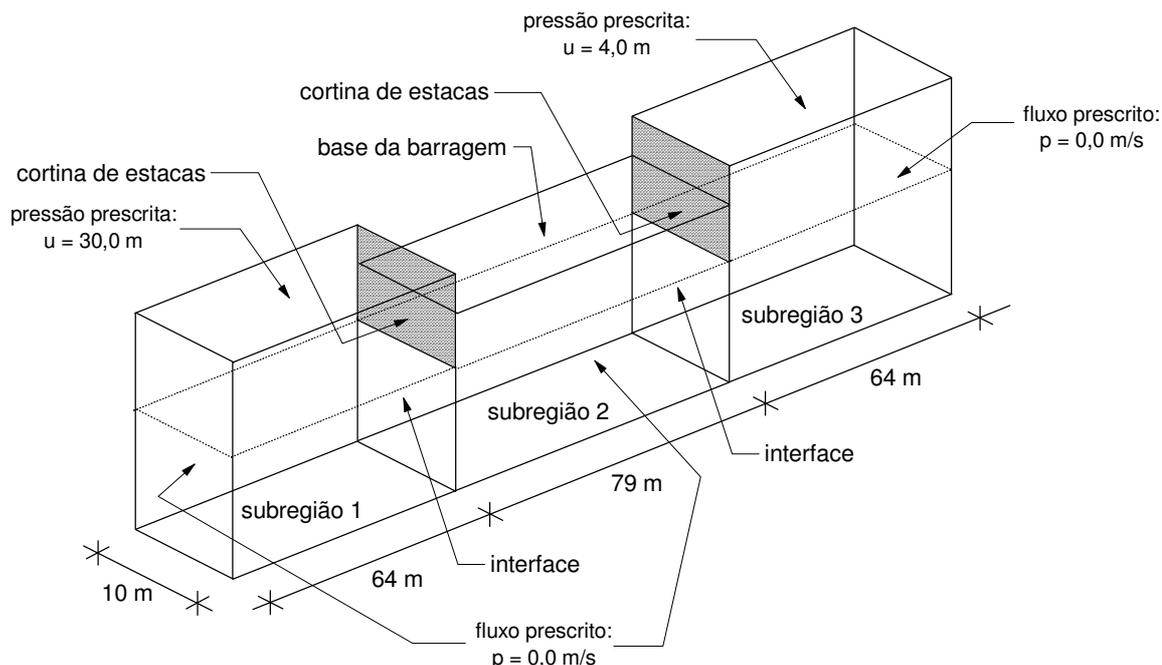


Figura 5.22 – Subestruturação da barragem

Tabela 5.1 – Modelos de EC para a barragem

Modelo	Número de elementos			Número total de nós
	Subregião 1	Subregião 2	Subregião 3	
1	38	48	38	378
2	68	82	68	660
3	104	104	104	942
4	132	152	132	1254
5	168	212	168	1650
6	216	268	216	2106
7	262	318	262	2532
8	312	372	312	2994
9	366	430	366	3492
10	424	492	424	4026

Menciona-se que todos os elementos utilizados nestes modelos são contínuos.

Nos gráficos das Figuras 5.23 e 5.24 apresentam-se, respectivamente, a distribuição de potencial/fluxo ao longo do solo e a pressão sob a base da barragem, considerando-se 8x8 pontos de integração e o modelo de contorno 10. O resultado obtido através do algoritmo de acoplamento desenvolvido (NAESY) é comparado com

o fornecido por Lambe e Whitman (1969). Observa-se que as duas soluções são praticamente coincidentes.

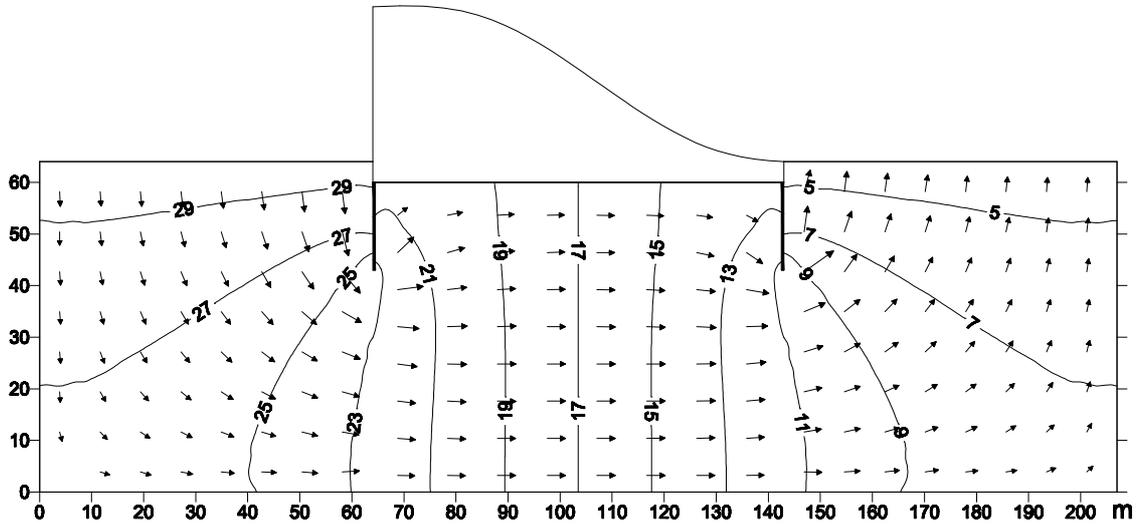


Figura 5.23 – Linhas equipotenciais e distribuição de fluxo

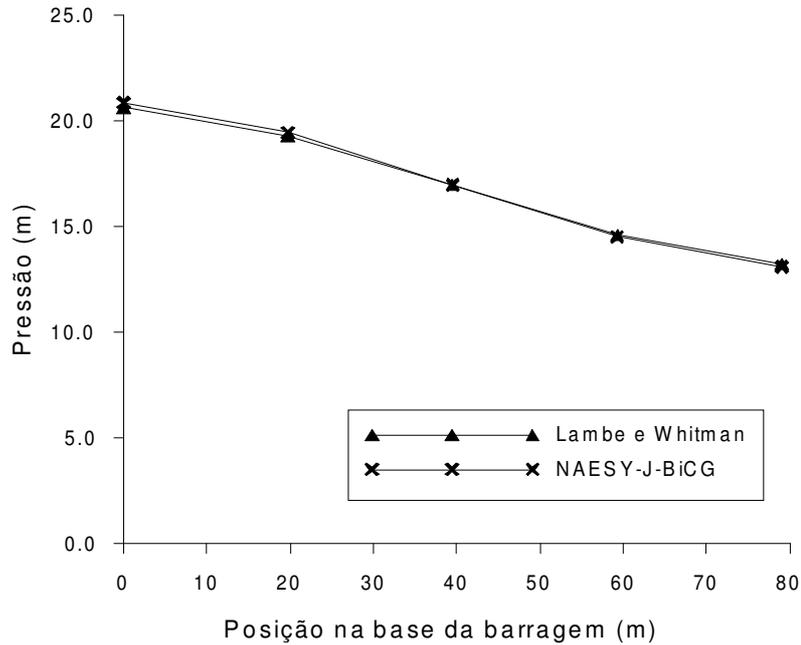


Figura 5.24 – Pressão sob a base da barragem

Com o objetivo de avaliar a eficiência dos processos de integração observou-se a variação da resposta do fluxo em dois pontos do contorno e em um ponto interno ao domínio, pertencentes à subregião 3, em função do número de pontos de integração.

Calcularam-se os erros na avaliação desses fluxos com relação aos valores obtidos considerando-se o processo especial $icpt = 3$ e uma integração de ordem 20×20 . Nos gráficos das Figuras 5.25, 5.26 e 5.27 apresentam-se os erros obtidos e os correspondentes tempos de montagem da matriz do sistema em função da ordem de integração adotada.

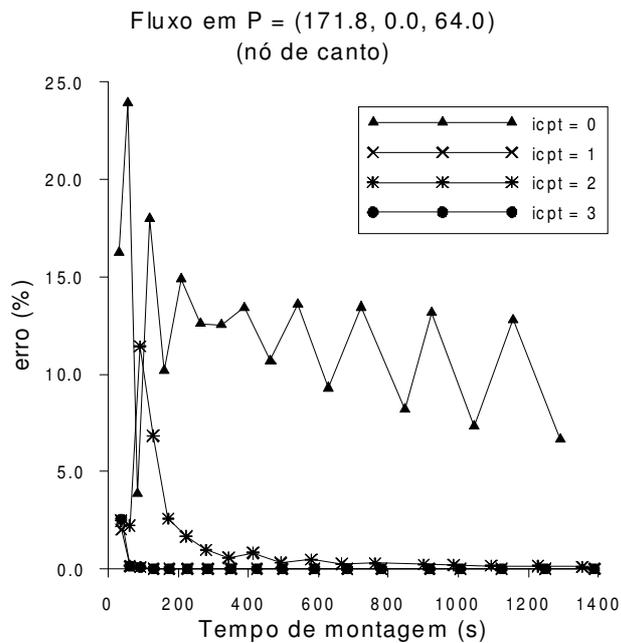


Figura 5.25 – Fluxo no nó $P = (171.8, 0.0, 64.0)$

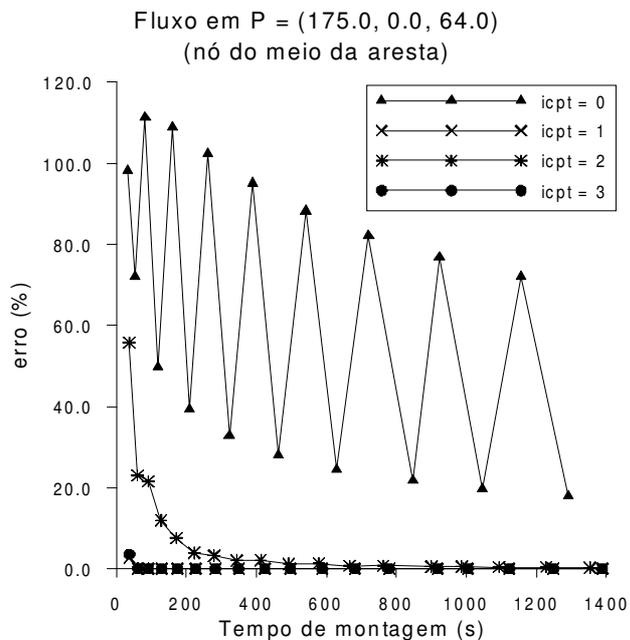


Figura 5.26 – Fluxo no nó $P = (175.0, 0.0, 64.0)$

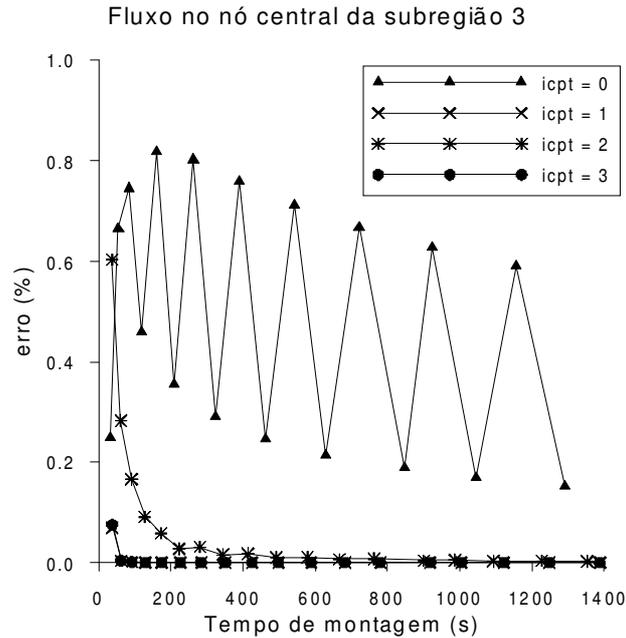


Figura 5.27 – Fluxo no nó central de Ω_3

Observa-se que o processo de integração $icpt = 0$ apresenta erros relativamente elevados e para que resultados precisos sejam obtidos, utilizando-se este processo, necessita-se de um grande número de pontos de integração, portanto este torna-se inviável. Os processos 1 e 3 mostraram-se mais eficientes apresentando resultados bastante precisos mesmo para ordens de integração relativamente baixas. Considerando-se os tempos de montagem do sistema de equações não se nota diferença significativa entre os processos.

No gráfico da Figura 5.28 apresenta-se o tempo de resolução do sistema de equações para os *solvers* J-BiCG e DLSARG, referentes aos vários modelos, em função da ordem do sistema e do processo de integração adotados na análise, considerando-se $n_{pi} = 8 \times 8$.

A esparsidade do sistema, definida como a relação entre o número de coeficientes nulos e o número total de elementos da matriz global do sistema, para este problema é de aproximadamente 63%.

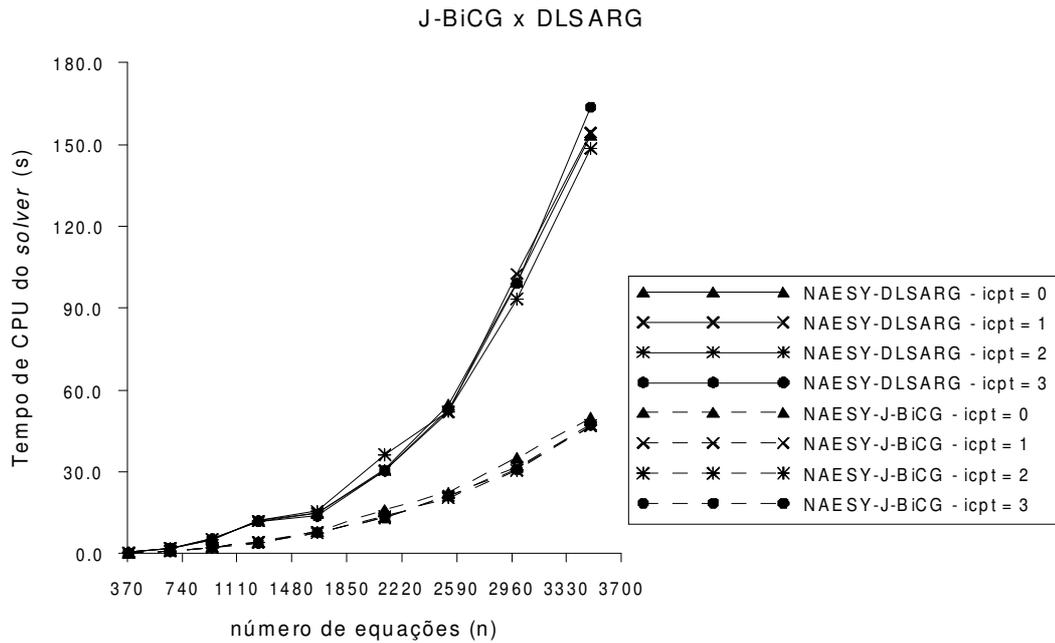


Figura 5.28 – Tempo de CPU dos *solvers*

Nota-se que o *solver* iterativo apresentou desempenho consideravelmente melhor que o *solver* direto, principalmente para os sistemas maiores. Não se verificou, para este problema, alterações significativas nas medidas do tempo de processamento em função do processo de integração adotado.

5.4. Exemplo 3: Casca cilíndrica com enrijecedores internos e fratura

Um dos principais objetivos deste exemplo é estudar a eficiência dos processos especiais de integração ($icpt = 1$ e 3) na análise de sistemas estruturais compostos por elementos de parede fina o que, conseqüentemente, implica na avaliação de integrais *quasi*-singulares. Para tal, considera-se a casca cilíndrica com enrijecedores internos e fratura submetida a pressão interna ($p_0 = 1MPa$) apresentada na Figura 5.29.

As propriedades físicas e geométricas do problema são $E = 205000MPa$, $\nu = 0,0$, e adotaram-se os seguintes valores de espessura para a casca cilíndrica: $h = 0,025a$, $0,050a$, $0,075a$ e $0,100a$.

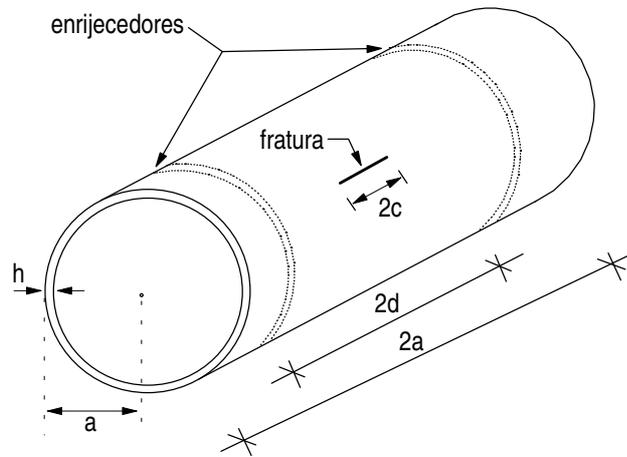


Figura 5.29 – Casca cilíndrica

Devido à simetria do problema apenas um oitavo da casca é discretizada (Figuras 5.30 e 5.31).

