

Finite Difference Methods For Ordinary Differential Equation

Finite-Difference Methods For Linear and Nonlinear ODE And Modeling with MATLAB

Marwa Shehadeh

Palestine Polytechnic University
Applied Sciences College
Hebron
Palestine

Marwa_shehadeh-ppu@hotmail.com

Yasmin Mujahed

Palestine Polytechnic University
Applied Sciences College
Hebron
Palestine

sweet.smile1990@live.com

Abstract— we introduce finite difference methods for solving ordinary differential equation numerically.

In our research we discuss two types of finite difference, the first one is linear finite difference methods, and write MATLAB program to get the approximation solution, also we put two modeling with linear differential equation form. At the second one we talk about nonlinear finite difference methods, and write MATLAB program which approximate the solution of equations of this form, then an example was presented.

Finite-Difference Methods For Linear Problem

The finite difference method for the linear second-order boundary-value problem,
 $y'' = p(x)y' + q(x)y + r(x) \quad a \leq x \leq b,$
 $y(a) = \alpha, \quad y(b) = \beta \quad (1.1)$
 approximations be used to approximate both y' and y'' . we select an integer $N > 0$ and divide the interval $[a, b]$ into $(N + 1)$ equal subintervals whose endpoints are the mesh points $x_i = a + ih,$ for $i = 0, 1, \dots, N + 1,$ where $h = (b - a)/(N + 1)$. Choosing the step size h in this manner facilitates the application of a matrix.

the differential equation to be approximated is
 $y''(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i) \quad (1.2)$

$$w_0 = \alpha, \quad w_{N+1} = \beta$$

w_i is approximation of $y(x_i)$ for $i = 1, \dots, N$

$$\left(\frac{-w_{i+1} + 2w_i - w_{i-1}}{h^2}\right) + p(x_i)\left(\frac{w_{i+1} - w_{i-1}}{2h}\right) + q(x_i)w_i = -r(x_i), \quad [4] \quad (1.3)$$

for each $i = 1, 2, \dots, N$.

In the form we will consider, Eq(1.3) is rewritten as

$$\begin{aligned} & -\left(1 + \frac{h}{2}p(x_i)\right)w_{i-1} + (2 + h^2q(x_i))w_i \\ & -\left(1 - \frac{h}{2}p(x_i)\right)w_{i+1} = -h^2r(x_i), \quad [4] \end{aligned}$$

and the resulting system of equation, in the tridiagonal $N \times N$ matrix form

$$Aw = b \quad (1.4)$$

$$A =$$

$$\begin{bmatrix} 2+h^2q(x_1) & -1+\frac{h}{2}p(x_1) & 0 & \dots & \dots & \dots & 0 \\ -1-\frac{h}{2}p(x_2) & 2+h^2q(x_2) & -1+\frac{h}{2}p(x_2) & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & & & 0 \\ \vdots & & & & & & -1+\frac{h}{2}p(x_{N-1}) \\ 0 & \dots & \dots & \dots & -1-\frac{h}{2}p(x_N) & 2+h^2q(x_N) & \end{bmatrix}$$

$$.b = \begin{bmatrix} -h^2 r(x_1) + \left(1 + \frac{h}{2} p(x_1)\right) w_0 \\ -h^2 r(x_2) \\ \vdots \\ -h^2 r(x_{N-1}) \\ -h^2 r(x_N) - \left(1 + \frac{h}{2} p(x_N)\right) w_{N+1} \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}$$

MATLAB Program For Linear

Finite Difference

To approximate the solution of linear boundary value problem

$$y'' = p(x)y' + q(x)y + r(x) \quad a \leq x \leq b,$$

$$y(a) = \alpha, \quad y(b) = \beta.$$

n: number of subintervals.

function [T,Y] = findiff(p,q,r,a,b,alpha,beta,n)

T = zeros(1,n+1);

Y = zeros(1,n-1);

Va = zeros(1,n-2);

Vb = zeros(1,n-1);

Vc = zeros(1,n-2);

