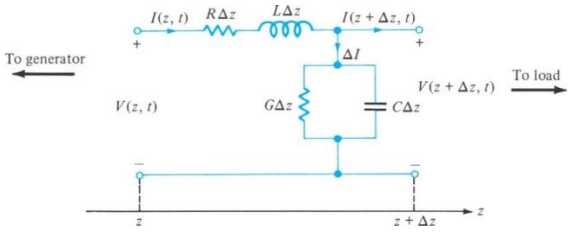
FINITE DIFFERENCE SOLUTIONS

SOLUTION OF THE ONE-DIMENSIONAL LOSSLESS TRANSMISSION LINE

A uniform TL may be modelled by the following circuit representation for a length Δz :



The transmission line equations for a lossless TL (i.e. $R = G = 0$) can be written by using KCL and KVL as:

$$\frac{\partial I}{\partial z} = -C\frac{\partial V}{\partial t} \qquad (1)$$

$$\frac{\partial V}{\partial z} = -L\frac{\partial I}{\partial t} \qquad (2)$$

Where L and C are the inductance and capacitance per unit length respectively. Note that, the TL equations are special case of Maxwell's equations.

Forward time and cantered space:

$$V_k^{n+1} = V_k^n - \frac{\Delta t}{2C\Delta z}\left(I_{k+1}^n - I_{k-1}^n\right)$$

$$I_k^{n+1} = I_k^n - \frac{\Delta t}{2L\Delta z}\left(V_{k+1}^n - V_{k-1}^n\right)$$

Cantered time and cantered space (Leapfrog algorithm):

$$V_k^{n+1} = V_k^{n-1} - \frac{\Delta t}{C\Delta k}\left(I_{k+1}^n - I_{k-1}^n\right)$$

$$I_k^{n+1} = I_k^{n-1} - \frac{\Delta t}{L\Delta z}\left(V_{k+1}^n - V_{k-1}^n\right)$$

In practice when you have coupled equations, interleaving the finite difference meshes for two variables often give better results. In this case, first order coupled equations are converted to difference equations at different grid points.

Now, approximate equation (1) at z_k and $t_{n+\frac{1}{2}}$ by using central differencing in both space and time, write the update equation for V:

$$\left.\frac{\partial I}{\partial z}\right|_{z_k}^{t_{n+\frac{1}{2}}} \approx \frac{I_{k+\frac{1}{2}}^{n+\frac{1}{2}} - I_{k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z}$$

$$\left.\frac{\partial V}{\partial t}\right|_{z_k}^{t_{n+\frac{1}{2}}} \approx \frac{V_k^{n+1} - V_k^n}{\Delta t}$$

The update equation for V :

$$V_k^{n+1} = V_k^n - \frac{\Delta t}{C\Delta z}\left(\frac{I_{k+\frac{1}{2}}^{n+\frac{1}{2}} - I_{k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z}\right)$$

This update equation can be used for $k = 2,3,....,N_z - 1$ for $n \geq 2$.

Where,

$z_k = (k-1)\Delta z, \quad k = 1,2,....,N_z$

$\Delta z = \frac{h}{N_z - 1}, \quad N_z \geq 2$

$t_n = (n-1)\Delta t, \quad n = 1,2,....$

$\Delta t = \frac{T}{M-1}, \quad M \geq 2$

Now consider the second TL equation. i.e. Eq. (2):

$$\left.\frac{\partial V}{\partial z}\right|_{k+\frac{1}{2}}^{n} \approx \frac{V_{k+1}^n - V_k^n}{\Delta z}$$

$$\left.\frac{\partial I}{\partial t}\right|_{k+\frac{1}{2}}^{n} \approx \frac{I_{k+\frac{1}{2}}^{n+\frac{1}{2}} - I_{k+\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta t}$$

The update equation for I :

$$I_{k+\frac{1}{2}}^{n+\frac{1}{2}} = I_{k+\frac{1}{2}}^{n-\frac{1}{2}} - \frac{\Delta t}{L\Delta z}\left(V_{k+1}^{n} - V_{k}^{n}\right)$$

This update equation can be used for $k = 1,2,3,\ldots.,N_z - 1$ for $n \geq 2$.

