Universidade Estadual de Campinas
Faculdade de Engenharia Mecânica
Departamento de Mecânica Computacional

Método Newmark

Trabalho da disciplina de pós - graduação IM566A, referente a atividade 05, enviado na data 29 de setembro de 2025.

Discente: Me. Wanderson Vinicius de Oliveira Monteiro
Docente: Prof. Dr. Alberto Luiz Serpa

Campinas - SP
29 de setembro de 2025

SUMÁRIO

1 Método de Newmark

1.1 Problema

Implementar o método de Newmark. Testar com o exemplo 8.4 do livro do Bathe & Wilson.

1.2 Referencial teórico

O método de Newmark é uma extensão do método linear, assumindo a velocidade e deslocamento da forma:

$$\dot{\mathbf{U}}_{t+\Delta t} = \dot{\mathbf{U}}_t + \left[(1-\delta)\ddot{\mathbf{U}}_t + \delta\ddot{\mathbf{U}}_{t+\Delta t}\right]\Delta t \tag{1.1}$$

$$\mathbf{U}_{t+\Delta t} = \mathbf{U}_t + \dot{\mathbf{U}}_t\Delta t + \left[\left(\frac{1}{2}-\alpha\right)\ddot{\mathbf{U}}_{\mathbf{t}} + \alpha\ddot{\mathbf{U}}_{t+\Delta t}\right]\Delta t^2 \tag{1.2}$$

com α e δ parâmetros que são escolhidos para obter a precisão e a estabilidade do algoritmo. se $\delta = \frac{1}{2}$ e $\alpha = \frac{1}{6}$ se torna o método da aceleração linear (Wilson - θ), com $\theta = 1$. Para $\delta = \frac{1}{2}$ e $\alpha = \frac{1}{4}$ se torna o método da aceleração média (Figura 1.1)

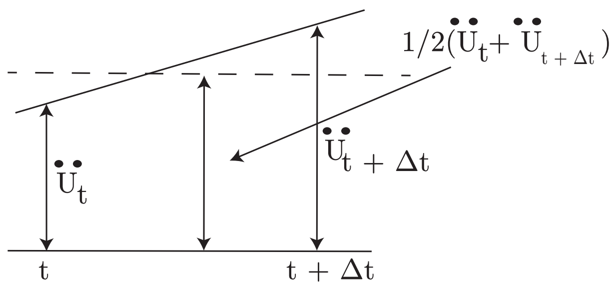


Figura 1.1: Newmark - Aceleração constante.

A Equação de equilíbrio para $t+\Delta t$ é dada por:

$$[\mathbf{M}]\,\ddot{\mathbf{u}}_{t+\Delta t} + [\mathbf{C}]\dot{\mathbf{u}}_{t+\Delta t} + [\mathbf{K}]\mathbf{u}_{t+\Delta t} = \mathbf{R}_{t+\Delta t} \tag{1.3}$$

Da equação (1.2) temos:

$$\begin{aligned} \mathbf{u}_{t+\Delta t} - \mathbf{u}_t - \dot{\mathbf{u}}_{t+\Delta t} &= \left[\left(\frac{1}{2}-\alpha\right)\ddot{\mathbf{u}}_t + \alpha\ddot{\mathbf{u}}_{t+\Delta t}\right]\Delta t^2 \\ \ddot{\mathbf{u}}_{t+\Delta t} &= \frac{1}{\alpha}\left[\mathbf{u}_{t+\Delta t}\frac{1}{\Delta t^2}(\mathbf{u}_{t+\Delta t} - \mathbf{u}_t) - \left(\frac{1}{2}-\alpha\right)\ddot{\mathbf{u}}_t\right] \end{aligned} \tag{1.4}$$

Substituindo a equação (1.4) na equação (1.1):

$$\dot{\mathbf{u}}_{t+\Delta t} = \left(1-\frac{\delta}{\alpha}\right)\dot{\mathbf{u}}_t + \left(1-\frac{\delta}{2\alpha}\right)\Delta t\ddot{\mathbf{u}}_t + \frac{\delta}{\alpha\Delta t}\mathbf{u}_{t+\Delta t} - \frac{\delta}{\alpha\Delta t}\mathbf{u}_t \tag{1.5}$$