Vd = zeros(1,n-1);

h=(b-a)/n;

for j=1:n-1,

 Vt(j)=a + h*j;

end

for j=1:n-1,

 Vb(j) = -h^2*feval(r,Vt(j));

end

Vb(1) = Vb(1) + (1 + h/2*feval(p,Vt(1)))*alpha;

Vb(n-1) = Vb(n-1)+(1 - h/2*feval(p,Vt(n-1)))*beta;

for j = 1:n-1,

 Vd(j) = 2 + h^2*feval(q,Vt(j));

end

for j = 1 : n-2,

 Va(j) = -1 -h/2*feval(p,Vt(j+1));

end

for j = 1:n-2,

 Vc(j) = -1 +h/2*feval(p,Vt(j));

end

Y = trisys(Va,Vd,Vc,Vb);

T = [a,Vt,b];

Y = [alpha,Y,beta]

Y'

plot(T,Y,'o',t,f(t))

(f(t) the exact solution)

end

function Y=trisys(A,D,C,B)

n=length(B);

for k=2:n,

 mult=A(k-1)/D(k-1);

 D(k)=D(k)-mult*C(k-1);

 B(k)=B(k)-mult*B(k-1);

end

Y(n)=B(n)/D(n);

for k= (n-1):-1:1,

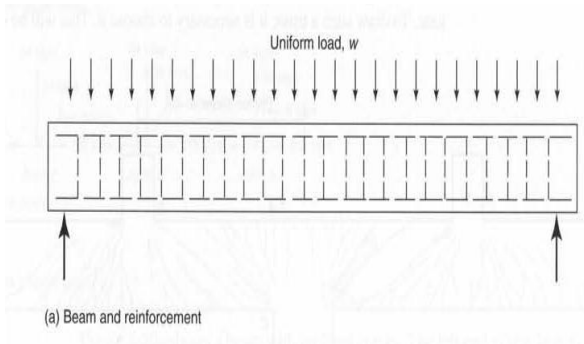
 Y(k)=(B(k)-C(k)*Y(k+1))/D(k);

end [4]

Note that 'trisys' algorithm is represent the solution of a tridiagonal linear system using [Algorithm (6.7.), Richard L. Burden, Numerical Analysis, 8th edition, p 408]

and we modified the program to plot the exact and approximation solution.

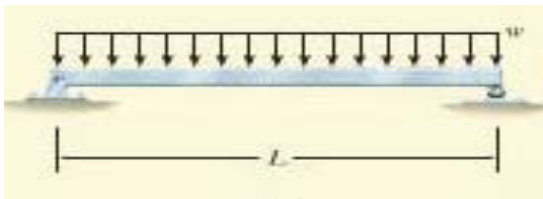
Bending Modeling And Example



Figure(1.1)

The distributed loads on the beam is represented by WL where W is the force of the loads on the beam, L is the length of the beam. [5]

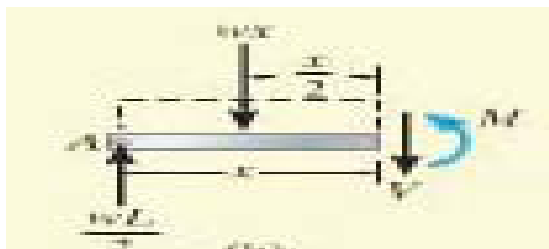
We determine all the reactive forces and moment acting on the beam, and resolve all the forces in to component acting perpendicular and parallel to the beams axis.



Figure(1.2)

The goal here is to find a relationship between the curvature (bending) and loads (external force).

Section the beam at each distance x , and draw the free body diagram of one of the segment as figure(1.3).



Figure(1.3)

The distributed loading on this segment, Wx , is represented by it's resultant force only after the segment is isolated as a free body diagram, this force acts through the centroid of the area

comprising the distributed loading, a distance of $\frac{x}{2}$ from the right end as show in figure(1.3) .