Additional grid points at half-space and half-time points are introduced:

$$z_{k+\frac{1}{2}} = \left(k - \frac{1}{2}\right)\Delta z, \quad k = 1,2,\ldots,N_z - 1$$

$$t_{n+\frac{1}{2}} = \left(n - \frac{1}{2}\right)\Delta t, \quad n = 1,2,3,\ldots$$

We will compute $V(z,t)$ at points (z_k, t_n) and $I(z,t)$ at points $\left(z_{k+\frac{1}{2}}, t_{n+\frac{1}{2}}\right)$. We have two dimensional arrays representing the voltage and the current. In each array the row represents the temporal evolution of the field at a particular point in space and a column represents the spatial distribution of the field at a particular point in time. But it is possible to store only some of the rows of each array.

```
%Initialize variables
dz=0.1;
c=3e8;
dt=dz/(c);
nz=50;
nstop=200;
cp=1e-9;
ind=1e-7;
for k=1:nz
v(k)=0;
i(k)=0;
end
%Time loop

for n=1:nstop
   %upgrade the voltage equation
     for k=2:nz-1
```

```
        v(k)=v(k)-(dt/(cp*dz))*(i(k)-i(k-1));
    end
    %gaussian pulse excitation, wide band source
    %v(2)=exp(-(n-30)^2/60);
   % Sine wave excitation, single frequency source
    v(2)=sin(2e8*pi*n*dt);
    %store the wave at different space points for all time values
     v1(n)=v(2);
     v2(n)=v(5);
     v3(n)=v(10);
    %upgrade the current excitation
    for k=1:nz-1
        i(k)=i(k)-(dt/(ind*dz))*(v(k+1)-v(k));
    end
end
y=1:nstop;

plot( y,v1,y,v2,y,v3)
```

SOLUTION OF THE WAVE EQUATION

Example: Solve the wave equation $\frac{\partial^2\Phi}{\partial t^2}=\frac{\partial^2\Phi}{\partial x^2}\quad 0<x<1,\quad t\geq 0$, subject to the boundary conditions $\Phi(0,t)=0=\Phi(1,t)\quad t\geq 0$ and the initial conditions

$$\Phi(x,0)=\sin\pi x,\quad 0<x<1,$$
$$\frac{\partial\Phi}{\partial t}(x,0)=0,\qquad 0<x<1$$

Solution:

Discretize the wave equation:

$$\frac{\Phi_i^{n+1}-2\Phi_i^n+\Phi_i^{n-1}}{(\Delta t)^2}-\frac{\Phi_{i+1}^n-2\Phi_i^n+\Phi_{i-1}^n}{(\Delta x)^2}=0$$

Take $r=\left(\frac{\Delta t}{\Delta x}\right)^2=1$

$$\Phi_i^{n+1}=\Phi_{i+1}^n+\Phi_{i-1}^n-\Phi_i^{n-1},\quad n\geq 1$$

Initial Condition (n=0):

$$\frac{\partial \Phi(x,0)}{\partial t} \simeq \frac{\Phi_i^1 - \Phi_i^{-1}}{2\Delta t} = 0$$

$$\Phi_i^1 = \Phi_i^{-1}$$

Substitute to get the starting formula:

$$\Phi_i^1 = \Phi_{i+1}^0 + \Phi_{i-1}^0 - \Phi_i^{-1}$$

$$\Phi_i^1 = \Phi_{i+1}^0 + \Phi_{i-1}^0 - \Phi_i^1$$

$$\Phi_i^1 = \frac{1}{2}\left(\Phi_{i+1}^0 + \Phi_{i-1}^0\right)$$

Since v_p =1, r=1, chose

$$\Delta x = \Delta t = 0.1$$

Matlab Program:

```
% Wave Equation
dx=.1;
% initialization
 for i=1:11
    for j=1:11
        phi(i,j)=0;
    end
end

for i=1:11
    x=dx*i;
    phi(i,1)=sin(pi*x);
end

for i=2:10
    phi(i,2)=(phi(i+1,1)+phi(i-1,1))/2;
end
 % apply the explicit formula
for j=3:11
        for i=2:10
        phi(i,j+1)=phi(i+1,j)+phi(i-1,j)-phi(i,j-1);
    end
end

 mesh(phi(1:11,1:11))
```

Accuracy and Stability FD Solutions

The question of accuracy and stability of numerical methods is extremely important if our solution is to be reliable and useful.

Accuracy is the closeness of the approximate solution to the exact solutions.

Stability is the requirement that the scheme does not increase the magnitude of the solution with increase in time.

Unavoidable errors in numerical solution of physical problems:

1) modeling errors,
2) truncation (or discretization) errors,
3) round-off errors.