Retornando para a equação de movimento, temos:

$$
\begin{aligned}
\Big([M]\tfrac{1}{\alpha\Delta t^2} + [C]\tfrac{\delta}{\alpha\Delta t} + [K]\Big)\mathbf{u}_{t+\Delta t} &= \mathbf{R}_{t+\Delta t} \\
&+ [M]\Big(\tfrac{1}{\alpha\Delta t^2}\mathbf{u}_t + \tfrac{1}{\alpha\Delta t}\dot{\mathbf{u}}_t + (\tfrac{1}{2\alpha} - 1)\ddot{\mathbf{u}}_t\Big) \\
&+ [C]\Big(\tfrac{\delta}{\alpha\Delta t}\mathbf{u}_t + (\tfrac{\delta}{\alpha} - 1)\dot{\mathbf{u}}_t + \tfrac{\Delta t}{2}(\tfrac{\delta}{\alpha} - 2)\ddot{\mathbf{u}}_t\Big)
\end{aligned}
\tag{1.6}
$$

1.3 Pseudo-código do Método de Newmark

Algorithm 1 Método de Newmark

Iniciar: Matrizes $[\mathbf{K}], [\mathbf{M}], [\mathbf{C}]$, vetor de carga $\mathbf{R}(t)$, deslocamento inicial $\mathbf{u}_0$, velocidade inicial $\mathbf{v}_0$, aceleração inicial $\mathbf{a}_0$, passo de tempo Δt, tempo inicial t_0, tempo final t_f, parâmetros de Newmark δ, α

Criar vetor de tempo: $t = t_0 : \Delta t : t_f$

Calcular constantes:

$$\begin{aligned}
&a_0 = \frac{1}{\alpha \Delta t^2}, \quad a_1 = \frac{\delta}{\alpha \Delta t}, \quad a_2 = \frac{1}{\alpha \Delta t}, \\
&a_3 = \frac{1}{2\alpha} - 1, \quad a_4 = \frac{\delta}{\alpha} - 1, \quad a_5 = \frac{\Delta t}{2}\left(\frac{\delta}{\alpha} - 2\right), \\
&a_6 = \Delta t(1-\delta), \quad a_7 = \delta \Delta t
\end{aligned}$$

Calcular matriz de rigidez efetiva:

$$[\mathbf{K}_e] = [\mathbf{K}] + a_0[\mathbf{M}] + a_1[\mathbf{C}]$$

Inicialização:

$$\mathbf{u}_t(:,1) = \mathbf{u}_0, \quad \mathbf{v}_t(:,1) = \mathbf{v}_0, \quad \mathbf{a}_t(:,1) = \mathbf{a}_0$$

for $i = 1$ **to** $N_{\text{passos}} - 1$ **do**

Calcular força efetiva:

$$\begin{aligned}
\mathbf{R}_{\text{eff}}(:,i+1) = \mathbf{R}(:,i+1) &\\
&+ [\mathbf{M}]\left(a_0\mathbf{u}_t(:,i) + a_2\mathbf{v}_t(:,i) + a_3\mathbf{a}_t(:,i)\right) \\
&+ [\mathbf{C}]\left(a_1\mathbf{u}_t(:,i) + a_4\mathbf{v}_t(:,i) + a_5\mathbf{a}_t(:,i)\right)
\end{aligned}$$

Resolver deslocamento:

$$\mathbf{u}_t(:,i+1) = [\mathbf{K}_e]^{-1}\mathbf{R}_{\text{eff}}(:,i+1)$$

Atualizar aceleração:

$$\mathbf{a}_t(:,i+1) = a_0\left(\mathbf{u}_t(:,i+1) - \mathbf{u}_t(:,i)\right) - a_2\mathbf{v}_t(:,i) - a_3\mathbf{a}_t(:,i)$$

Atualizar velocidade:

$$\mathbf{v}_t(:,i+1) = \mathbf{v}_t(:,i) + a_6\mathbf{a}_t(:,i) + a_7\mathbf{a}_t(:,i+1)$$

end for

Resultado: Retornar $\mathbf{u}_t, \mathbf{v}_t, \mathbf{a}_t$

1.4 Resultados

Calcule a resposta do deslocamento do sistema considerado nos Exemplos 8.1 ao 8.3 para o método de Newmark. Use $\alpha = 0.25$ e $\delta = 0.5$.