$\sum M = 0$ (M is the moment), where

$$M(x) = \frac{WL}{2}x - Wx\left(\frac{x}{2}\right)$$

$$M(x) = \frac{W}{2}(Lx - x^2) , [5] \quad (1.5)$$

We want to find a relation between moment and curvature:

The curvature equation is:

$$\frac{1}{\rho} = \frac{\frac{d^2y}{dx^2}}{\left(1 + \left(\frac{dy}{dx}\right)^2\right)^{\frac{3}{2}}} \quad \text{since } \frac{dy}{dx} \text{ is very small we}$$

ignore it

$$\text{, then } \frac{1}{\rho} = \frac{d^2y}{dx^2} \quad \text{so } \frac{1}{\rho} = y'' \quad (1.6)$$

By **Hooke's law**: $\sigma = E \cdot \varepsilon$

Where (σ is the stress, E is represent the constant of proportionality which is called modulus of elasticity, ε is the strain)

$$\varepsilon = \frac{\Delta l}{l} = \frac{ds - dx}{dx}$$

(Where ds is the new length after bending)

$$\text{So } \varepsilon = \frac{ds}{dx} - 1$$

Since $\rho \cdot d\theta = dx$ and

$$ds = (\rho - y)d$$

$$\text{So } \varepsilon = \frac{(\rho - y)d\theta - \rho d\theta}{\rho d\theta}$$

$$\rightarrow \varepsilon = \frac{-y}{\rho}$$

Note that $\sigma = -\frac{M}{I}y$

{where I is the moment of inertia }

$$\Rightarrow -\frac{M}{I}y = E \cdot \varepsilon$$

$$\Rightarrow -\frac{M}{I}y = \frac{-y}{\rho}E$$

$$\Rightarrow \frac{1}{\rho} = \frac{M}{EI} \quad \text{since } \frac{1}{\rho} = y''$$

$$\Rightarrow y'' = \frac{M}{EI} . [5]$$

Example : A beam has rectangular cross section with height(H) of 1m and width (b) of 2m, the length of the beam(L) is 6m while the load on

the beam is 75 k N/m. Determine the bending on the beam resulting of loading as shown in figure (1.4), where the boundary condition is $y(0) = 0$ and $y(L) = 0$?

Sol :

$$y'' = \frac{M}{EI} \text{ and } M(x) = \frac{W}{2}(Lx - x^2)$$

$$M(x) = \frac{75(kN/m)}{2}(6(m)x - x^2)$$

$$= 225x - \frac{75}{2}x^2$$

$$I = \frac{1}{12}(bH^3) \rightarrow I = \frac{1}{12}(2m * (1m)^3) \rightarrow$$

$$I = \frac{1}{6}m^4$$

$$\text{, then } y'' = \frac{(225x - \frac{75}{2}x^2)kN/m}{(20 * 10^3 MPa) * (\frac{1}{6}m^4)}$$

$$y'' = 0.0675x - 0.01125x^2$$

We want to get the exact solution by integrating

y'' ,

$$\Rightarrow y = \frac{0.0675x^3}{6} - \frac{0.01125x^4}{12} + c_1x + c_2 \text{ , by}$$

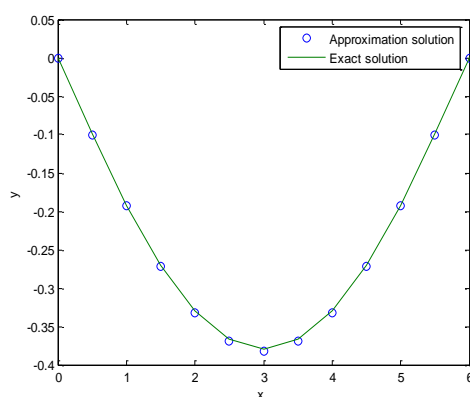
using the boundary condition

$$y = \frac{0.0675}{6}x^3 - \frac{0.01125}{12}x^4 - .2025x \text{ it is the}$$

exact solution

Let the step size ($h = 0.5$) on $[0,6]$,

$n = 12$.