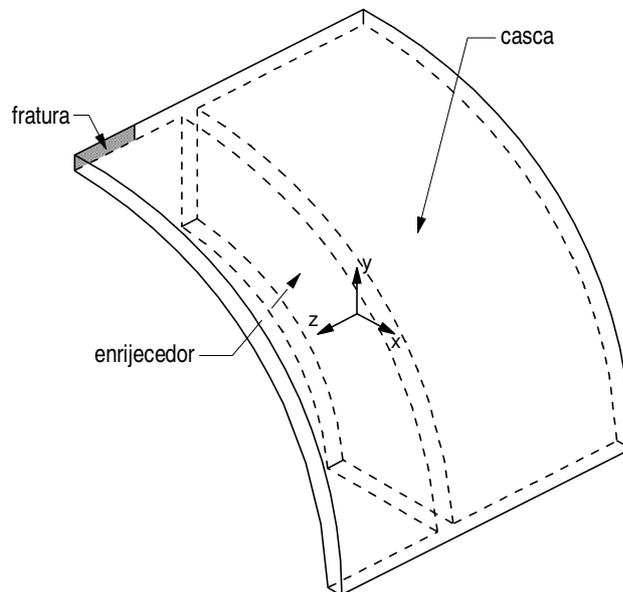


Figura 5.30 – Um oitavo da casca

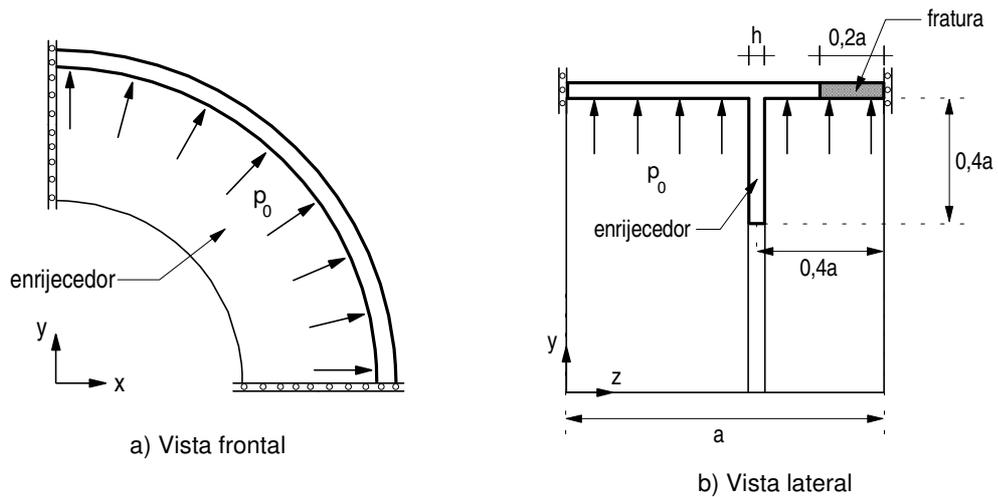


Figura 5.31 – Vistas da casca

A discretização é feita utilizando-se três modelos de subestruturação distintos. No modelo 1, utilizam-se duas subregiões (1 para a casca e 1 para o enrijecedor), no modelo 2, seis subregiões (2x2 para a casca e 2 para o enrijecedor), e no modelo 3, doze subregiões (3x3 para a casca e 3 para o enrijecedor). Em cada um destes modelos de subestruturação empregaram-se três malhas de elementos de contorno diferentes; sendo que utilizam-se elementos de contorno descontínuos nas interfaces entre subregiões. As tabelas 5.2, 5.3 e 5.4 apresentam a descrição das malhas adotadas nas análises.

Tabela 5.2 – Dados das malhas para o Modelo 1

Modelo 1	Número de elementos (ne) e de nós (nno)			
	$h = 0,025a$		$h = 0,050a, 0,075a$ e $0,100a$	
Malha	ne	nno	ne	nno
1	192	580	192	580
2	490	1474	392	1180
3	656	1972	490	1474

Tabela 5.3 – Dados das malhas para o Modelo 2

Modelo 2	Número de elementos (ne) e de nós (nno) para todos os valores de h adotados	
	ne	nno
Malha		
1	276	840
2	368	1116
3	492	1488

Tabela 5.4 – Dados das malhas para o Modelo 3

Modelo 3	Número de elementos (ne) e de nós (nno) para todos os valores de h adotados	
Malha	ne	nno
1	318	978
2	576	1752
3	666	2022

As respostas em termos de deslocamento na direção y ao longo do eixo z na região da fratura, para todos os modelos e espessuras adotadas, são apresentadas nos gráficos das Figuras de 5.32 a 5.43, em função do processo especial de integração para $npi = 8 \times 8$. Nestes gráficos faz-se uma comparação entre os resultados obtidos através do NAESY-J-BiCG e os obtidos via ANSYS.