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{U}_1 \\ \ddot{U}_2 \end{Bmatrix} + \begin{bmatrix} 6 & -2 \\ -2 & 4 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 10 \end{Bmatrix} \tag{1.7}$$

Considerando o primeiro caso para $\Delta t = 0.28\ s$, temos para os primeiros 12 passos; a comparação entre os resultados obtidos pelo script em MATLAB e pela referência (BATHE; WILSON, 1976).

Tabela 1.1: Comparação entre os resultados $\Delta t = T2/10$.

Time	Δt	$2\Delta t$	$3\Delta t$	$4\Delta t$	$5\Delta t$	$6\Delta t$	$7\Delta t$	$8\Delta t$	$9\Delta t$	$10\Delta t$	$11\Delta t$	$12\Delta t$
$\mathbf{U_t}$ - Bathe	0.00673	0.0504	0.189	0.485	0.961	1.58	2.23	2.76	3.00	2.85	2.28	1.40
	0.364	1.35	2.68	4.00	4.95	5.34	5.13	4.48	3.64	2.90	2.44	2.30
$\mathbf{U_t}$ - Wand.	0.0067	0.0504	0.1894	0.4846	0.9613	1.5805	2.2328	2.7607	3.0035	2.8505	2.2840	1.3968
	0.3637	1.3510	2.6833	3.9954	4.9497	5.3366	5.1296	4.4781	3.6424	2.8967	2.4352	2.3129

Observa-se que a Tabela 1.1 apresenta concordância entre o código implementado e a referência, com diferenças apenas na quarta casa decimal. Essa proximidade confirma a consistência da implementação numérica.

Agora, considerando um $\Delta t = 28\ s$, temos:

Tabela 1.2: Comparação entre os resultados $\Delta t = T2 \cdot 10$.

Time	Δt	$2\Delta t$	$3\Delta t$	$4\Delta t$	$5\Delta t$	$6\Delta t$	$7\Delta t$	$8\Delta t$	$9\Delta t$	$10\Delta t$	$11\Delta t$	$12\Delta t$
$\mathbf{U_t}$ - Bathe	1.99	0.028	1.94	0.112	1.82	0.248	1.67	0.429	1.47	0.648	1.23	0.894
	5.99	0.045	5.99	0.177	5.72	0.393	5.47	0.685	5.14	1.04	4.76	1.45
$\mathbf{U_t}$ - Wand.	1.9929	0.0284	1.9364	0.1124	1.8259	0.2480	1.6666	0.4293	1.4655	0.6478	1.2320	0.8937
	5.9888	0.0447	5.8998	0.1773	5.7248	0.3931	5.4700	0.6847	5.1441	1.0420	4.7584	1.4529

Novamente, para um $\Delta t = 28.0\ s$, temos a concordância entre os resultados da referência e a rotina implementada em MATLAB.

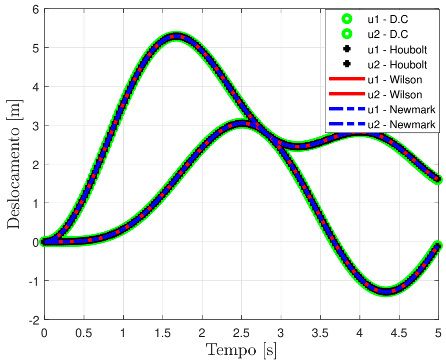
Alterando as condições iniciais, fazendo $\ddot{\mathbf{U}}_0 = \{0\ 0\}^T$, temos os resultados:

Tabela 1.3: Comparação entre os resultados $\Delta t = T2 \cdot 10$.