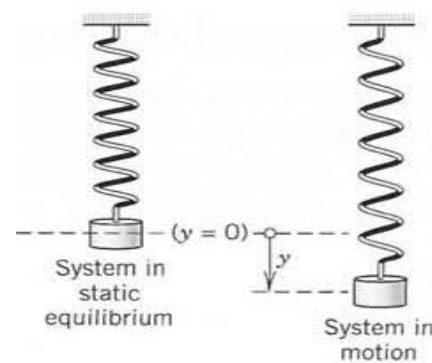
Figure(1.5)

Bending example

Vibrations Modeling And Example (Single Degree Of Freedom System)

** **single degree of freedom system** is that system consist of one mass. the number of independent displacements required to define the displaced positions of all the masses relative to their original position is called the number of **degree of freedom** for dynamic analysis.

We take an ordinary spring that resists compression as well extension and suspend it vertically from fixed support as shown in figure(1.6) . [1]



Figure(1.6)

At the lower end of the spring we attach a body of mass m . we assume m to be so large that we can neglect the mass of the spring. If we pull the body down a certain distance and then release it, it starts vibrating. We assume that it moves strictly vertically.

How can we obtain the motion of the body, say, the displacement $y(t)$ as a function of time t ? Now this motion is determine by **Newton's second law Force = Mass x acceleration**

$$\Rightarrow F_1 = m y''$$

$$\text{Where } y'' = \frac{d^2y}{dt^2} \text{ ,}$$

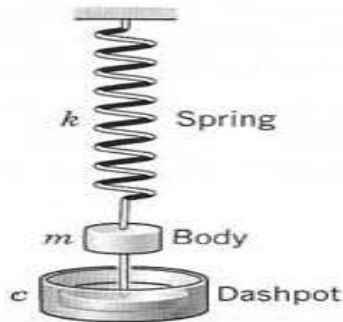
This force is called **Inertial force**.

We choose the downward direction as the positive direction, thus regarding downward forces as positive and upward as negative. Note

that the Inertial force is in the direction of motion.

From position $y = 0$ we pull the body downward. This stretches the spring by some amount $y > 0$ (the distance we pull it down). By **Hook's law** this causes an upward force F_2 in the spring. Where F_2 is proportional to the stretch and the constant $k > 0$ is called the spring constant

A **damping force**: the process by which free vibration steadily diminishes in amplitude. Physically this can be done by connecting the body to the dashpot as the figure(1.9)



Figure(1.9)

$F_3 = -cy'$, where c is the coefficient of viscous damping, y' is the velocity of body. It should be remembered that the damping force always opposes motion. Note that the dynamic equilibrium requires that the sum of forces is equal zero $\sum F = 0$

$$\text{, so } m y'' + c y' + k y = 0$$

$$\Rightarrow y'' = -\frac{c}{m} y' - \frac{k}{m} y. \quad [1]$$

Example: An iron of weight $w = 98 N$ stretches a spring $1.09 m$ what will it's motion(the displacement) be if we pull down the weight an additional $16 cm$ let the damping constant $c = 10 kg/sec$ with boundary condition $y(0) = 0.16$ and

$$y(10) = 0?$$

Sol :

$$\Rightarrow k = \frac{f}{y} = \frac{98}{1.09} = 90 \text{ kg/sec}^2$$

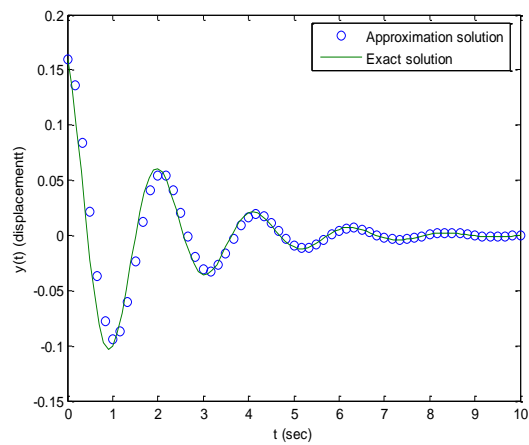
$$\Rightarrow m = \frac{w}{g} = \frac{98}{9.8} = 10 \text{ kg}$$

$$\text{by using } y'' = -\frac{c}{m} y' - \frac{k}{m} y \Rightarrow y''(t) = -\frac{10}{10} y'(t) - \frac{90}{10} y(t).$$

The exact solution is:

$$y(t) = e^{-.5t} (0.16x\cos(2.95804t) - 0.04337\sin(2.95804t))$$

Let the step size $h = \frac{1}{6}$ on $[0,10]$, $n = 60$



Figure(1.10)

Vibrations example

Finite-Difference Methods For Nonlinear Problem

The nonlinear problem is solved by a monotone iterative method which leads to a sequence of linearized equations.