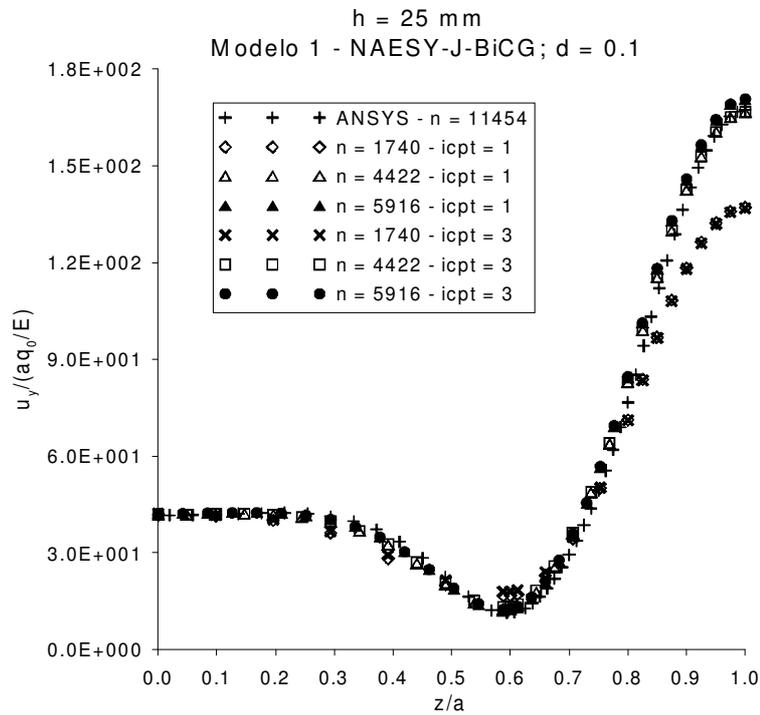


Figura 5.32 – Deslocamento u_y ao longo do eixo z : $h = 0,025a$; modelo 1

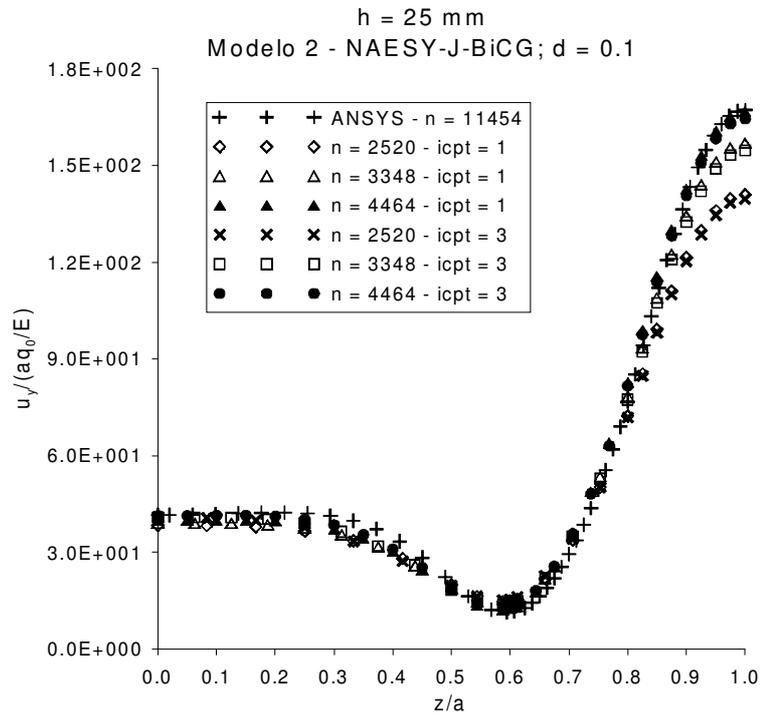


Figura 5.33 – Deslocamento u_y ao longo do eixo z : $h = 0,025a$; modelo 2

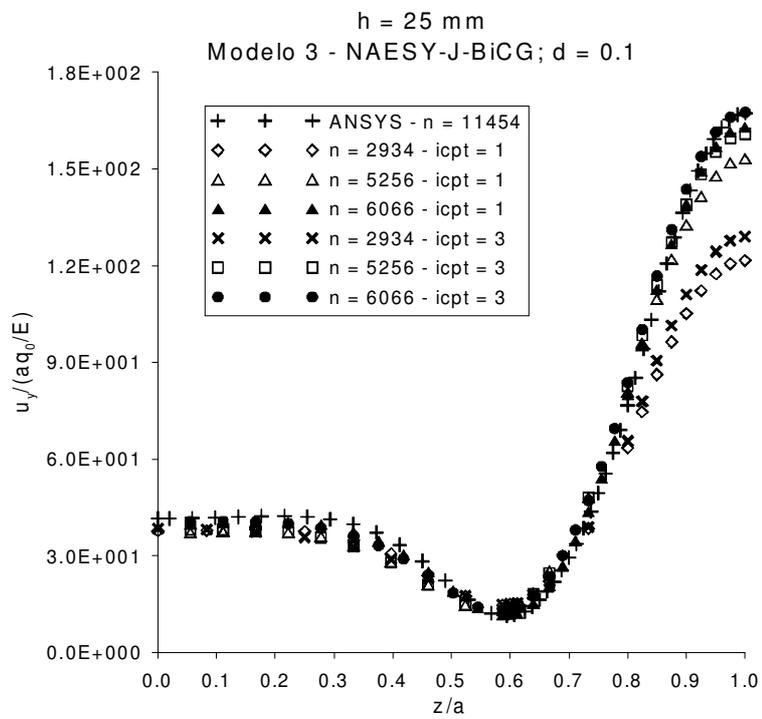


Figura 5.34 – Deslocamento u_y ao longo do eixo z : $h = 0,025a$; modelo 3

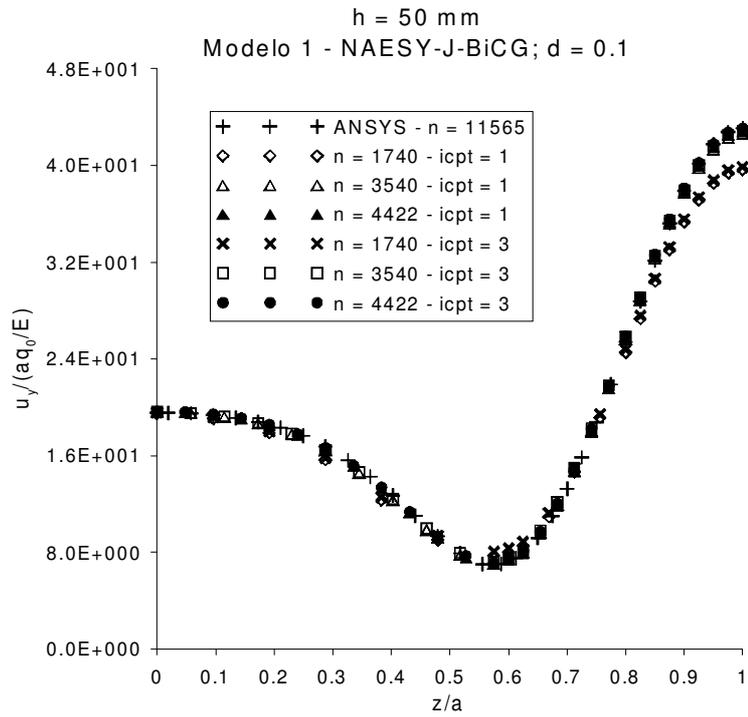


Figura 5.35 – Deslocamento u_y ao longo do eixo z : $h = 0,050a$; modelo 1

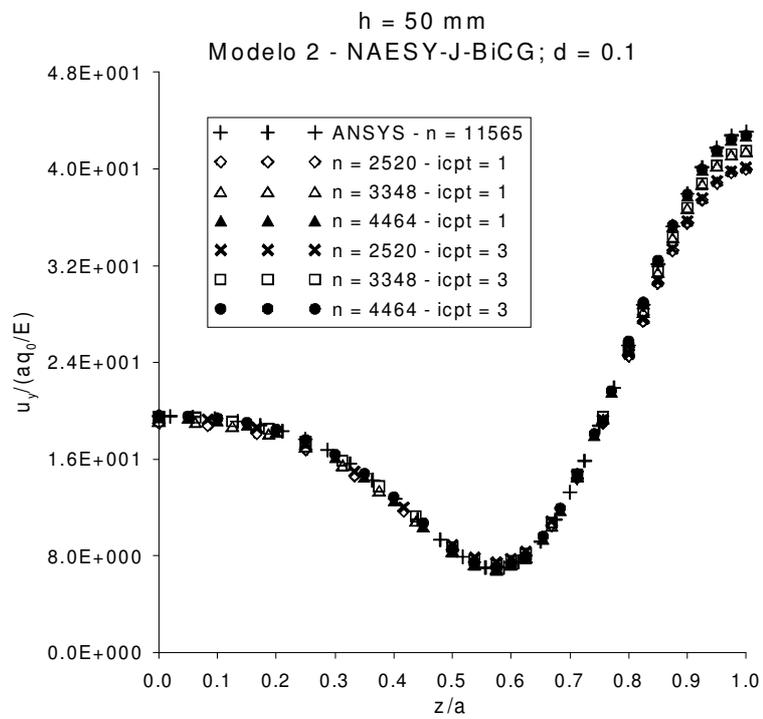


Figura 5.36 – Deslocamento u_y ao longo do eixo z : $h = 0,050a$; modelo 2

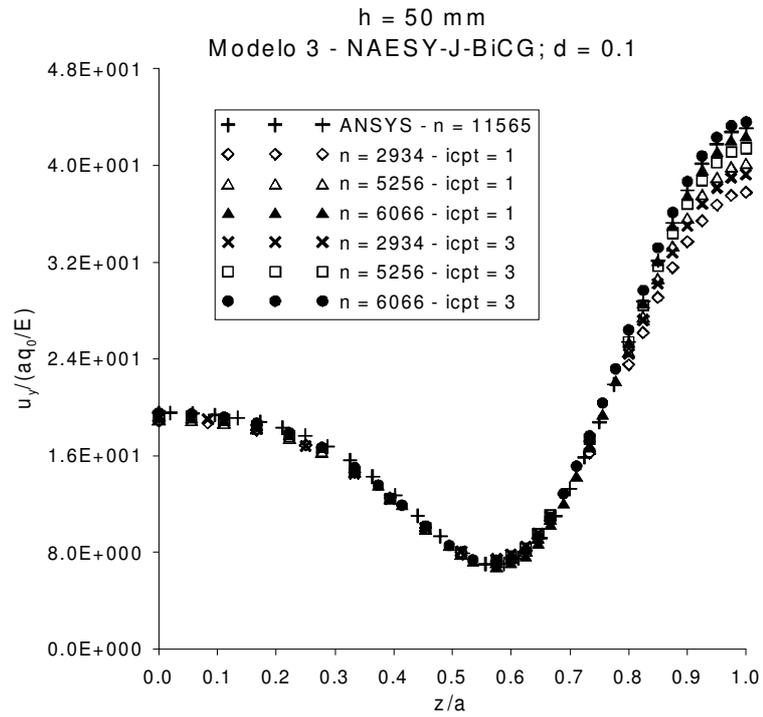


Figura 5.37 – Deslocamento u_y ao longo do eixo z : $h = 0,050a$; modelo 3

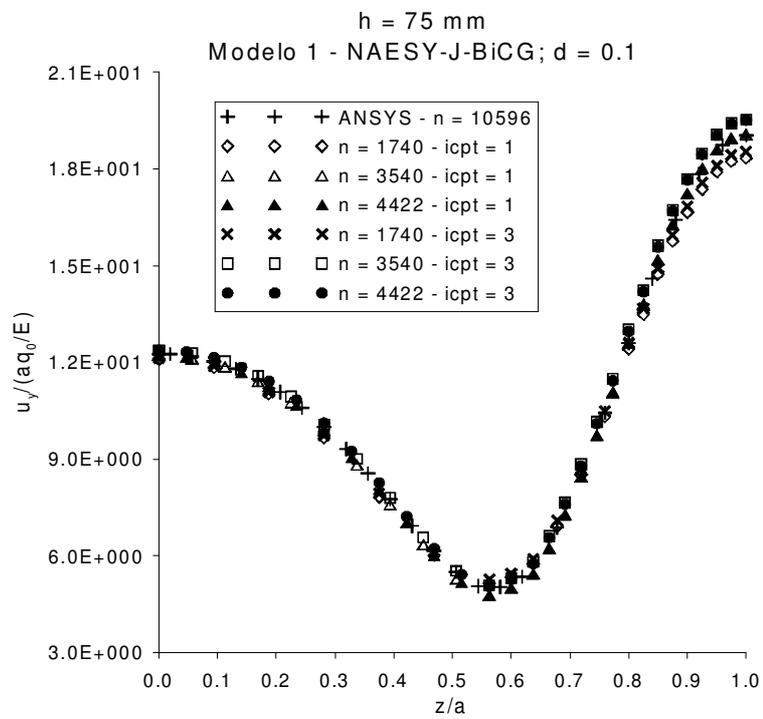


Figura 5.38 – Deslocamento u_y ao longo do eixo z : $h = 0,075a$; modelo 1

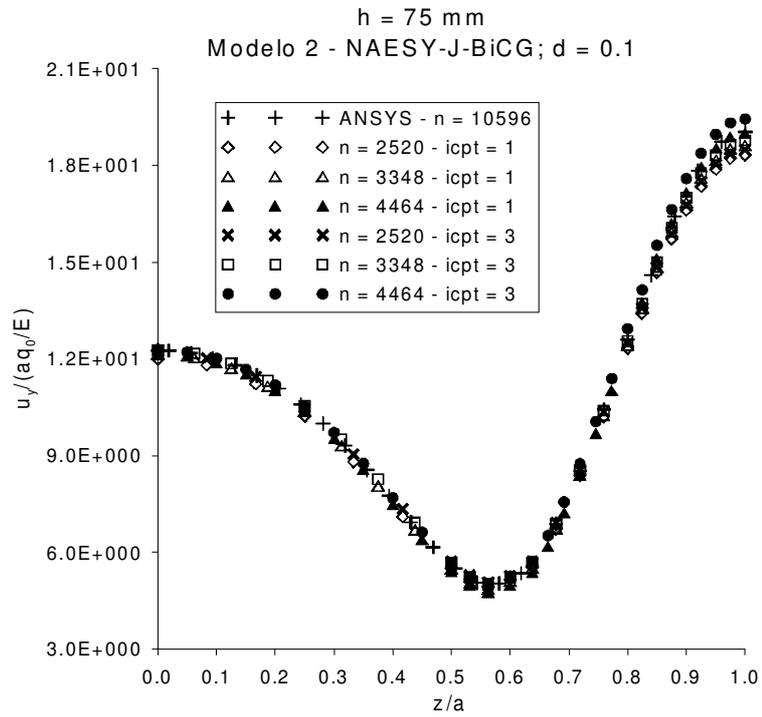


Figura 5.39 – Deslocamento u_y ao longo do eixo z : $h = 0,075a$; modelo 2

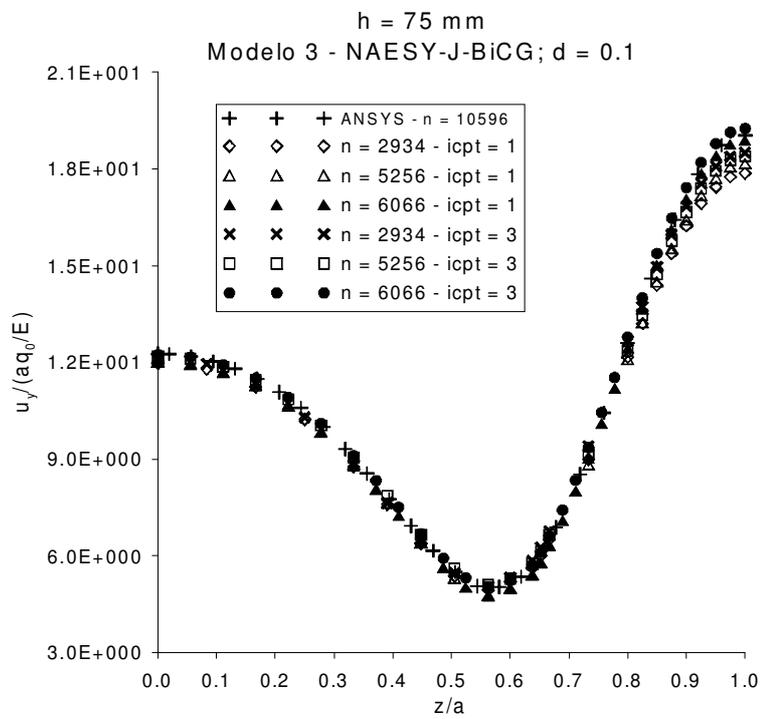


Figura 5.40 – Deslocamento u_y ao longo do eixo z : $h = 0,075a$; modelo 3

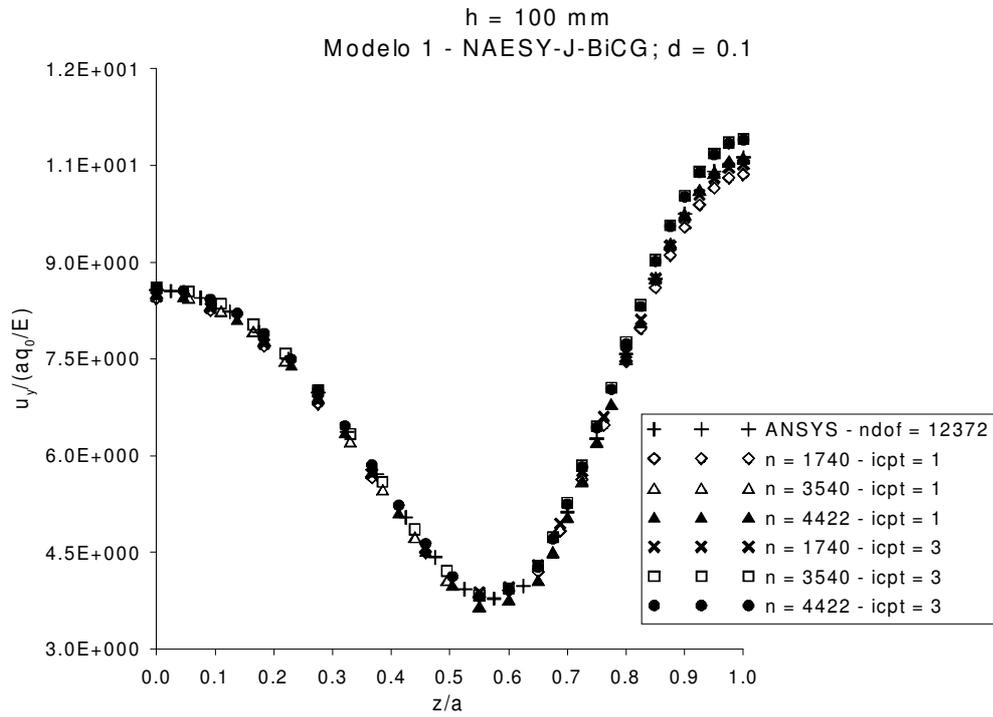


Figura 5.41 – Deslocamento u_y ao longo do eixo z : $h = 0,100a$; modelo 1

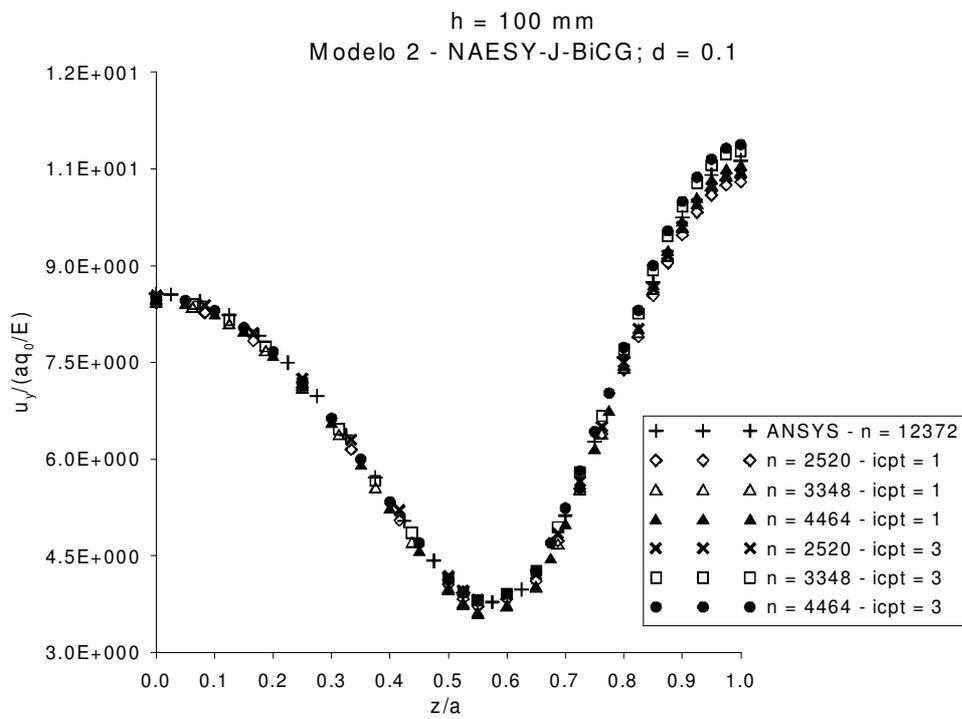


Figura 5.42 – Deslocamento u_y ao longo do eixo z : $h = 0,100a$; modelo 2

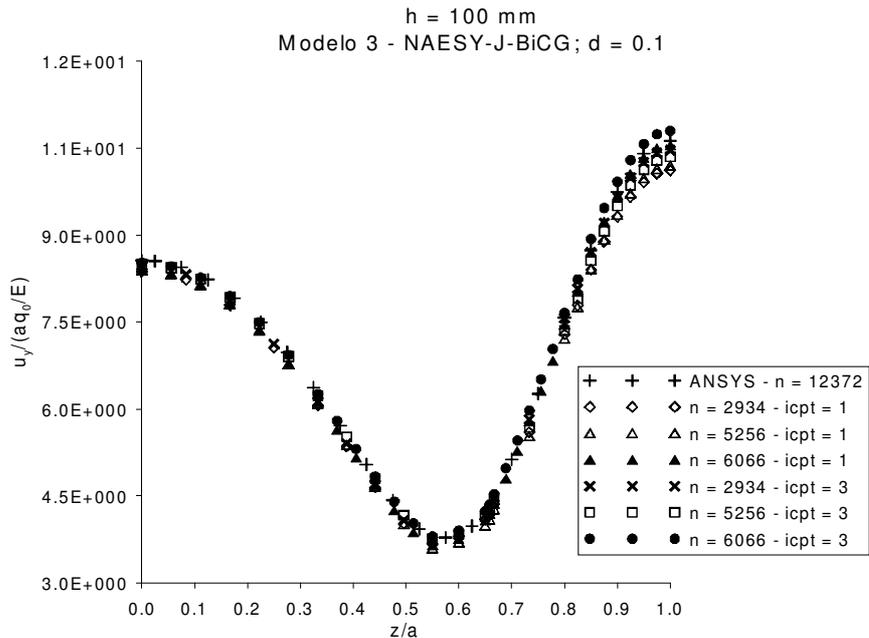


Figura 5.43 – Deslocamento u_y ao longo do eixo z : $h = 0,100a$; modelo 3

Menciona-se que também se utilizou, juntamente com o programa NAESY, o *solver* direto IMSL–DLSARG de alta precisão na análise deste problema. Todavia, esse apresenta uma limitação no que diz respeito à ordem do sistema de equações a ser resolvido. Sendo assim, nem todos os modelos de malhas adotados puderam ser resolvidos a partir deste *solver*. Nos gráficos abaixo apresentam-se os resultados.