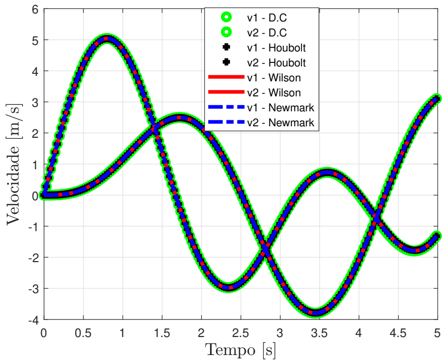
Time	Δt	$2\Delta t$	$3\Delta t$	$4\Delta t$	$5\Delta t$	$6\Delta t$	$7\Delta t$	$8\Delta t$	$9\Delta t$	$10\Delta t$	$11\Delta t$	$12\Delta t$
$\mathbf{U_t}$ - Bathe	0.363	1.44	0.632	1.29	0.782	1.17	0.875	1.09	0.929	1.05	0.960	1.03
	1.09	4.33	1.89	3.87	2.32	3.52	2.60	3.31	2.77	3.18	2.86	2.11
$\mathbf{U_t}$ - Wand.	0.9964	1.0106	0.9824	1.0244	0.9691	1.0370	0.9573	1.0479	0.9474	1.0567	0.9399	1.0628
	2.9944	3.0168	2.9722	3.0385	2.9510	3.0590	2.9315	3.0774	2.9144	3.0931	2.9002	3.1056

Para o caso em que $\ddot{\mathbf{U}}_0 = \{0\ 0\}^T$, observou-se uma não concordância entre os resultados apresentados no livro e aqueles obtidos numericamente. Essa discrepância decorre do fato de que a referência considera apenas a condição inicial sem definir um passo de tempo, enquanto a implementação numérica integra a equação ao longo de um intervalo de $\Delta t = 28\ s$.

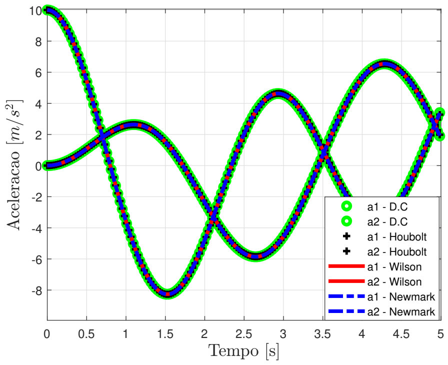
Os gráficos da Figura 1.2 apresentam o deslocamento, a velocidade e a aceleração dos dois graus de liberdade do sistema, considerando as condições iniciais, e comparando os métodos da diferença central, Houbolt, Wilson-θ e Newmark, para $\Delta t = 0.028\ s$. Observa-se que, para um passo de tempo suficientemente pequeno, todos os quatro métodos apresentam convergência consistente, produzindo respostas praticamente coincidentes no intervalo de tempo de 0 a 5 segundos.



(a) Deslocamento.



(b) Velocidade.



(c) Aceleração.

Figura 1.2: Respostas do sistema no domínio do tempo $\Delta t = 0.028\ s$.