For The general nonlinear boundary-value problem

$$f(x, y, y') = y'' \quad a \leq x \leq b \quad y(a) = \alpha, \quad y(b) = \beta \quad (2.1)$$

$$y''(x_i) = f(x_i, y(x_i), y'(x_i)) \quad (2.2)$$

we get

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} = f(x_i, y(x_i), y'(x_i)),$$

$$\frac{y(x_{i+1}) - y(x_{i-1}))}{2h} - \frac{h^2}{6} y'''(\eta_i) + \frac{h^2}{12} y^{(4)}(\xi_i)$$

Now by making the approximation by employing the boundary conditions and deleting the errors we get the difference method

$$w_0 = \alpha \quad w_{N+1} = \beta,$$

$$\text{and } -\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} + f(x_i, w_i, \frac{w_{i+1} - w_{i-1}}{2h}) = 0 \quad (2.3)$$

for each $i = 1, 2, \dots, N$

The $N \times N$ nonlinear system obtained from this method is

for $i = 1$

$$2w_1 - w_2 + h^2 f(x_1, w_1, \frac{w_2 - \alpha}{2h}) - \alpha = 0$$

for $i = 2$

$$-w_1 + 2w_2 - w_3 + h^2 f(x_2, w_2, \frac{w_3 - w_1}{2h}) = 0$$

⋮

for $i = N - 1$:

$$-w_{N-2} + 2w_{N-1} - w_N + h^2$$

$$f(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}) = 0$$

for $i = N$:

$$-w_{N-1} + 2w_N + h^2 f(x_N, w_N, \frac{\beta - w_{N-1}}{2h}) - \beta = 0.$$

[4]

We use the Newton's method for nonlinear system to approximate the solution to this system. A sequence of

iterates $\{(w_1^{(k)}, w_2^{(k)}, \dots, w_N^{(k)})^t\}$ is generated that converges to the solution of the above system,

provided that the initial approximation

$(w_1^{(0)}, w_2^{(0)}, \dots, w_N^{(0)})^t$ is sufficiently close to the solution $(w_1, w_2, \dots, w_N)^t$, [4]

$$J(w_1, w_2, \dots, w_N) =$$

$$\begin{bmatrix} -1 + \frac{h}{2} f_y(x_1, w_1, \frac{w_2 - \alpha}{2h}) & -1 + \frac{h}{2} f_y(x_1, w_1, \frac{w_2 - \alpha}{2h}) & 0 & \dots & 0 \\ -1 + \frac{h}{2} f_y(x_2, w_2, \frac{w_3 - w_1}{2h}) & 2 + h^2 f_{yy}(x_2, w_2, \frac{w_3 - w_1}{2h}) & -1 + \frac{h}{2} f_y(x_2, w_2, \frac{w_3 - w_1}{2h}) & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & -1 + \frac{h}{2} f_y(x_N, w_N, \frac{\beta - w_{N-1}}{2h}) & 2 + h^2 f_{yy}(x_N, w_N, \frac{\beta - w_{N-1}}{2h}) \end{bmatrix}$$

Now from Newton's method for nonlinear system

we can note that

$$V^{K-1} = -J(w^{K-1})^{-1} F(w^{K-1})$$

$$\text{Where } V^{K-1} = \begin{bmatrix} v_1^{K-1} \\ \vdots \\ v_N^{K-1} \end{bmatrix}$$

and $F(w^{K-1}) =$

$$\begin{bmatrix} 2w_1^{K-1} - w_2^{K-1} + h^2 f(x_1, w_1^{K-1}, \frac{w_2^{K-1} - \alpha}{2h}) - \alpha \\ -w_1^{K-1} + 2w_2^{K-1} - w_3^{K-1} + h^2 f(x_2, w_2^{K-1}, \frac{w_3^{K-1} - w_1^{K-1}}{2h}) \\ \vdots \\ -w_{N-1}^{K-1} + 2w_N^{K-1} + h^2 f(x_N, w_N^{K-1}, \frac{\beta - w_{N-1}^{K-1}}{2h}) - \beta \end{bmatrix}$$

$$\Rightarrow w_i^K = w_i^{K-1} + v_i \quad \text{for each } i = 1, 2, \dots, N,$$