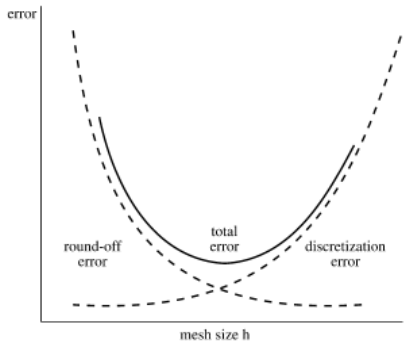
Modeling errors: Several assumptions are made for obtaining the mathematical model. i.e. nonlinear system may be represented by a liner PDE.

Truncation errors arise from the fact that in numerical analysis we can deal only with finite number of terms of a series. Truncation errors may be reduced,

1) by using finer meshes. i. e. smaller time and space step sizes and more number of points,
2) by using a large number of terms in the series expansion of derivatives.

Round-off Errors are due to finite precision of computers. Round-off error may be reduced by using double precision.

Error as a Function of a Mesh Size



Reducing mesh size will increase the accuracy, but it's not possible to infinitely decrease the mesh size. On the other hand using finer mesh may increase the truncation error. As can be seen from the figure above, a minimum total error can be found for a particular solution.

Stability of FD Solutions

A numerical solution is said to be stable, if a small error at any stage, produces a smaller cumulative error. Otherwise it is unstable.

To determine whether the FD scheme is stable, define an error ε^n, which occurs at time step ***n***, assuming a single independent variable. Define the amplification of this error at time step ***n+1*** as:

$$\varepsilon^{n+1} = g\varepsilon^n$$

where, g is the amplification factor.

For the stability of the difference scheme it is required that the above equation satisfies:

$$\left|\varepsilon^{n+1}\right| \le \left|\varepsilon^n\right|$$

Or,

$$\left|g\right| \le 1$$

Von Neumann's Method

One useful and simple method of finding a stability criterion for a difference scheme is to construct a Fourier analysis of the difference equation and derive the amplification factor. We illustrate this technique, known as von Neumann's Method, by considering the explicit FD formulation of the Diffusion equation.

The diffusion equation:

$$\frac{\partial V}{\partial t} = k\frac{\partial^2 V}{\partial x^2} \qquad (3)$$

The explicit scheme:

$$V_i^{n+1} = V_i^n + \frac{\Delta t}{(RC)(\Delta x)^2}\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right) \qquad (4)$$

with

$$r = \frac{\Delta t}{(RC)(\Delta x)^2} \qquad (5)$$

Simplified as:

$$V_i^{n+1} = V_i^n(1-2r) + r\left(V_{i+1}^n + V_{i-1}^n\right) \qquad (6)$$

Consider the solution:

$$V_i^n = A^n e^{jk(i\Delta x)} \qquad 0 < x < 1 \qquad (7)$$

Where, k is the wavenumber and $x = i\Delta x$, substitute in equation (6):

$$A^{n+1}e^{jk(i\Delta x)} = A^n e^{jk(i\Delta x)}(1-2r) + r\left(e^{jk(i+1)\Delta x} + e^{jk(i-1)\Delta x}\right)A^n$$

$$A^{n+1} = A^n(1-2r) + r\left(e^{jk\Delta x} + e^{-jk\Delta x}\right)A^n$$

Since,

$$\cos kx = \frac{e^{jk\Delta x} + e^{-jk\Delta x}}{2}$$

$$A^{n+1} = A^n(1-2r+2r\cos k\Delta x)$$

The amplification factor:

$$g = \frac{A^{n+1}}{A^n} = 1-2r+2r\cos k\Delta x$$

$$\cos k\Delta x = \cos^2\frac{k\Delta x}{2} - \sin^2\frac{k\Delta x}{2} = 1-2\sin^2\frac{k\Delta x}{2}$$

Thus,

$$g = \frac{A^{n+1}}{A^n} = 1-2r+2r\left(1-2\sin^2\frac{k\Delta x}{2}\right)$$

$$g = 1-4r\sin^2\frac{k\Delta x}{2}$$

For stability $|g| \le 1$.

$$\left|1-4r\sin^2\frac{k\Delta x}{2}\right| \le 1$$

This condition is true for all k . Let us take the maximum value of the sine function. Then,

$$|1-4r| \le 1$$

Or,

$$1-4r \ge -1 \quad \text{and} \quad r \ge 0$$

Or,

$$r \le \frac{1}{2} \quad \text{and} \quad r \ge 0$$

$r = 0$ implies that $\Delta t = 0$ which is impractical. Thus,

$$0 < r \le \frac{1}{2}$$

Finite Differencing of Elliptic PDE's

Consider the two dimensional Poisson's Equation

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = g(x, y)$$

Central difference approximation for the partial derivatives:

$$\frac{\partial^2 V}{\partial x^2} = \frac{V(i+1, j) - 2V(i, j) + V(i-1, j)}{(\Delta x)^2}$$

$$\frac{\partial^2 V}{\partial y^2} = \frac{V(i, j+1) - 2V(i, j) + V(i, j-1)}{(\Delta y)^2}$$