1.5 Rotina em Matlab

```
function [u_t,v_t,a_t] = Newmark(K,M,C,R0,u0,v0,acel_0,Delta_t, t0,tf
    ,delta,alpha)
    %% === Entradas ====
    %% K       -> Matriz  de rigidez
    %% M       -> Matriz  de Massa
    %% C       -> Matriz de amortecimento
    %% R0      -> Vetor de carga
    %% u0      -> Deslocamento inicial
    %% v0      -> velocidade inicial
    %% acel_0  -> aceleracao inicial
    %% Delta_t -> Discretizacao do tempo
```

```
%% t0       -> Tempo inicial
%% tf       -> Tempo final
%% delta    -> constante (>= 0.5)
%% alpha    -> constante
%% === Saidas ===
%% u_t      -> Deslocamento  no tempo
%% v_t      -> velocidade  no tempo
%% a_t      -> aceleracao  no tempo

%% === Vetor tempo ===
t = t0:Delta_t:tf;
%% Caculo da constante alpha
% Verificacao dos parametros Newmark
if delta < 0.5
    error('delta deve ser maior do que 0.5 !')
elseif alpha < 0.25 * (0.5 + delta)^2
    error('alpha deve ser maior do que 0.25 * (0.5 + delta)^2 !')
end



%% Calculo das constantes iniciais
a0 = 1/(alpha * Delta_t^2); a1 = delta/(alpha * Delta_t); a2 =
   1/(alpha * Delta_t);
a3 = 1/(2* alpha) - 1; a4 = delta/alpha -1 ; a5 = Delta_t/2 * (
   delta/alpha-2);
a6 = Delta_t * (1 - delta);
a7 = delta * Delta_t;

%% Matriz de rigidez efetiva, nomeada  K_e
K_e = K + a0*M + a1*C;

%% Triangulizacao da matriz efetiva
[L,D,~] = ldl(K_e);
K_e = L*D*L';

%% Alocacao de memoria
R_t = zeros(size(R0,1),length(t));
a_t = zeros(size(acel_0,1),length(t));
v_t = zeros(size(v0,1),length(t));
u_t = zeros(size(u0,1),length(t));
```

```
%%  para o instante t = 0, recebe os valores iniciais
u_t(1:size(u0,1),1) = u0;
v_t(1:size(v0,1),1) = v0;
a_t(1:size(acel_0,1),1) = acel_0;

%% Loop temporal
for i = 1 :length(t)-1

%% Vetor da forca no instante t + Deltat

  R_t(:,i + i) = R0(:,i + 1) + M * (a0 * u_t(:, i) + a2 * v_t(:,i
     ) + a3 * a_t(:,i))...
      + C* (a1 * u_t(:,i) + a4 * v_t(:,i) + a5 * a_t(:,1));
%% Vetor do deslocamento no instante t + Delta_t
  u_t(:,i + 1) = K_e\R_t(:,i + i);

%% Calculo da velocidade e aceleracao em t + delta_t
a_t(:,i + 1) = a0 * (u_t(:,i + 1) - u_t(:,i)) - a2 * v_t(:,i) -
   a3 * a_t(:,i);
v_t(:,i + 1) = v_t(:,i) + a6 * a_t(:,i) + a7 * a_t(:,i + 1);



end


%% =============  Referencias ===============

%% Bathe, K.-J.; Wilson, E. L. Numerical Methods in Finite
   Element Analysis.
%% Prentice-Hall Civil Engineering and Engineering Mechanics
   Series, 1976.
%% ISBN: 0-13-627190-1.

end
end
```

2 Conclusão

A implementação do método de Newmark para o sistema estudado mostrou-se consistente com os resultados de referência apresentados por (BATHE; WILSON, 1976). Para pequenos passos de tempo, como $\Delta t = 0.28\ s$, os deslocamentos obtidos pelo script em MATLAB concordam com os valores do livro, com diferenças apenas na quarta casa decimal, indicando alta precisão da implementação.

Ao aumentar o passo de tempo para $\Delta t = 28.0\ s$, a rotina continua reproduzindo o comportamento esperado do sistema, confirmando a estabilidade do método mesmo para intervalos maiores, embora a precisão diminua ligeiramente, conforme esperado.

A alteração das condições iniciais, assumindo $\ddot{\mathbf{U}}_0 = \{0\ 0\}^T$, evidenciou que a discrepância entre os resultados numéricos e a referência se deve à ausência de definição de passo de tempo no livro.

Além disso, a análise apresentada na Figura 1.2 mostra que, para $\Delta t = 0.028\ s$, os métodos da diferença central, Houbolt, Wilson-θ e Newmark fornecem respostas praticamente coincidentes para deslocamento, velocidade e aceleração no intervalo de 0 a 5 segundos. Isso evidencia que, para passos de tempo suficientemente pequenos, todos os métodos convergem, tornando a escolha do método menos crítica. Entretanto, para passos maiores, a estabilidade e precisão podem variar significativamente, reforçando a importância da escolha adequada do Δt em simulações dinâmicas.

Em síntese, o método de Newmark implementado provou-se eficiente, estável e capaz de fornecer resultados confiáveis para diferentes condições iniciais e passos de tempo.

Referências

BATHE, K.-J.; WILSON, E. L. *Numerical Methods in Finite Element Analysis*. [S.l.]: Prentice-Hall, 1976. (Civil Engineering and Engineering Mechanics Series). ISBN 0-13-627190-1.