[4]

and note that $w_i = \alpha + i \frac{(\beta - \alpha)}{b - a} h$ for

$i = 1, 2, \dots, N$

(this is obtained by passing a straight line through (a, α) and (b, β)).

MATLAB Program For Non-Linear Finite

Difference

To approximate the solution of nonlinear boundary value problem:

$$y'' = f(x, y, y') \quad a \leq x \leq b, \quad y(a) = \alpha, \\ y(b) = \beta$$

n: number of subinterval, m: number of iteration,
tol: tolerance.

```
function [T,w] =
nonlinear(f,fy,fyp,a,b,alpha,beta,n,m,tol)
T = zeros(1,n+1);
w = zeros(1,n-1);
v = zeros(1,n-1);
Va = zeros(1,n-2);
Vb = zeros(1,n-1);
Vc = zeros(1,n-2);
Vd = zeros(1,n-1);
h=(b-a)/n;
for j=1:n-1,
    Vt(j)=a + h*j;
end
for j=1:n-1,
    w(j)=alpha +j*((beta-alpha)/(b-a))*h;
end
k=1;
while k<=m,
Vb(1)=- (2*w(1)-w(2)-
alpha+h^2*feval(f,Vt(1),w(1),(w(2)-alpha)/
(2*h)));
Vb(n-1)=- (2*w(n-1)-beta-w(n-
2)+h^2*feval(f,Vt(n-1),w(n-1),(beta-w(n-
2))/(2*h)));
Vd(1)=2+feval(fy,Vt(1),w(1),(w(2)-
alpha)/(2*h))*h^2;
Vd(n-1)=2+feval(fy,Vt(n-1),w(n-1),(beta-w(n-
2))/(2*h))*h^2;
Va(1)=-1+h/2*feval(fyp,Vt(1),w(1),(w(2)-
alpha)/(2*h));
Vc(n-1)=-1-h/2*feval(fyp,Vt(n-1),w(n-1),(beta-
w(n-2))/(2*h));
    for j=2:n-2,
        Vb(j)=- (2*w(j)-w(j+1)-w(j-
1)+h^2*feval(f,Vt(j),w(j),(w(j+1)-w(j-
1))/(2*h)));
        Vd(j)=2+feval(fy,Vt(j),w(j),(w(j+1)-w(j-
1))/(2*h))*h^2;
```

```
Va(j)=-1+h/2*feval(fyp,Vt(j),w(j),(w(j+1)-
w(j-1))/(2*h));
Vc(j)=-1-h/2*feval(fyp,Vt(j),w(j),(w(j+1)-w(j-
1))/(2*h));
    end
v=trisys(Va,Vd,Vc,Vb);
for i=1:n-1,
    w(i)=w(i)+v(i);
end
if (abs(v)<=tol),
T = [a,Vt,b];
w = [alpha,w,beta]
w'
plot (T, w)
break
k=k+1;
end
end
```

*note 'trisys' here is the same as in

MATLAB Program For Linear Finite
Difference

Motor Motion Example

Direct-current motors are extensively used in variable-speed drives, where good dynamic response and steady-state performance are required. DC motors can provide a high starting torque and it is also possible to obtain speed control over a wide range. examples are in robotic drives, printers, machine tools and many others. [3]

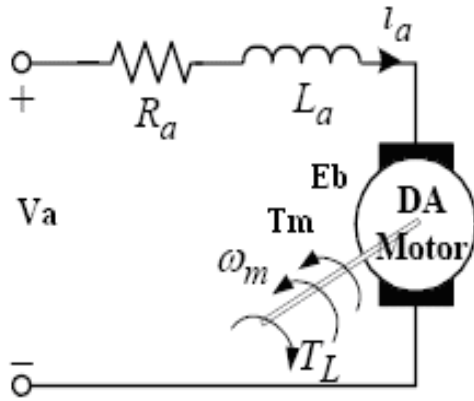


Figure (2-1)

The motor motion equation is:

$$J \frac{dw}{dt} = T - (T_L + Bw) \quad (2.4)$$

Where, J is the moment of inertia, w is the angular velocity (rad/sec), T is the motor torque ($N.m$), T_L is the load torque ($N.m$), B is the viscous friction constant ($N.m/rad/sec$).