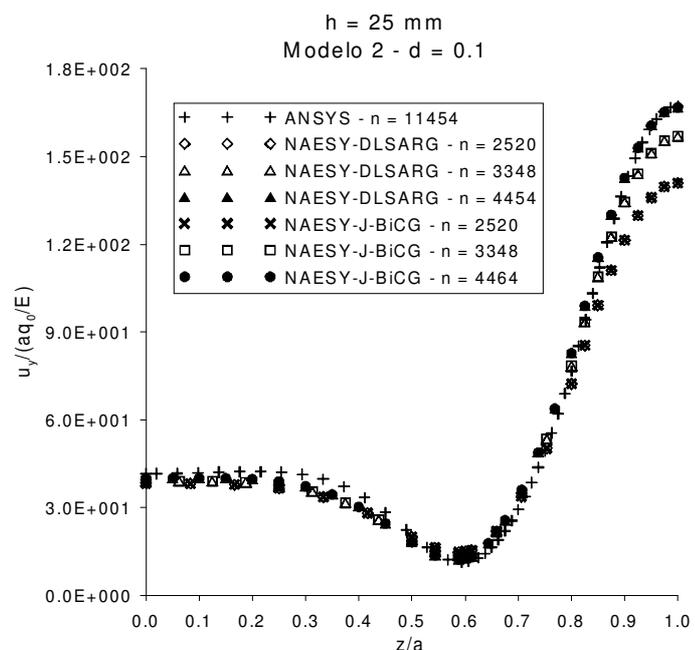


Figura 5.44 – DLSARG x J–BiCG: $h = 0,025a$; modelo 2

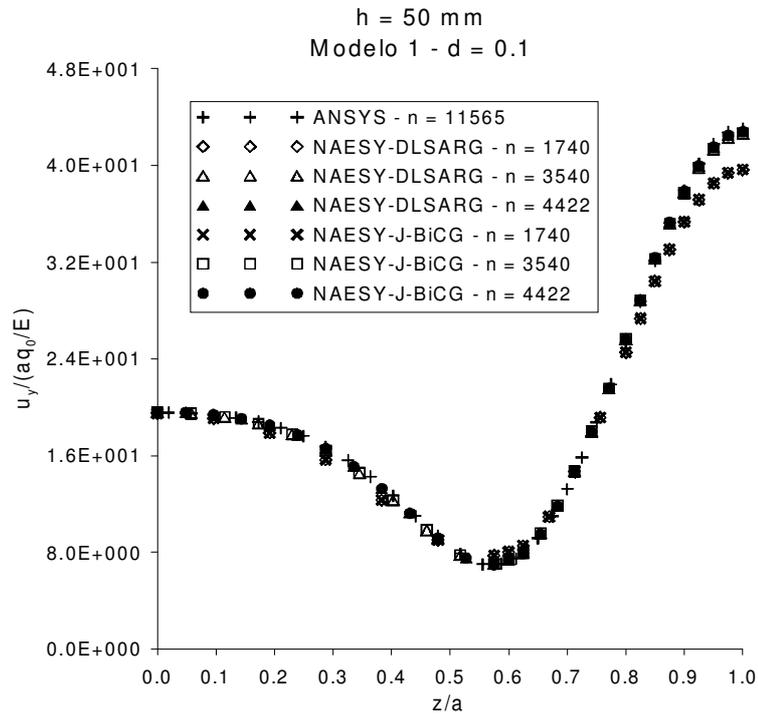


Figura 5.45 – DLSARG x J-BiCG: $h = 0,050a$; modelo 1

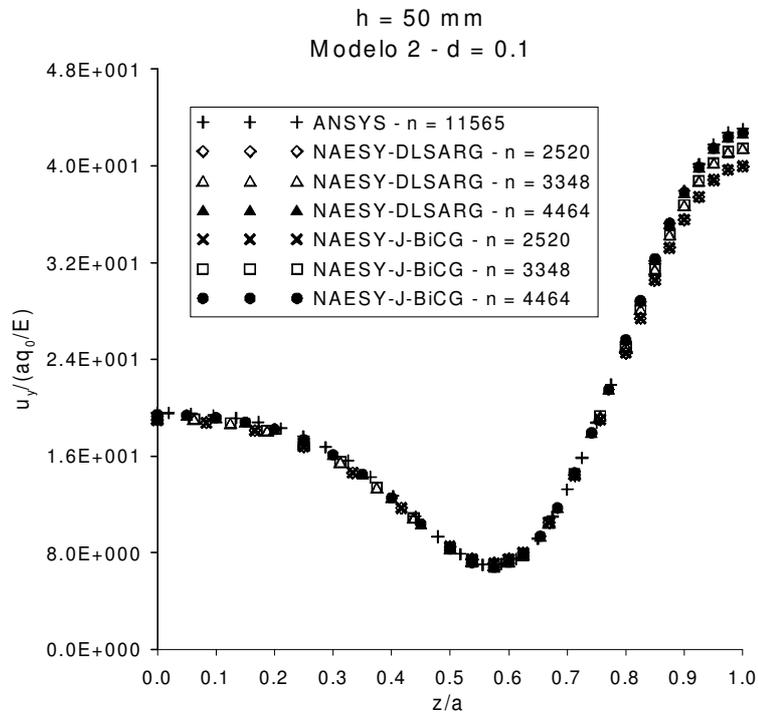


Figura 5.46 – DLSARG x J-BiCG: $h = 0,050a$; modelo 2

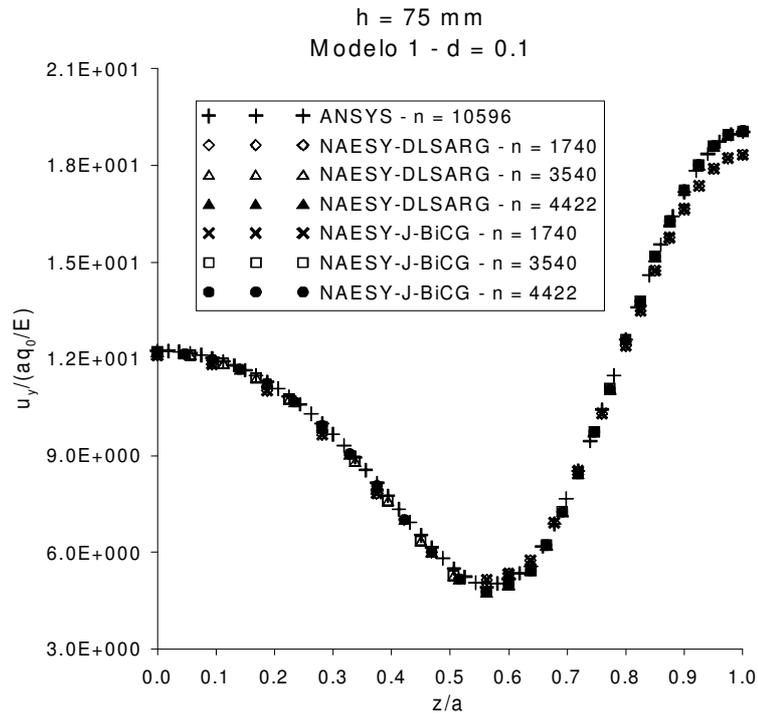


Figura 5.47 – DLSARG x J-BiCG: $h = 0,075a$; modelo 1

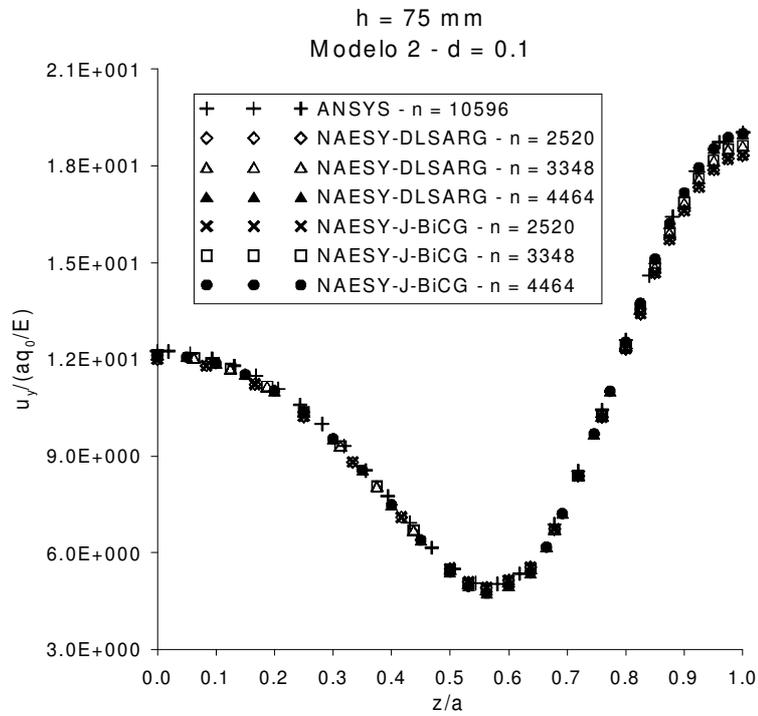


Figura 5.48 – DLSARG x J-BiCG: $h = 0,075a$; modelo 2

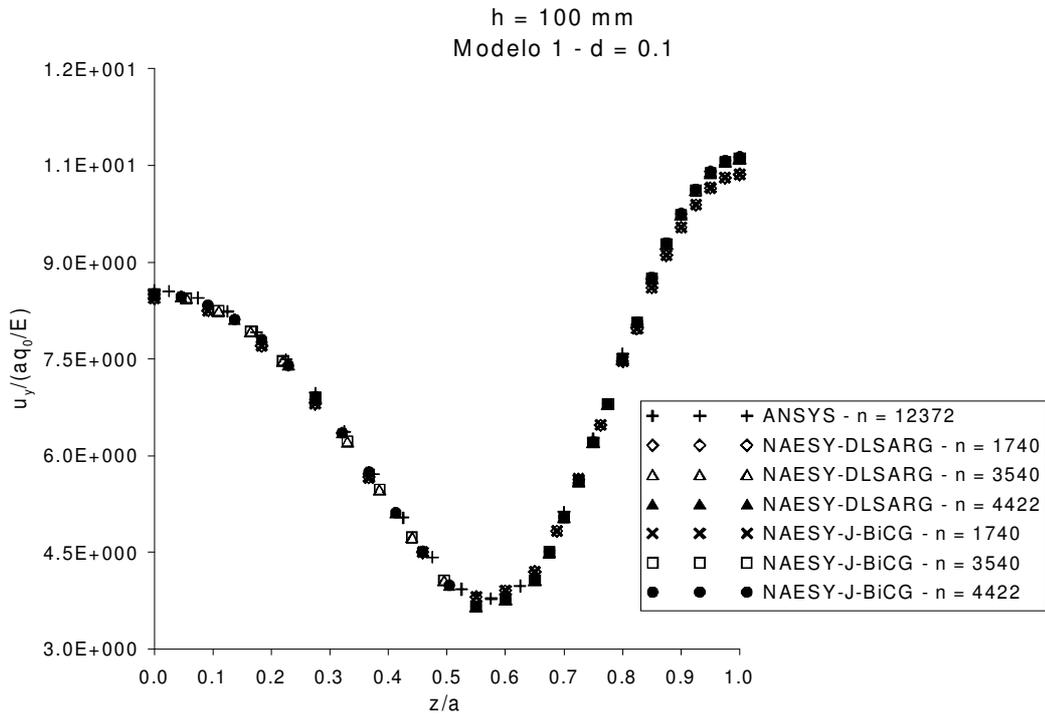


Figura 5.49 – DLSARG x J-BiCG: $h = 0,100a$; modelo 1

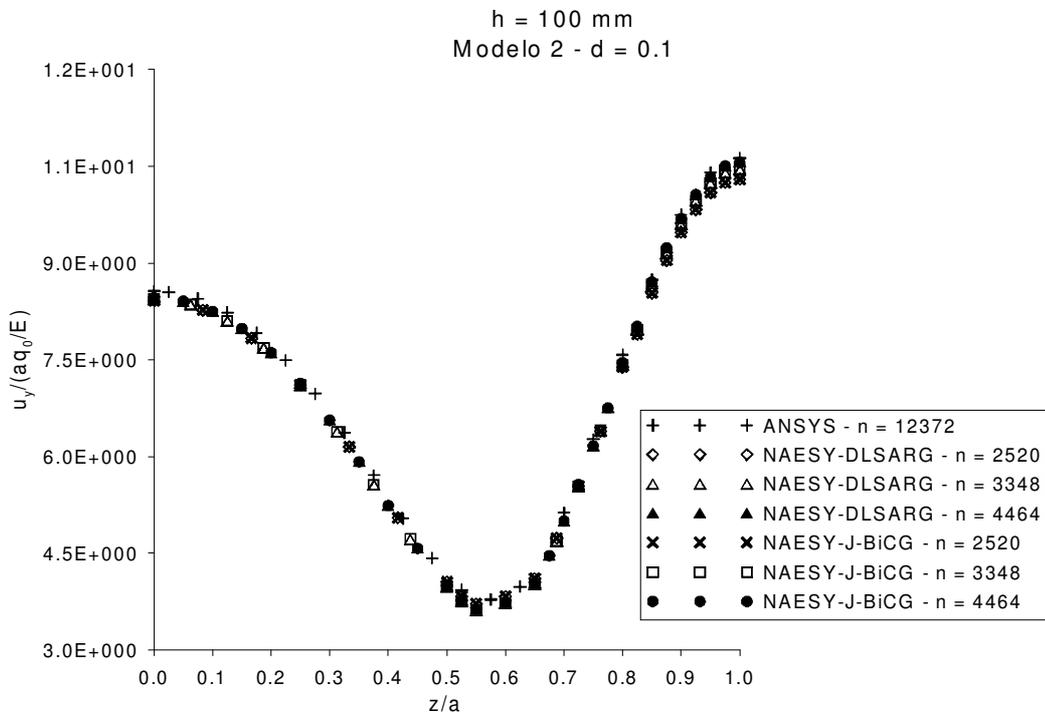


Figura 5.50 – DLSARG x J-BiCG: $h = 0,100a$; modelo 2

Avaliou-se a performance do algoritmo de acoplamento EC-EC por meio da relação entre o tempo de CPU do *solver* iterativo (J-BiCG) e do *solver* direto (DLSARG). Nos gráficos a seguir plotam-se esses resultados.