Where,

$$x = i\Delta x, y = j\Delta y \ \ and \ \ i, j = 1, 2, 3...$$

$$\Delta x = \Delta y = h$$

FD approximation of the Poisson's equation after simplification

$$\left[V(i+1, j) + V(i-1, j) + V(i, j+1) + Vi, j-1\right] - 4V(i, j) = h^2 g(i, j)$$

$$V(i, j) = \frac{1}{4}\left[V(i+1, j) + V(i-1, j) + V(i, j+1) + V(i, j-1) - h^2 g(i, j)\right]$$

When the source term vanishes, the Poisson's equation leads to the Laplace's equation. Thus for the same mesh size *h:*

$$V(i,j)=\frac{1}{4}\left[V(i+1,j)+V(i-1,j)+V(i,j+1)+V(i,j-1)\right]$$

The application of the finite difference method to elliptic PDEs often leads to a large system of algebraic equations to be solved.

Solution of such equations is a major problem. **Band matrix** and **iterative methods** are commonly used to solve the system of equations.

Band Matrix Method

Notice that, only the nearest neighboring nodes affect the value of *V,* at each node. Application of the FD equations results in a set of equations such that:

$$[A][X]=[B]$$

Where $\bar{A}$ is a sparse matrix (it has many zeros), X is the column matrix consisting of the unknown values, and $\bar{B}$ is the column matrix containing the known values of V . So:

$$[X]=[A]^{-1}[B]$$

Iterative Methods

An iterative method is one in which the first approximation is used to calculate the second approximation which in turn is used to calculate the third and so on until a specified tolerance is achieved. The three common iterative methods are Jacobi, Gauss-Seidel and successive over-relaxation (SOR) methods. Apply SOR to equation:

$$V(i,j)=\frac{1}{4}\left[V(i+1,j)+V(i-1,j)+V(i,j+1)+V(i,j-1)-h^2g(i,j)\right]$$

Define a residual function $R(i,j)$ at point (i,j) as the value of $V(i,j)$ does not satisfy the equation given above:

$$R(i,j)=\left[V(i+1,j)+V(i-1,j)+V(i,j+1)+V(i,j-1)-4V(i,j)-h^2g(i,j)\right]$$

The value of residual at nth iteration (i.e. $R^n(i,j)$, may be considered as a correction factor to be added to $V(i,j)$ to correct it. As $R^n(i,j)$ approaches to zero, the value of $V(i,j)$ will converge to its correct

value. Hence, to improve the rate of convergence we multiply $R^n(i,j)$ by ω and add that to $V(i,j)$ at nth to get the value at (n+1). Thus,

$$V_{(i,j)}^{n+1} = V_{(i,j)}^{n} + \frac{\omega}{4} R_{(i,j)}^{n}$$

The parameter ω is known as the relaxation factor which lies between 1 and 2. The optimum value must be obtained by trial and error.

```
a=1;
b=1;
%x(0)=v4, x(1)=v2, y(0)=v1, y(1)=20
v1=0;
v2=10;
v3=20;
v4=-10;
nx=30;
ny=30;
h=a/nx;
nstop=100;%number of iterations

%initialization of fixed nodes
for i=1:nx-1
    for j=1:ny-1
        v(i,j)=(v1+v2+v3+v4)/4.;
    end
end
 % B.C's
for i=1:nx-1
    v(i,1)=v1;
    v(i,ny)=v3;
end
 for j=1:ny-1
    v(1,j)=v4;
    v(nx,j)=v2;
end
v(1,1)=(v1+v4)/2.;
v(nx,1)=(v1+v2)/2.;
v(1,ny)=(v3+v4)/2.;
v(nx,ny)=(v2+v3)/2.;
w=1;
for n=1:nstop

for i=2:nx-1
    x=h*i;
     for j=2:ny-1
         y=h*j;
         g=-36*pi*x*(y-1);
         r=w*((v(i+1,j)+v(i-1,j)+v(i,j+1)+v(i,j-1)-h*h*g))/4.-w*v(i,j);

v(i,j)=v(i,j)+r;

     end
end
end
```

```
mesh(v(1:nx,1:ny))
```