Note that $w = \frac{d\theta}{dt}$. [3] (2.5)

$$\Rightarrow J \frac{d^2\theta}{dt^2} = T - (T_L + B \frac{d\theta}{dt})$$

$$\theta'' = \frac{T}{J} - \frac{(T_L + B\theta')}{J} \quad (2.6)$$

w becomes constant at steady state $\Rightarrow T = T_L$

Example: Separately excited DC motor with the viscous friction constant $0.0025 \text{ kg.m}^2/sec$, and the torque of the motor 344.25 N.m , the load torque 51.64 N.m . the moment of inertia equal to 2% of the angular velocity. Where the boundary condition is $\theta(0) = 0 \text{ rad}$, $\theta(5) = 1500 \text{ rad}$?

Sol:

Note that $J = 0.02 w \Rightarrow J = 0.02 \theta'$

By using equation (2.8) we get

$$\theta'' = \frac{344.25}{0.02 \theta'} - \frac{(51.64 + 0.0025\theta')}{0.02 \theta'}$$

$$\theta'' = \frac{14630.5}{\theta'} - 0.125$$

We want to get the approximation of solution, so let the step size ($h = 0.05$) on $[0,5]$, $n = 100$.

t_i	θ_i
0.00	0.000000
0.05	0.007583
0.10	0.015402
0.15	0.023452
0.20	0.031726
0.25	0.040218
0.30	0.048922
⋮	⋮
4.75	1.398589
4.80	1.418691
4.85	1.438884
4.90	1.459166
4.95	1.479539
5.00	1.500000

** note : θ_i equal $\theta_i * 10^3$

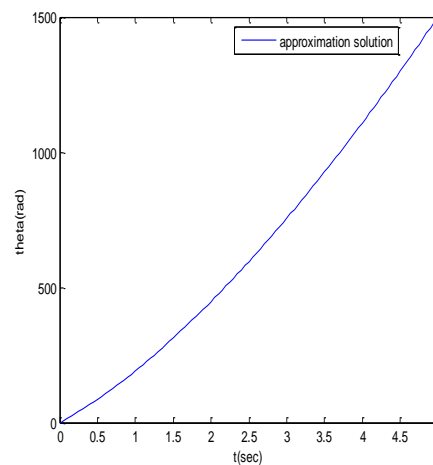
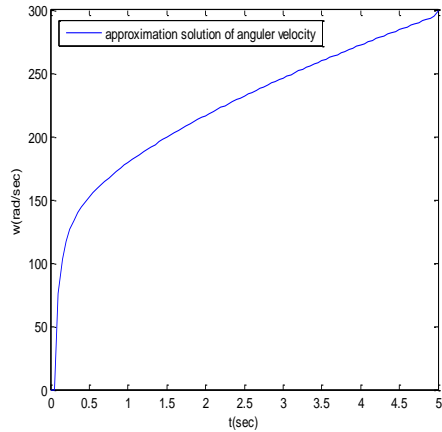


Figure (2-2)

Motor motion exampl



Figure(2-3)
Angular velocity

References

- [1] Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, Inc, 9th edition.

- [2] James K. Wight, and James G. MacGregor, Reinforced Concrete Mechanics And Design, Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, 5th edition

- [3] Muhammad H. Rashid, Power Electronics Circuits, Devices ,And Applications , Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, 3th edition.

- [4] Richard L. *Burden*, and J. Douglas Faires, Numerical Analysis, 8th edition.

- [5] R. C. Hibbeler, Mechanics Of Materials, Pearson Prentice Hall, 8th edition.