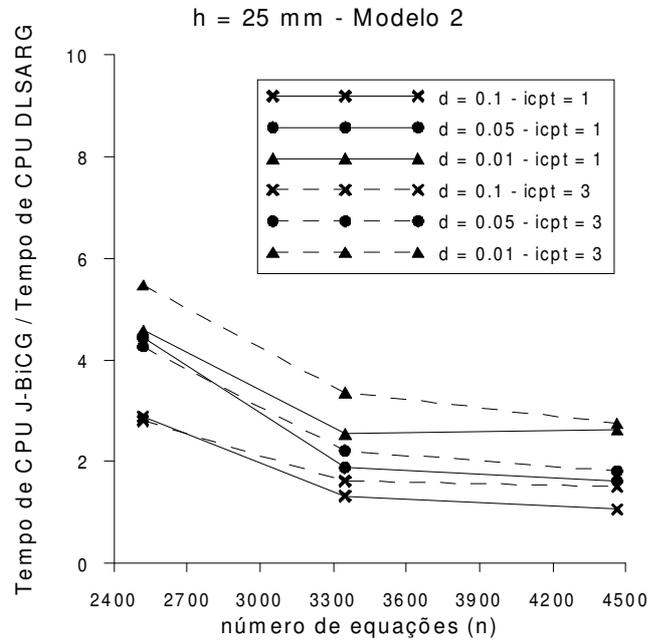


Figura 5.51 – Tempo de CPU dos *solvers*: $h = 0,025a$; modelo 2

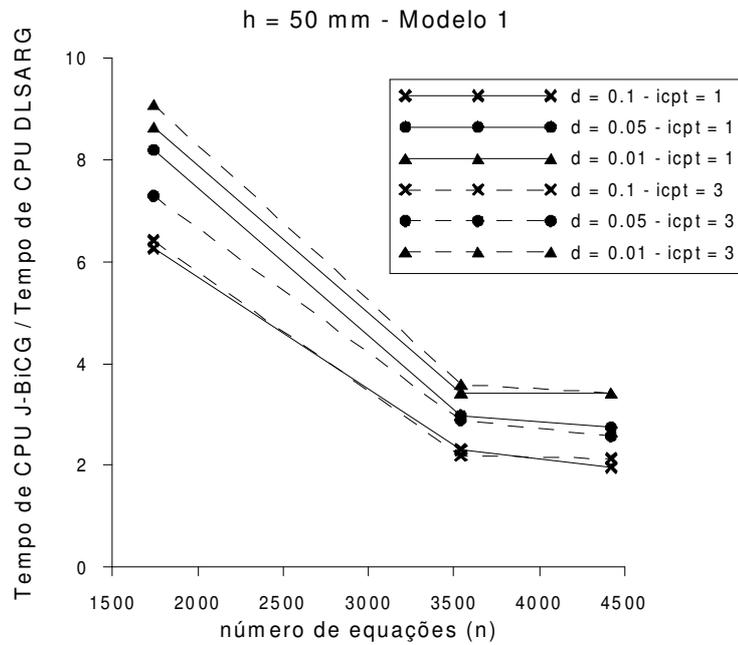


Figura 5.52 – Tempo de CPU dos *solvers*: $h = 0,050a$; modelo 1

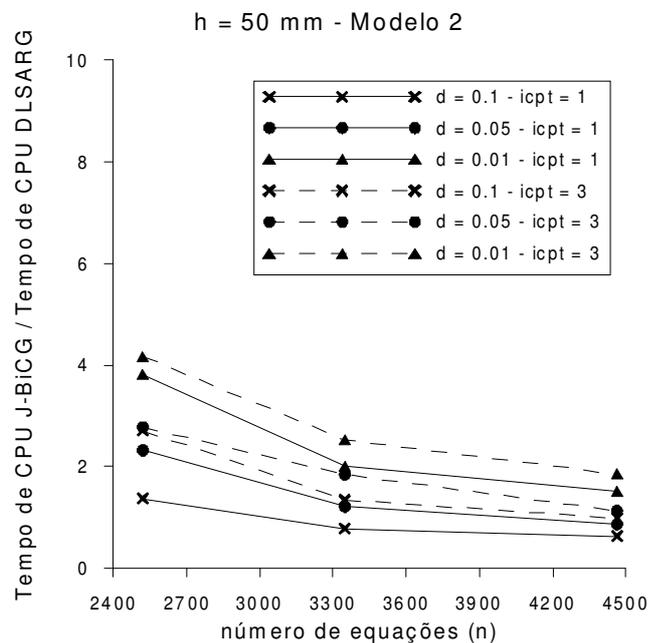


Figura 5.53 – Tempo de CPU dos solvers: $h = 0,050a$; modelo 2

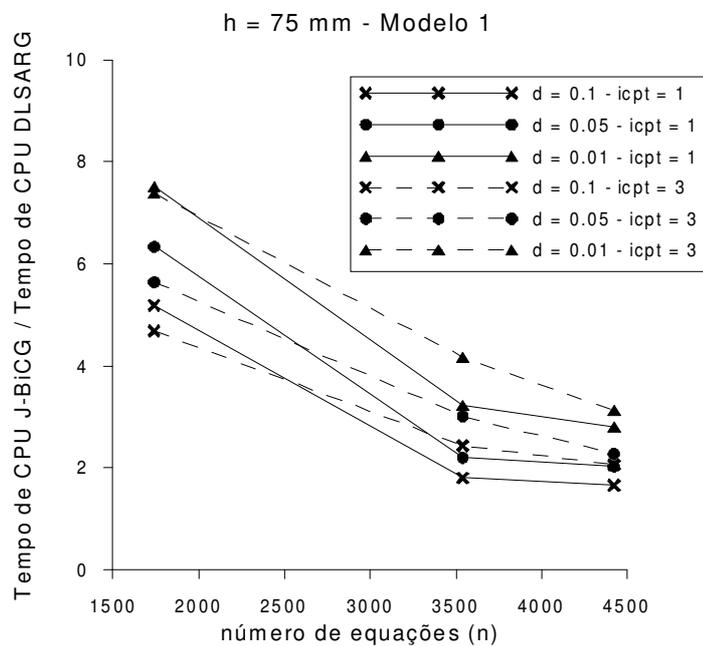


Figura 5.54 – Tempo de CPU dos solvers: $h = 0,075a$; modelo 1

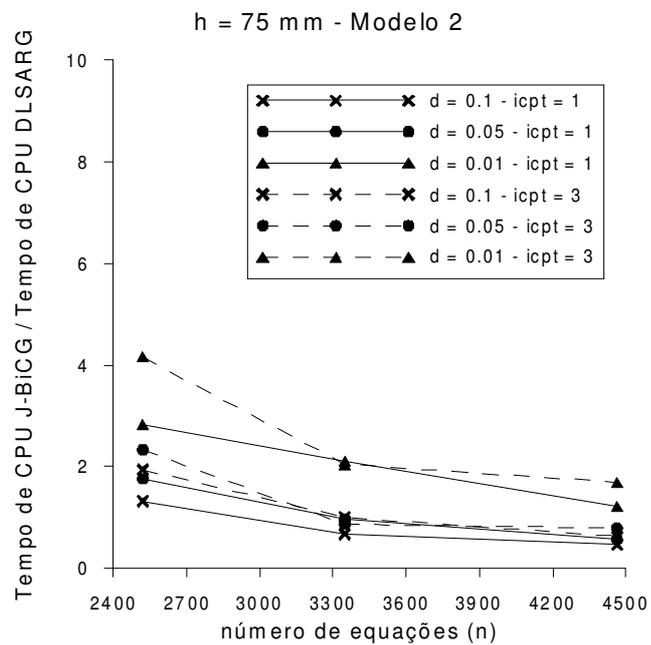


Figura 5.55 – Tempo de CPU dos solvers: $h = 0,075a$; modelo 2

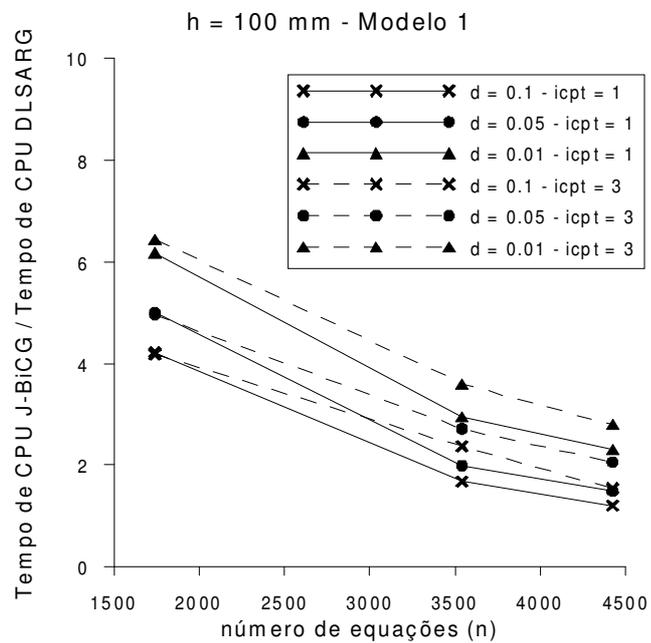


Figura 5.56 – Tempo de CPU dos solvers: $h = 0,100a$; modelo 1

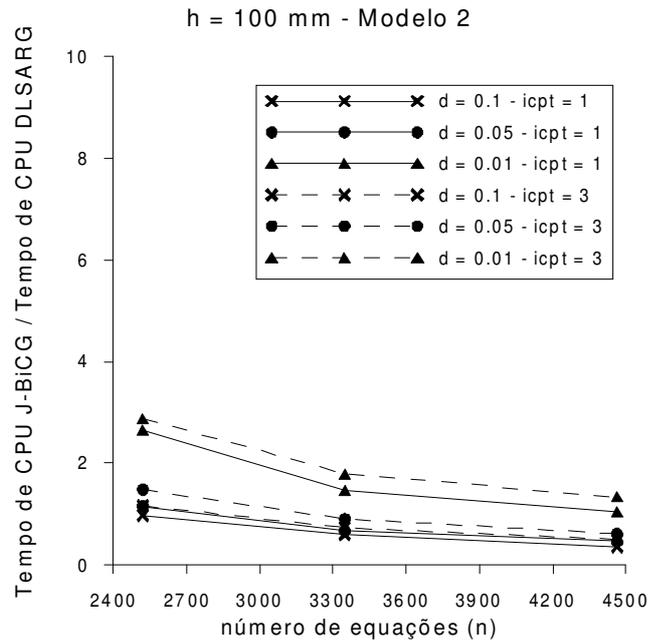


Figura 5.57 – Tempo de CPU dos solvers: $h = 0,100a$; modelo 2

5.5. Exemplo 4: Interação solo–estrutura

Este exemplo apresenta o estudo da interação solo–estrutura e tem como objetivo validar a estratégia de acoplamento para o caso de problemas elastostáticos e estudar a performance do *solver* iterativo.

Realizaram–se as análises de dois problemas distintos. No primeiro, tem–se um bloco de fundação submetido a uma carga uniformemente distribuída (q) apoiado no solo (ver Figura 5.58), e no segundo problema, consideram–se dois blocos acoplados com o solo (Figuras 5.69 e 5.70).

O módulo de elasticidade e o coeficiente de Poisson do solo são, respectivamente, $E_s = 200MPa$ e $\nu_s = 0,35$. Para o bloco de fundação tem–se $\nu_f = 0,35$ e adotaram–se diversos valores de módulos de elasticidade, a saber: $E_f = E_s$, $E_f = 50E_s$, $E_f = 100E_s$, $E_f = 500E_s$, $E_f = 1000E_s$ e $E_f = 10000E_s$. Deste modo, simulam–se fundações flexíveis e rígidas.

Na(s) face(s) superior(es) do(s) bloco(s) atua uma carga uniformemente distribuída, $q = 4\text{MPa}$.

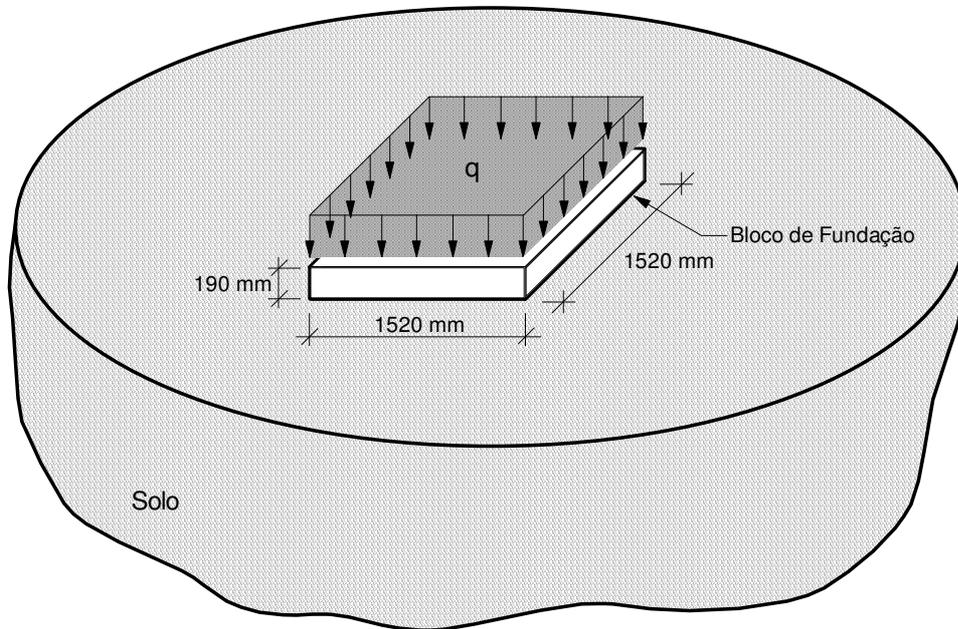


Figura 5.58 – Interação solo–estrutura

Problema 1:

Como mencionado anteriormente, neste problema considera-se o acoplamento entre um bloco de fundação e o solo. Tem-se que o domínio do problema foi subdividido em duas subregiões, uma correspondente ao bloco e outra ao solo.

Nas Figuras 5.59, 5.60 e 5.61 apresenta-se o modelo de contorno adotado, o qual considera apenas elementos de contorno contínuos. O bloco de fundação (Figura 5.61) é discretizado com 264 elementos (794 nós) e o solo (Figura 5.59), com 300 elementos (949 nós), resultando em um sistema acoplado com 5229 equações. A malha de elementos de fechamento (*enclosing elements* – e.e.) possui 80 elementos e 337 nós.

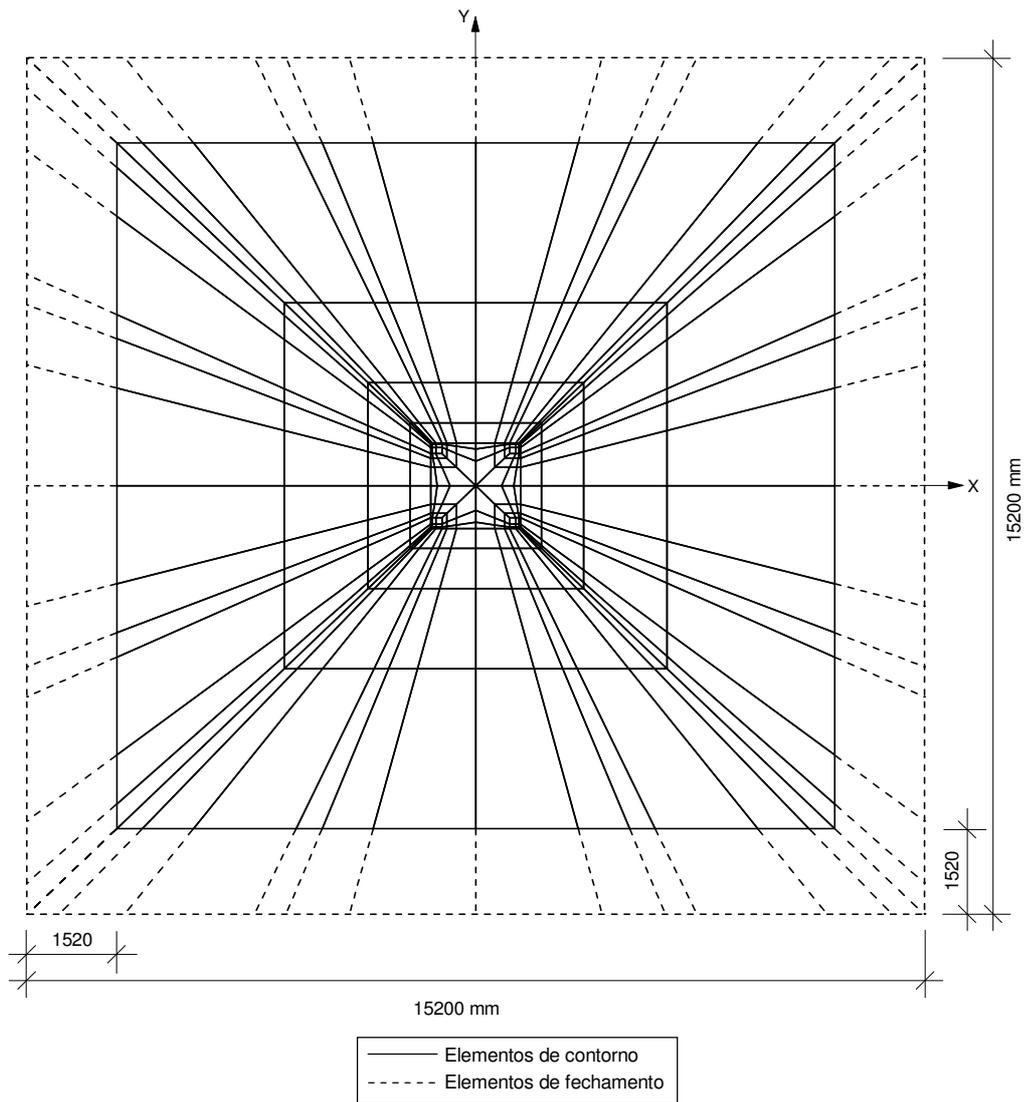


Figura 5.59 – Vista superior da malha: 1 bloco de fundação

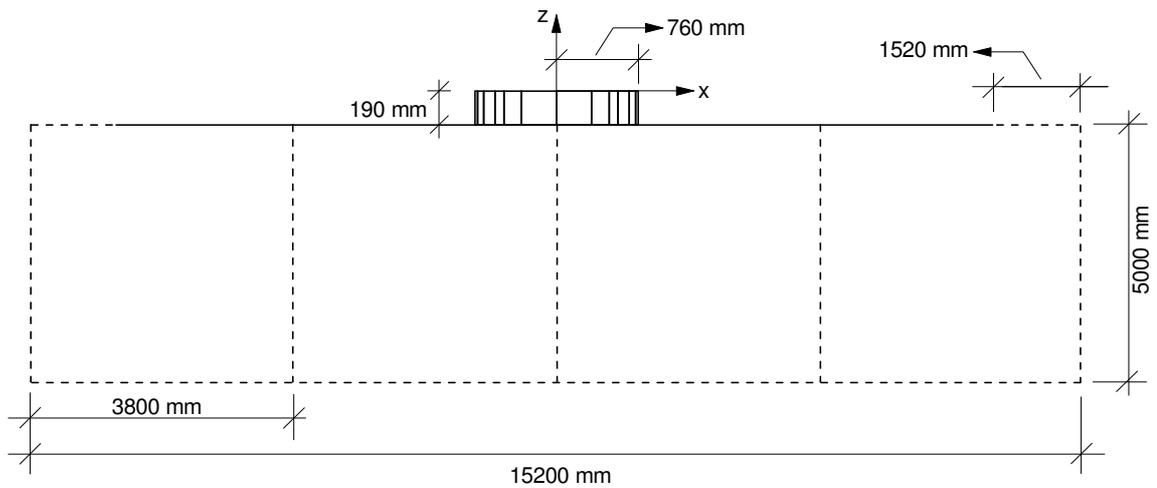


Figura 5.60 – Vista lateral da malha: 1 bloco de fundação

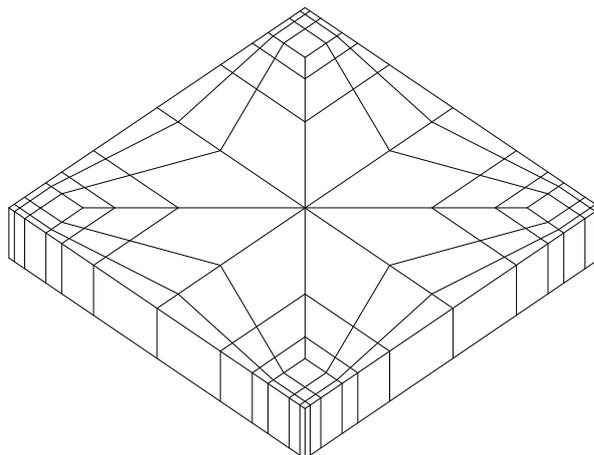


Figura 5.61 – Vista 3D da malha do bloco

O bulbo da tensão principal máxima, σ_1 , para a relação $E_f = 100E_s$ e o perfil de deslocamento vertical no plano de simetria xz para as diversas relações E_f/E_s são mostrados nas Figuras 5.62 e 5.63, respectivamente.

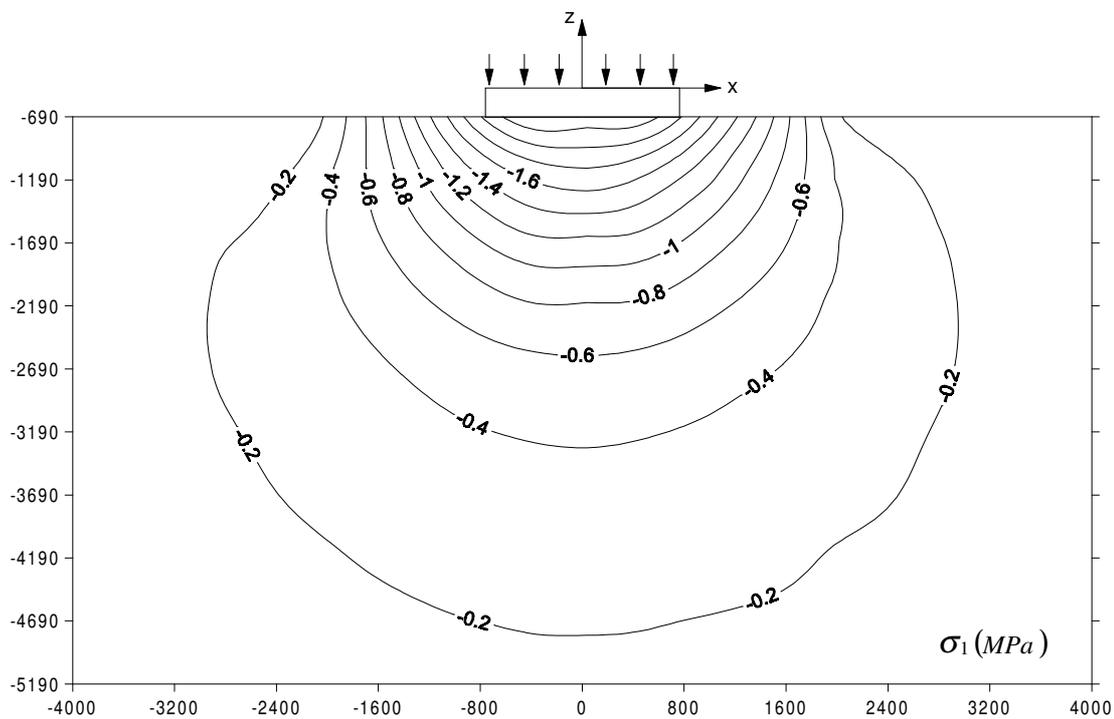


Figura 5.62 – Bulbo de tensões: 1 bloco de fundação

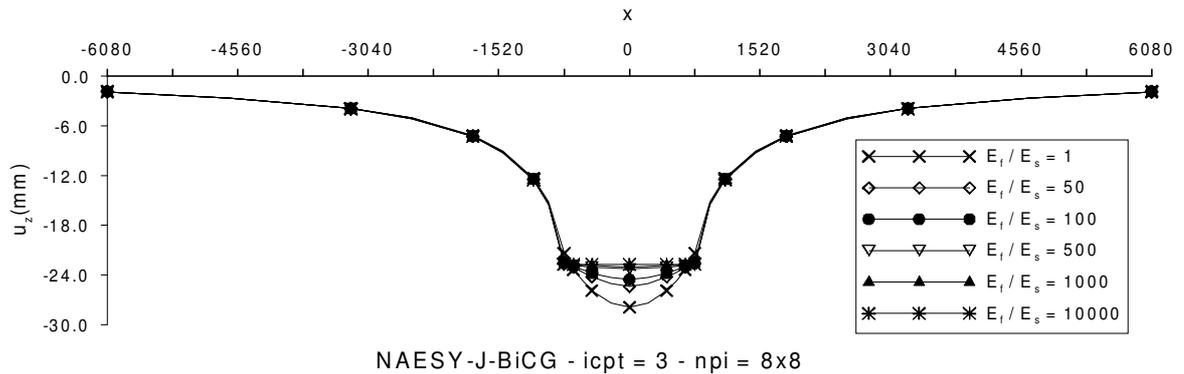


Figura 5.63 – Perfil de deslocamento: 1 bloco de fundação

Observou-se também a variação da resposta de deslocamentos em alguns pontos da base do bloco (Figura 5.64) em função da ordem de integração adotada na análise.

