ESTUDO SOBRE O MÉTODO DOS RESÍDUOS PONDERADOS E SUA APLICAÇÃO EM PROBLEMAS DE ENGENHARIA MECÂNICA

André Luiz Amancio [1], Rodrigo Heyse Polidoro,[2] Eduardo Lenz Cardoso

[1] Acadêmico(a) do Curso de Engenharia Mecânica/CCT - PROBIC/UDESC

[2] Acadêmico do Curso de Engenharia Mecãnica/CCT

[3] Orientador, Departamento de Engenharia Mecânica/CCT - eduardo.cardoso@udesc.br

Palavras-chave: Equações diferenciais, resíduos ponderados, métodos numéricos

O estudo realizado neste projeto consiste na utilização de uma plataforma computacional para solucionar equações diferenciais, com o objetivo de resolver numericamente problemas referentes à área da mecânica dos sólidos. A plataforma utilizada foi a FEniCS, utilizando a

linguagem de programação Python para a implementação dos códigos. Inicialmente foram estudados alguns fundamentos da mecânica dos sólidos, além dos métodos de elementos finitos e dos resíduos ponderados, utilizados pelo FEniCS para a resolução dos problemas.

O método dos resíduos ponderados consiste em determinar uma solução aproximada para uma dada equação diferencial. Se essa solução aproximada não satisfaz a equação em todos os pontos, é então gerado um resíduo decorrente do erro de aproximação. Portanto, quanto menor o resíduo, melhor a aproximação. Isto significa que se a função de aproximação for igual à função que satisfaz a equação diferencial em todos os pontos, o resíduo será igual à zero.

O FEniCS Project é uma biblioteca de funções do Python, que permite descrever a equação diferencial em sua forma fraca, bem como a definição do domínio do problema e das condições de contorno. Com isto, é possível descrever matematicamente o problema de forma simples, automatizando a implementação computacional do método. A seguir será ilustrada a sequência de comandos necessária para a solução de um problema de elasticidade tridimensional, cuja forma fraca é dada por :

$$\int_\Omega \boldsymbol{\varepsilon}(\mathbf{u}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{w}) d\Omega = \int_\Omega \mathbf{f}.\mathbf{w} d\Omega$$

As etapas e comandos utilizados para a solução de um problema de elasticidade 3D são:

- Informações a respeito da geometria (domínio de integração) e criação da malha (discretização do domínio de integração). Essa discretização implica diretamente na qualidade da resposta obtida nos cálculos realizados posteriormente

mesh = BoxMesh(Point(0, 0, 0), Point(L, W, W), 40, 12, 12)

- Operadores diferenciais associados a forma fraca da equação diferencial

def epsilon(u):return 0.5(grad(u) + grad(u).T)*

- Propriedades do material (relação constitutiva)

*def sigma(u): return lambda_*tr(epsilon(u))*Identity(len(u)) + 2.0*mu*epsilon(u)*

- Espaços envolvidos na definição da função teste e da função aproximada

V = VectorFunctionSpace(mesh, 'CG', 1); u = TrialFunction(V); w = TestFunction(V)

- Condições de contorno essenciais (apoios, no caso mecânico) e naturais (solicitações, no caso mecânico)

def clamped_boundary(x, on_boundary): return on_boundary and (abs(x[0]-0.0) < tol)

bc = DirichletBC(V, Constant((0, 0, 0)), clamped_boundary)

- Definição da forma fraca da equação diferencial

*a = inner(sigma(u), epsilon(w))*dx; b = dot(f, w)*dx*

- Solução do problema, obtendo o campo de deslocamentos U

U = Function(V); solve(a==b, U,bc)

- Pós-processamento e interpretação dos resultados.