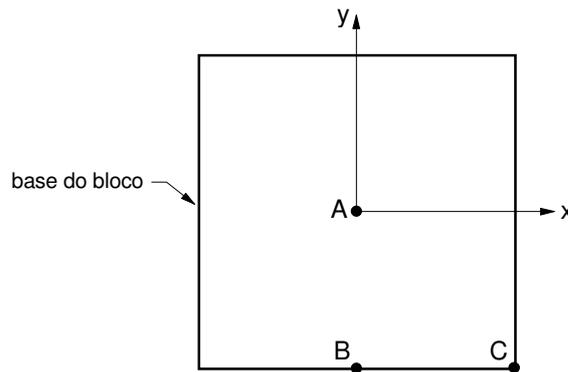


Figura 5.64 – Pontos da base do bloco

Nas Figuras 5.65, 5.66 e 5.67 apresentam-se os resultados obtidos para os *solvers* direto e iterativo. Menciona-se que na resolução deste problema aplicaram-se os processos especiais de integração $icpt = 1$ e $icpt = 3$, todavia não observou-se diferença significativa entre as respostas obtidas utilizando-se esses processos. Deste modo, todos os resultados aqui apresentados referem-se ao processo $icpt = 3$.

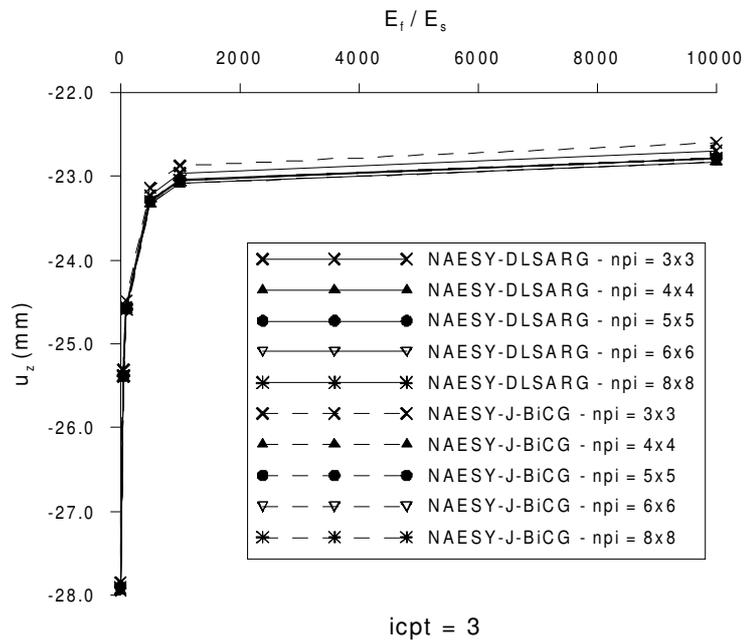


Figura 5.65 – Deslocamento vertical do ponto A

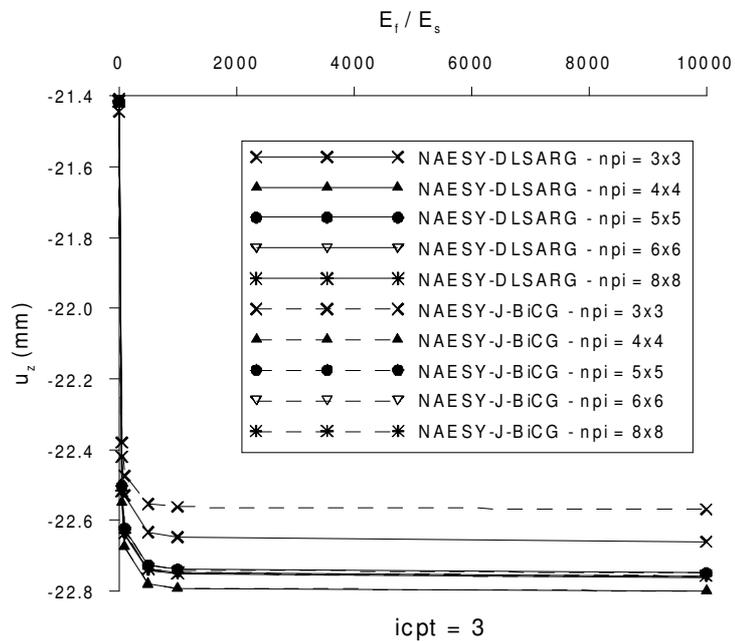


Figura 5.66 – Deslocamento vertical do ponto B

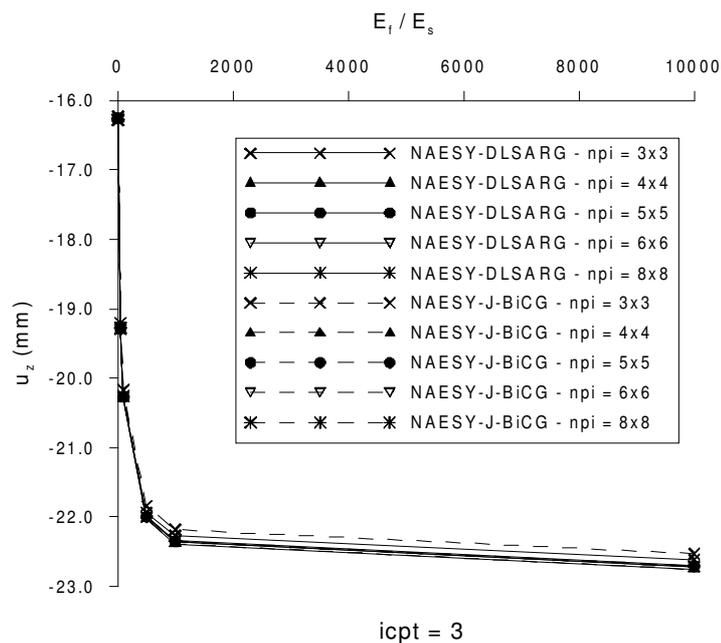


Figura 5.67 – Deslocamento vertical do ponto C

O recalque da fundação rígida ($E_f = 10000E_s$) considerando-se os processos de integração 1 e 3 e o *solver* iterativo J-BiCG é mostrado na Figura 5.68. Comparam-se as respostas obtidas com as soluções semi-analíticas para placas rígidas submetidas a carga uniforme e apoiadas em superfície de espaço semi-infinito fornecidas por Whitman e Richart (1967), e Sieffert e Cevaer (1991).

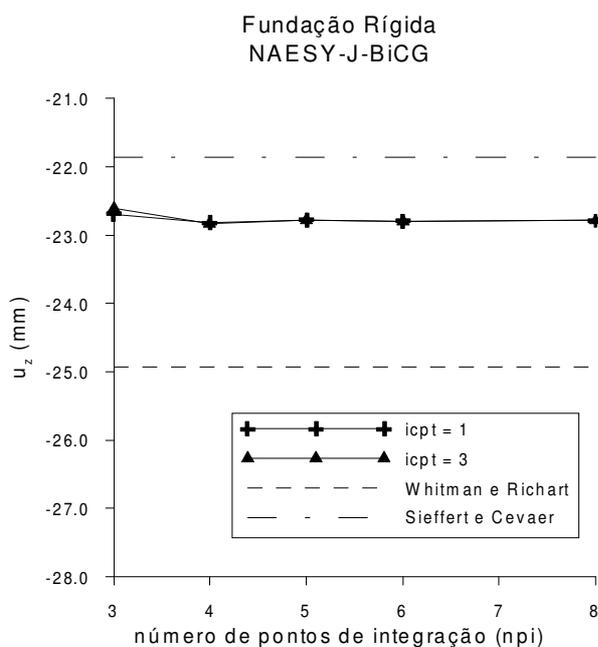


Figura 5.68 – Comparação com as soluções analíticas

A performance do algoritmo de acoplamento foi avaliada a partir da relação entre os tempos de processamento dos *solvers* iterativo e direto, apresentada no gráfico da Figura 5.69, para o processo $icpt = 1$, e da relação entre o número de iterações e a ordem do sistema de equações, apresentada no gráfico da Figura 5.70, para os processos de integração 1 e 3. Na Figura 5.71, plotam-se as medidas tempo de CPU referentes ao tempo de resolução e montagem do sistema de equações para o *solver* direto DLSARG. Diversas ordens de integração e relações de rigidez (E_f / E_s) são consideradas.

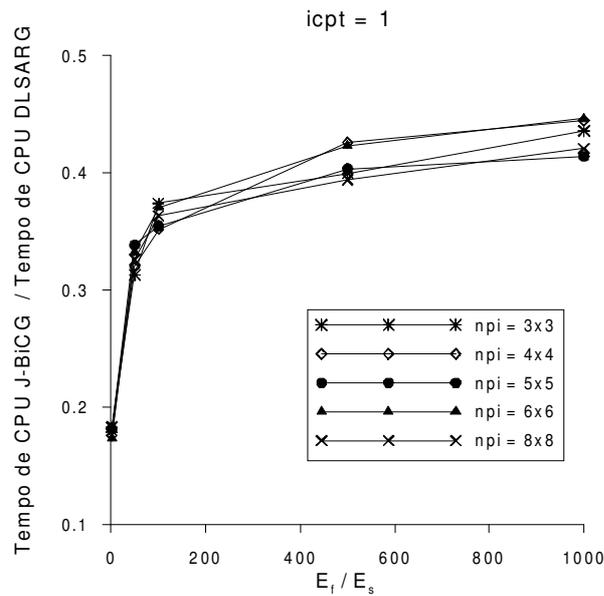


Figura 5.69 – Tempo de CPU dos *solvers*

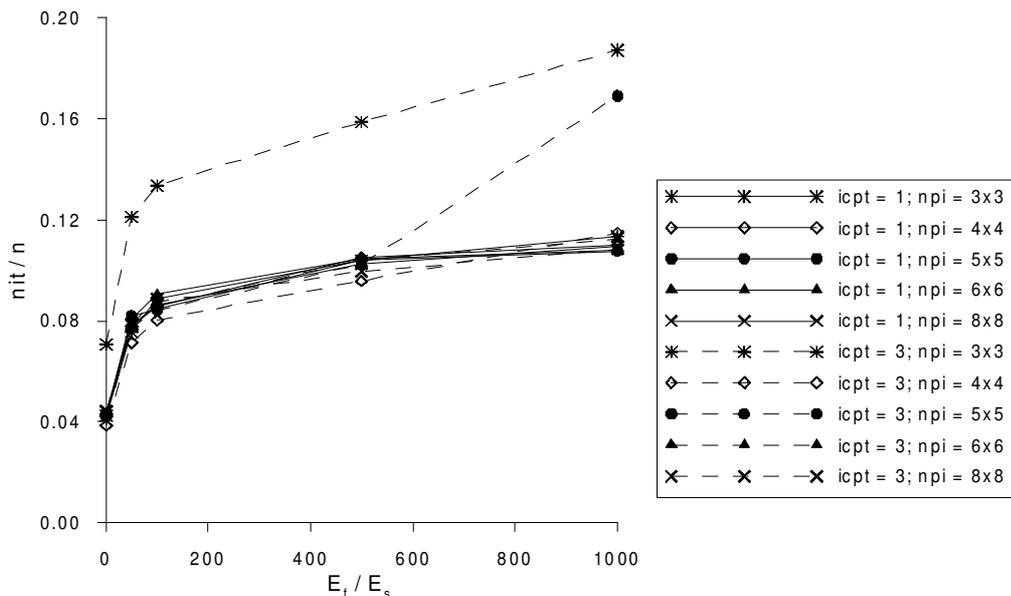


Figura 5.70 – Número de iterações/ordem do sistema: 1 bloco de fundação

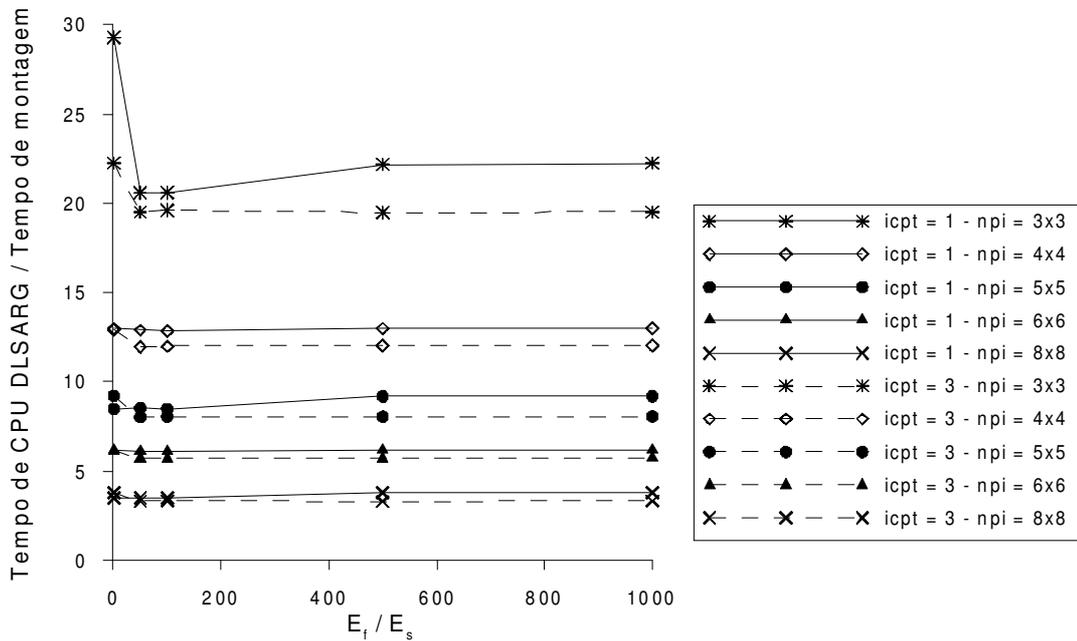


Figura 5.71 – Tempo de CPU DLSARG/Tempo de Montagem: 1 bloco de fundação

Problema 2:

Neste problema, consideram-se dois blocos de fundação acoplados com o solo, o mesmo carregamento e as mesmas propriedades físicas e geométricas do problema anterior (problema 1). Os blocos estão afastados de uma distância de 5000 mm medida a partir de suas faces laterais (Figura 5.73).

Tem-se que o domínio do problema foi subdividido em três subregiões, a saber: duas subregiões para os blocos e uma para o solo. Cada bloco é discretizado com 264 elementos (794 nós), e o solo com 600 elementos (1873 nós), sendo a ordem do sistema resultante igual a $n = 10383$. A respectiva malha de elementos de fechamento (e.e.) possui 128 elementos e 529 nós. As Figuras 5.72 e 5.73 apresentam uma visão geral dessas malhas.

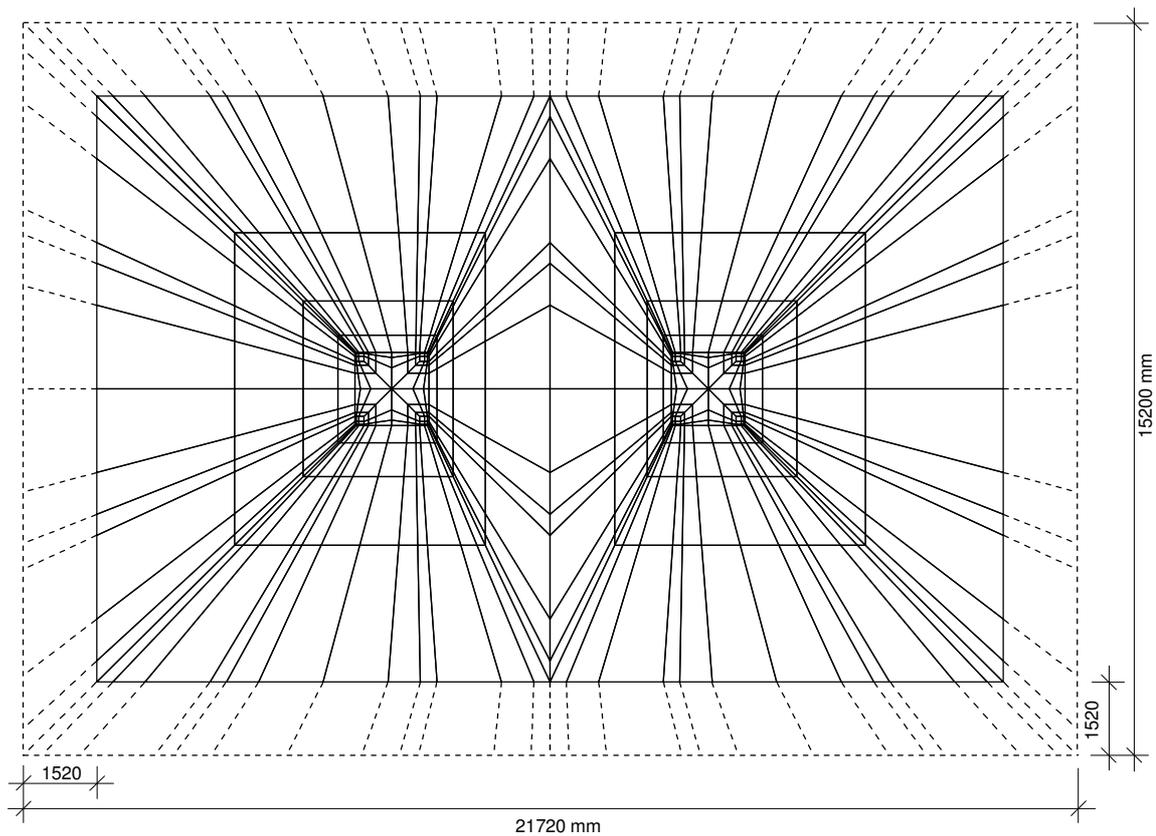


Figura 5.72 – Vista superior da malha: 2 blocos de fundação

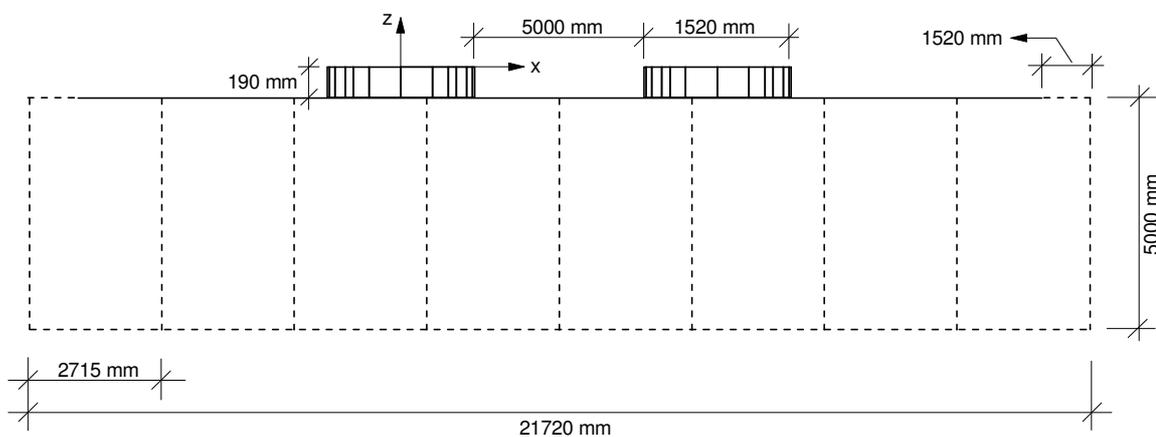


Figura 5.73 – Vista lateral da malha: 2 blocos de fundação

Nas Figuras 5.74 e 5.75 apresentam-se, respectivamente, o bulbo da tensão principal máxima, σ_1 , para a relação $E_f = 100E_s$ e o perfil de deslocamento vertical para as diversas relações E_f / E_s no plano de simetria xz .

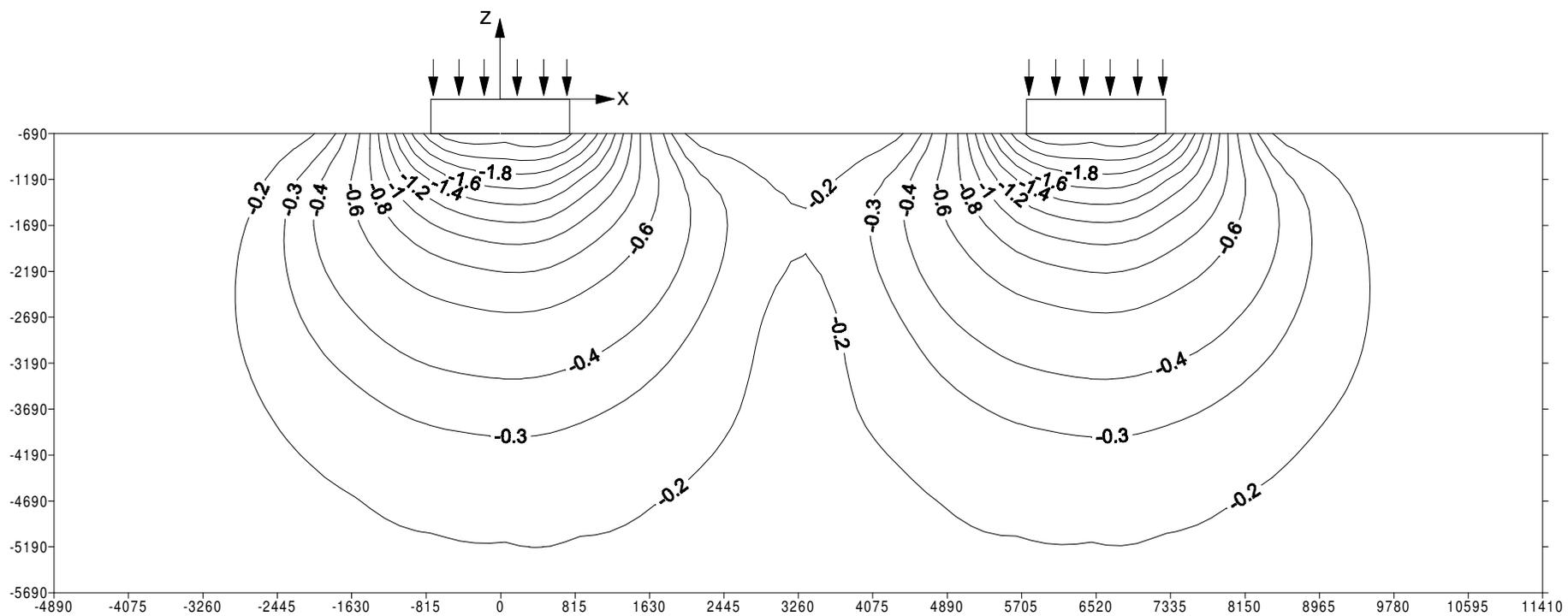


Figura 5.74 – Bulbo de tensões: 2 blocos de fundação

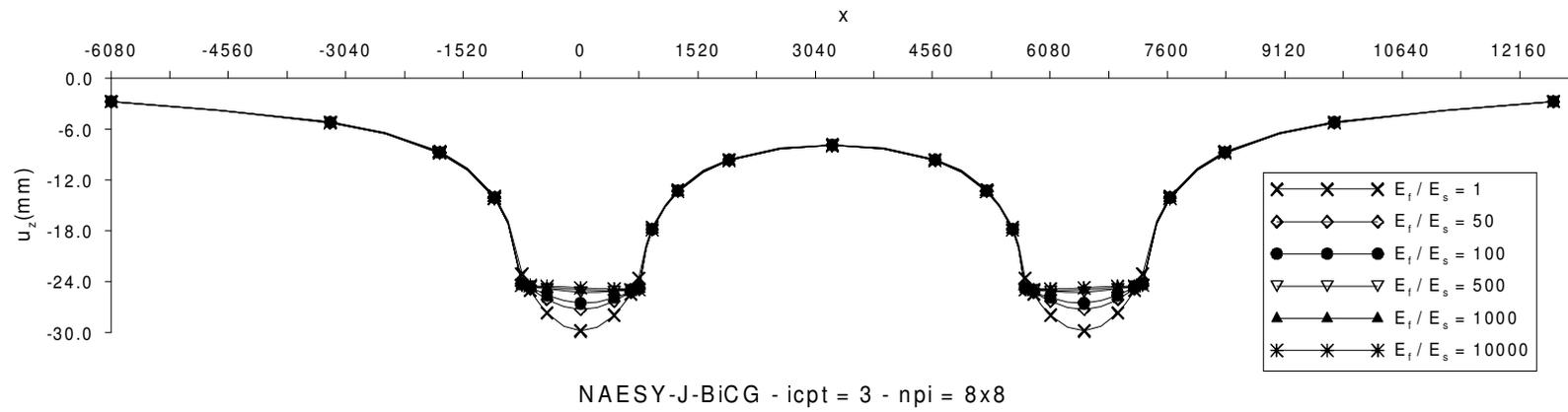


Figura 5.75 – Perfil de deslocamento: 2 blocos de fundação

A relação entre o tempo de resolução e montagem do sistema de equações, bem como o número de iterações do *solver* escalonado pela ordem do sistema, são apresentados nos gráficos abaixo para os processos de integração 1 e 3.

Menciona-se que devido à limitação da rotina IMSL não foi possível, neste problema, em função da dimensão da malha, comparar a performance do *solver* iterativo com o direto.

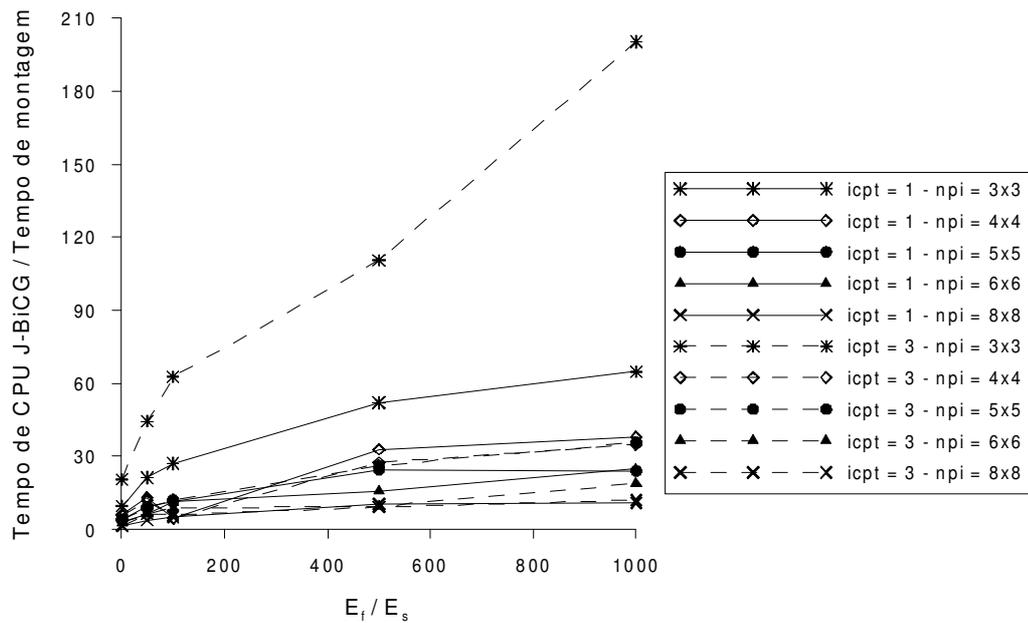


Figura 5.76 – Tempo de CPU J-BiCG/Tempo de montagem: 2 blocos de fundação

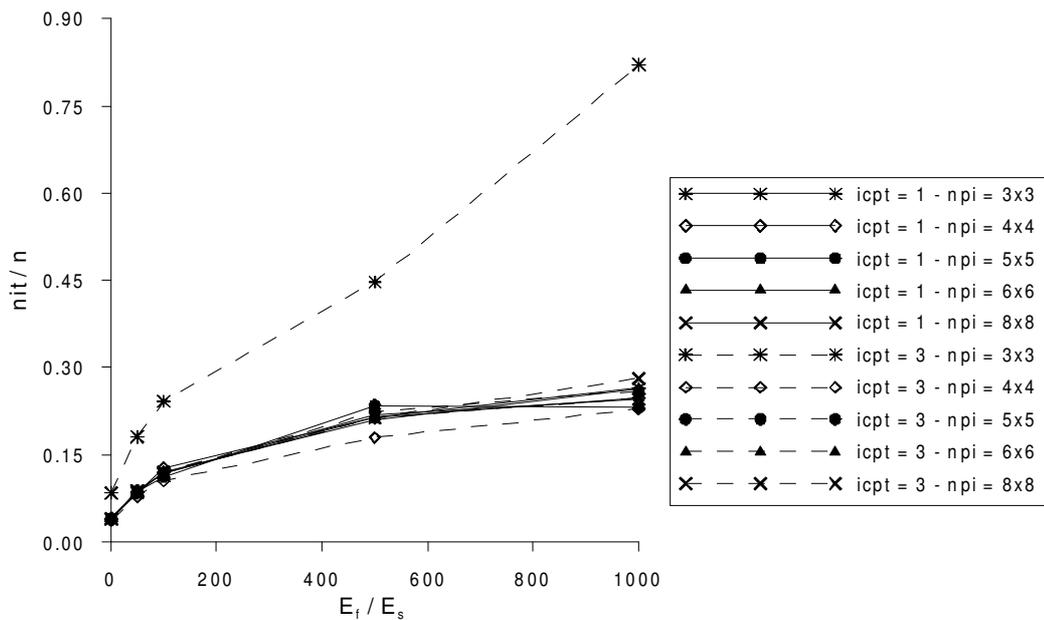


Figura 5.77 – Número de iterações/ordem do sistema: 2 blocos de fundação

CAPÍTULO 6 – CONCLUSÕES

6.1. Conclusões

A partir dos vários exemplos numéricos analisados foi possível validar e avaliar o desempenho das estratégias propostas e implementadas no programa computacional NAESY. No que refere-se à estratégia de acoplamento genérico EC-EC, tem-se que esta tornou-se consideravelmente mais simples com a introdução dos elementos de contorno descontínuos. Na atual versão do programa, a simulação de descontinuidades de fluxo ou força é feita aplicando-se apenas condições de equilíbrio e compatibilidade.

O estudo de performance dos processos de integração apresentado no capítulo 4 (seção 4.5) mostra que o processo combinado, $icpt = 3$, de modo geral, apresentou melhor desempenho na avaliação de integrais singulares e *quasi*-singulares que os outros processos. Considerando-se a avaliação dos termos de **H** e de **G** (Figuras de 4.6 a 4.13), bem como os resultados apresentados para os problemas de condução de calor 3D (Figuras de 5.7 a 5.12) e da barragem com cortinas de estaca-prancha (Figuras de 5.25 a 5.27), observa-se que os processos baseados na transformação de Telles e na transformação de coordenadas polares triangulares também produziram precisão satisfatória. No exemplo 1, têm-se erros inferiores a 0,4% e 1% para, respectivamente, a avaliação do potencial e do fluxo no ponto central de Ω_5 . No exemplo 2, as curvas de erro associados aos processos 1 e 3 são praticamente coincidentes, e o processo 2 apresenta erros inferiores a 0,4%, 8% e 30% para a avaliação do fluxo, respectivamente, nos dois nós do contorno e no ponto central de Ω_3 . Entretanto, utilizando-se o processo $icpt = 3$ pode-se trabalhar com ordens de integração mais aceitáveis ($npi \leq 6$), portanto, neste aspecto, o processo 3 é considerado mais eficiente. Por outro lado, o processo 2 requer menos tempo de CPU para montar o sistema de equações. Analisando-se os gráficos referentes ao tempo de montagem do sistema de equações para os modelos I e II do exemplo 1 (Figuras

5.17 e 5.18), observa-se que o processo $icpt = 2$ requer aproximadamente 80% do tempo de CPU do processo combinado. Nota-se também que não ocorreu diferença considerável entre os tempos de CPU de montagem do processo combinado e do processo $icpt = 1$ (Figuras 5.17, 5.18, 5.71 e 5.76).

A performance do *solver* J-BiCG e do algoritmo de acoplamento genérico foi avaliada em termos do tempo de CPU e do número de iterações (nit). Nos exemplos 1, 2 e 4 observou-se melhor desempenho do *solver* J-BiCG em relação ao *solver* direto DLSARG. Os gráficos das Figuras 5.13 e 5.14, referentes, respectivamente, aos modelos I e II do exemplo 1, mostram que o *solver* J-BiCG é aproximadamente de 5 a 8 vezes mais rápido que o *solver* DLSARG. No exemplo 2, tem-se em função da ordem do sistema de equações que esses valores variam de aproximadamente 1,5 a 3 (Figura 5.28), e, para o exemplo 4, considerando-se as várias relações de rigidez, de 2,5 a 6 (Figura 5.69). Nestes exemplos não se notaram alterações relevantes nas medidas de tempo em função do processo ou da ordem de integração adotados na análise. Para o problema da casca cilíndrica, observa-se que o *solver* J-BiCG não apresentou um bom desempenho com relação ao *solver* DLSARG (Figuras 5.51, 5.52, 5.54 e 5.56). Apenas para as espessuras maiores que $0,025a$, modelo de contorno 2 e $d = 0,1$ ou $d = 0,05$, a relação entre o tempo de CPU do *solver* iterativo e direto foi menor que 1 (Figuras 5.53, 5.55 e 5.57). Esta queda no desempenho do *solver* iterativo se explica pelo fato do domínio do problema ser composto por elementos esbeltos, e também, pelos valores relativamente baixos de deslocamento (d) utilizados na geração dos elementos descontínuos, o que implica no mal condicionamento do sistema. Na verdade, neste exemplo, o sistema de equações é *quasi*-singular. No entanto, observa-se uma tendência de redução no tempo de CPU do *solver* iterativo com o aumento da ordem do sistema de equações (n), portanto pode-se esperar uma melhor eficiência desse *solver* para problemas de ordem elevada.

No exemplo 1, o número de iterações obtidas varia de 8 a 20% da ordem do sistema para o modelo I (ver Figura 5.15) e de 6 a 15% para o modelo II (ver Figura 5.16). Baseado nas Figuras 5.13, 5.14, 5.15 e 5.16 conclui-se que a performance não é muito sensível ao aumento de condutividade relativa. No entanto, para $d = 0,01$, a

performance se deteriorou em comparação com a performance obtida para $d = 0,1$ e $d = 0,05$. De fato, apesar do *solver* convergir, espera-se uma queda na precisão se o número de condicionamento aumenta e a tolerância (*tol*) não seja correspondentemente reduzida. Nas Figuras 5.19 e 5.20 plota-se o número de condicionamento espectral estimado do sistema de equações associados aos modelos I e II em função da relação de rigidez. Para o exemplo 4, tem-se que *nit* varia de 4 a 11% e de 5 a 20% da ordem do sistema para, respectivamente, os problemas 1 e 2 (ver Figuras 5.70 e 5.77) e o aumento da relação de rigidez (E_f / E_s) implicou em uma pequena redução no desempenho do *solver* iterativo (ver Figuras 5.69, 5.70 e 5.77). Por outro lado, a variação do processo e da ordem de integração não influenciou na performance.

Com relação à precisão dos resultados verificou-se uma excelente concordância entre as respostas obtidas através do programa computacional desenvolvido (NAESY) e as soluções analíticas dos problemas ou as respostas obtidas via ANSYS. Por exemplo, nas Figuras 5.3 e 5.5 os resultados do NAESY para respectivamente, os modelos I e II do exemplo 1, são comparados com os do ANSYS, e também observa-se a precisão dos *solvers* J-BiCG e DLSARG. Nas Figuras 5.4 e 5.6 comparam-se os resultados calculados com o programa NAESY utilizando-se, respectivamente, os modelos I e II, diferentes valores de d e diferentes processos de integração, com os resultados do ANSYS. Pode-se observar que a resposta do ANSYS torna-se incorreta para relações de condutividade térmica maiores que $10^{10} k$, quando o sistema torna-se muito mal-condicionado (número de condicionamento $\geq 10^{11}$). Para o problema da barragem com cortinas de estaca-prancha tem-se que a pressão sob a base da barragem obtida via NAESY-J-BiCG praticamente coincidiu com a solução analítica fornecida por Lambe e Whitman (1969) (ver Figura 5.24). No exemplo da casca cilíndrica os gráficos das Figuras 5.32 a 5.50, referentes ao deslocamento na direção y ao longo do eixo z na região da fratura, mostram que a resposta do programa NAESY convergiu para todos os valores de d , processos de integração, modelos de subestruturação e espessuras da casca cilíndrica (h) adotados na análise. Nesses gráficos comparam-se os resultados do NAESY com os obtidos via ANSYS.

Considerando-se o acoplamento entre o solo e apenas uma fundação rígida, exemplo 4 (problema 1), tem-se que a resposta calculada com o programa NAESY está confinada entre as duas soluções semi-analíticas para o recalque da fundação fornecidas na literatura (ver Figura 5.68). Nota-se também, que o bulbo de tensão principal σ_1 e o perfil de deslocamento no plano de simetria xz têm os aspectos esperados (Figuras 5.62 e 5.63), e que a convergência da resposta já é verificada para $npi = 4 \times 4$ em ambos os processos de integração testados (Figuras de 5.65 a 5.68). Para o problema de interação do solo com dois blocos de fundação não há comparação com soluções semi-analíticas, mas devido à boa precisão observada em análises anteriores e aos aspectos do bulbo de tensão principal σ_1 (Figura 5.74) e do perfil de deslocamento (Figura 5.75) encontrados, considera-se que esses resultados sejam consistentes. Menciona-se que nesse problema não foi possível resolver o correspondente sistema de equações com o *solver* IMSL-DLSARG uma vez que a ordem do sistema ($n = 10383$) excedia o valor máximo permitido. Tampouco pôde-se resolver os problemas do exemplo 4 utilizando-se o pacote comercial ANSYS 6.0, pois não se observou convergência da resposta para as malhas testadas e não foi possível refinar a malha já que o número máximo de nós da versão disponível foi atingindo ($nno = 32000$).

O aumento de eficiência computacional de algoritmos de elementos de contorno, bem como sua aplicação a problemas de ordem elevada talvez seja o principal objetivo do algoritmo de acoplamento proposto. Particularmente em relação ao espaço de memória necessário para realização das análises, menciona-se que a esparsidade das matrizes globais para os modelos I e II do exemplo 1 é aproximadamente 78%, para os modelos adotados no exemplo 2, fica em torno de 60%, e para os modelos dos problemas 1 e 2 do exemplo 4, aproximadamente 28% e 48%, respectivamente. Portanto, a economia de memória é significativa. Além disso, o *solver* J-BiCG mostrou-se ser robusto, convergindo mesmo para sistemas muito mal-condicionados, como por exemplo, aqueles associados a condutividades elevadas ou rigidez das fundações ou deslocamento muito pequenos nos elementos descontínuos.

6.2. Aspectos Futuros

A seguir citam-se alguns tópicos importantes para abordagem em trabalhos futuros:

- Expansão da estrutura de acoplamento desenvolvida na pesquisa a outros módulos de análise do programa computacional;
- Implementação, direta, em plataformas de análise computacional paralela;
- Implementação de um arranjo qualquer de fundações rígidas (modelos próprios);
- Acoplamento solo-estruturas aporticadas espaciais;
- Estruturação das operações matriz-vetor presentes no *solver* iterativo, de modo que os comandos condicionais sejam eliminados;
- Inclusão de outros *solvers* de Krylov na estratégia de acoplamento;
- Melhoramento do esquema de integração para elementos esbeltos segundo estratégia de transformação baseada na abordagem da integral de linha (Liu *et al.* 1993, Krishnasamy *et al.*, 1994, Liu, 1998);
- Simulação de compósitos.

REFERÊNCIAS BIBLIOGRÁFICAS

Alberto, D. R. (2002). *Análise transiente tridimensional via formulação direta do Método dos Elementos de Contorno com opção de subestruturação – Aplicação a problemas de difusão e acústicos*. Dissertação de Mestrado, Dept^o Engenharia Civil, Universidade Federal de Ouro Preto.

Araújo, F. C. (1989). *Técnicas iterativas para resolução de sistemas de equações lineares originados do método dos elementos de contorno*. Dissertação de Mestrado, COPPE, Universidade Federal do Rio de Janeiro, Brasil.

Araújo, F. C. (1994). *Zeitbereichslösung linearer dreidimensionaler probleme der elastodynamik mit einer gekoppelten BE/FE–Methode*. Ph.D. Thesis, Institut für Angewandte Mechanik, Technische Universität Braunschweig.

Araújo, F. C. e Martins, C. J. (2001). *A study of efficiency of multizone BE/BE coupling algorithms based on iterative solvers – applications to 3D time harmonic problems*. In M. Denda, M. H. Aliabadi & A. Charafi (eds.), Proc. Joint Meeting of Boundary Element Techniques and International Association for Boundary Elements. New Brunswick–NJ–USA, 16 – 18 July 2001. Advances in Boundary Element Techniques II. Geneva: Hoggar, v.1, p. 21 – 29

Araújo, F. C.; Dors, C. e Mansur, W. J. (2003). *Dynamic cross–interaction between footings – 3D frequency–domain parallelized analysis via the BEM*. In: Proc. International Conference on Boundary Element Techniques IV, Granada, 2003. Queen Mary, University of London, London, v. 1, p. 253 – 258.

Araújo, F. C.; Dors, C.; Martins, C. J. e Mansur, W. J. (2002). *Analysis of 3D time–harmonic soil–foundation interaction problems by using efficient BE substructuring algorithms*. In: Proc. European Conference on Dynamics, Munich, 2002. A. A. Balkema Publishers: Rotterdam, v. 2, p. 1267 – 1272.

Araújo, F. C.; Dors, C.; Martins, C. J. e Mansur, W.J. (2004). *New developments on BE/BE multi-zone algorithms based on Krylov solvers – Applications to 3D frequency-dependent problems*. J. Braz. Soc. Mech. Sc. & Engrg., v. 26, p. 231 – 248.

Araújo, F. C.; Martins, C. J. e Mansur, W. J. (2001). *An efficient BE iterative-solver-based substructuring algorithm for 3D time-harmonic problems in elastodynamics*. Engineering Analysis with Boundary Elements, v. 25, p. 795 – 803.

Axelsson, O. (1994). *Iterative solution methods*, Cambridge University Press, New York.

Barra, L. P. S.; Coutinho, A. L. G. A.; Mansur, W. J. e Telles, J. C. F. (1992). *Iterative solution of BEM equations by GMRES algorithm*. Computers & Structures, v. 44, p. 1249 – 1253.

Barra, L. P. S.; Coutinho, A. L. G. A.; Telles, J. C. F. e Mansur, W. J. (1993). *Multi-level hierarchical preconditioners for boundary element systems*. Engng. Anal Boundary Elements, v. 12, p. 103 – 109.

Bathe, K. J. (1996). *Finite Element Procedures in Eng. Analysis*. Prentice-Hall, Inc., New Jersey.

Bonnet, M. (1999). *Boundary Integral Equation Methods for Fluids and Solids*. J. Wiley & Sons.

Brebbia, C. A.; Telles, J. C. F. e Wrobel, L. C. (1984). *Boundary element techniques: theory and applications in engineering*. Springer-Verlag, Berlin.

Cruse, T. A. (1996). *BIE fracture mechanics analysis 25 years of developments*. Comput. Mech., v. 18, p. 1 – 11.

Danson, D. (1982). *BEASY Boundary Element Analysis System. MANUAL*: Southampton, UK.

Dors, C. (2002). *Paralelização de algoritmos MEC via subestruturação baseada em solvers iterativos – Aplicação a problemas 3D escalares e vetoriais*. Dissertação de Mestrado, Dept^o Engenharia Civil, Universidade Federal de Ouro Preto.

Gray, L. J.; Martha, L. F. e Ingrassia, A. R. (1990). *Hypersingular integrals in boundary element fracture analysis*. Int. J. Numer. Methods Engrg., v. 29, p. 1135 – 1158.

Hageman, L. A. e Young, D. M. (1981). *Applied iterative methods*. Academic Press, Inc.

Hsiao, G. C.; Steinbach, O. e Wendland, W. L. (2000). *Domain decomposition methods via boundary integral equations*. Journal of Computational and Applied Mathematics, v. 125, p. 521 – 537.

Jun, L.; Beer, G. e Meek, J. L. (1985). *Efficient evaluation of integrals of order $1/r$, $1/r^2$, $1/r^3$ using Gauss quadrature*. Engineering Analysis with Boundary Elements, v. 2, p. 118 – 123.

Kane, J. H.; Kumar, B. L. e Saigal, S. (1990). *An arbitrary condensing, noncondensing solution strategy for large scale, multi-zone boundary element analysis*. Comput. Meth. Appl. Mech. Engrg., v. 79, p. 219 – 244.

Krishnasamy, G.; Rizzo, F. J. e Liu, Y. J. (1994). *Boundary integral equation for thin bodies*. Int. J. Numer. Methods Engrg., v. 37, p. 107 – 121.

Lachat, J. C. e Watson, J. O. (1976). *Effective numerical treatment of boundary integral equations: a formulation for three-dimensional elastostatics*. Int. J. Numer. Methods Engrg., v. 10, p. 991 – 1005.

Lambe, T. W. e Whitman, R. V. (1969). *Soil mechanics*. J. Wylie, New York.

Li, H. B.; Han, G. M. e Mang, H. A. (1985). *A new method for evaluating singular integrals in stress analysis of solids by the direct boundary element method*. Int. J. Numer. Methods Engrg., v. 21, p. 2071 – 2098.

Liu, Y. (1998). *Analysis of shell-like structures by the boundary element method based on 3-D elasticity: formulation and verification*. Int. J. Numer. Methods Engrg., v. 41, p. 41 – 558.

Liu, Y. J.; Zhang, D. e Rizzo, F. J. (1993). *Nearly singular and hypersingular integrals in the boundary element method*. In: C. A. Brebbia and J. J. Rencis (eds.), *Boundary Elements XV*, Computational Mechanics Publications, p. 453 – 468, Worcester, MA.

Mang, H.; Li, H. e Han, G. (1985). *A new method for evaluating singular integrals in stress analysis of solids by the direct BEM*. Int. J. Num. Methods in Engineering, v. 21, p. 2071 – 2098.

Mansur, W. J.; Araújo, F. C. e Malaghini, J. E. B. (1992). *Solution of BEM systems of equations via iterative techniques*. Int. J. Numer. Methods Engrg., v. 33, p. 1823 – 1841.

Martins, C. J. (2000). *Análise de problemas 3D no domínio da frequência via processo de acoplamento multidomínio BE/BE*. Dissertação de Mestrado, Dept^o Engenharia Civil, Universidade Federal de Ouro Preto.

Saad, Y. (1996). *Iterative methods for sparse linear systems*. PWS Publishing, New York.

Sieffert, J. G. e Cevaer, F. (1991). *Handbook of impedance functions*. Surface Foundations. Quest Editions.

Sonneveld, P. (1989). *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*. SIAM – J. Sci. Statist. Comput., v. 10, p. 36 – 52.

Telles, J. C. F e Oliveira, R. F. (1994). *Third degree polynomial transformation for boundary element integrals: further improvements*. Engineering Analysis with Boundary Elements, v. 13, p. 135 – 141.

Telles, J. C. F. (1987). *A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals*. Int. J. Numer. Methods Engrg., v. 24, p. 959 – 973.

van der Vorst, H. A. (1992). *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*. SIAM – J. Sci. Statist. Comput., v. 13, p. 631 – 644.

van der Vorst, H. A. (2002). *Efficient and reliable iterative methods for linear systems*. J. Comp. & Appl. Mathematics, v. 149, p. 251 – 256.

van der Vorst, H. A. (2003). *Iterative Krylov methods for large linear systems*. Cambridge University Press.

Voutsinas, S. e Bergeles, G. (1990). *Numerical calculation of singular integrals appearing in three-dimensional potential flow problems*. Appl. Math. Modeling, v. 14, p. 618 – 629.

Whitman, R. V. e Richart, F. E. (1967). *Design procedures for dynamically loaded foundations*. JSMFD-ASCE; v. 93, p. 169 – 193.

Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford.

Wrobel, L. C. e Aliabadi, M. H. (2002). *The boundary element method*. J. Wiley & Sons, vols. I & II.