Summary of Frequency Domain Method of Response Analysis

http://geotechnic.ucsd.edu/se203

1. Equation of motion of an SDOF system

SDOF system with viscous damping, the equation of motion is given by:

$$m\ddot{u} + c\dot{u} + ku = p(t) \tag{1}$$

SDOF system with rate independent damping, the equation of motion is given by:

$$m\ddot{u} + k(1 + i\eta)u = p(t) \tag{2}$$

The complete solution of the equation of motion is

$$u(t) = u_t(t) + u_s(t) \tag{3}$$

$$\dot{u}(t) = \dot{u}_t(t) + \dot{u}_s(t) \tag{4}$$

where $u_s(t)$ is the steady-state response or forced vibration of the SDOF, and $u_t(t)$ is the transient response or free vibration of the SDOF.

2. Steady-state response of an SDOF system by frequency domain method

Steady state response of an SDOF system to periodic excitation is obtained at the aid of Fourier Transform and Inverse Fourier Transform.

2.1 Transfer function of an SDOF system

The transfer function $H(\omega)$ describes the steady-state response of the SDOF system to complex excitation $p(t) = Pe^{i\omega t}$.

For an SDOF system with viscous damping, the transfer function is given by:

$$H(\omega)=\frac{1}{k}\frac{1}{[1-(\omega/\omega_n)^2]+i[2\xi(\omega/\omega_n)]} \tag{5}$$

where $\omega_n=\sqrt{\frac{k}{m}}$ is system natural frequency, and $\xi=\frac{c}{2m\omega_n}$ is damping ratio.

For an SDOF system with rate independent damping, the transfer function is given by:

$$H(\omega)=\frac{1}{k}\frac{1}{[1-(\omega/\omega_n)^2]+i\eta} \tag{6}$$

2.2 Steady-state response to complex excitation $p(t)=Pe^{i\omega t}$

If an SDOF system is subjected to a complex excitation $p(t)=Pe^{i\omega t}$, where P is the amplitude of the excitation (a scalar) and ω is frequency, then the steady-state responses of the SDOF system to this complex excitation are:

In frequency domain:

Displacement: $$U(\omega)=H(\omega)P \tag{7}$$

Velocity: $$\dot{U}(\omega)=i\omega H(\omega)P \tag{8}$$

In time domain:

Displacement: $$u_s(t)=U(\omega)e^{i\omega t} \tag{9}$$

Velocity: $$\dot{u}_s(t)=\dot{U}(\omega)e^{i\omega t} \tag{10}$$

where $H(\omega)$ is transfer function evaluated through Eq.(5) or (6) depending on the type of damping used.

2.3 Steady-state response to periodic excitation

A periodic function $p(t)=p(t+T_0)$ with period of T_0 can be expressed as a sum of infinite number of complex function $Pe^{i\omega t}$ described in Section 2.2 above. Each complex function has different amplitude P_j and frequency ω_j. The steady-state response of an SDOF system to each function $P_j e^{i\omega_j t}$ can be easily found by using Eqs.(7)-(10). Superposing the response to each function or frequency gives the steady-state response of the SDOF system to $p(t)$.

Such expression of p(t), called Fourier series expansion of p(t), is shown below by Eq.(11).

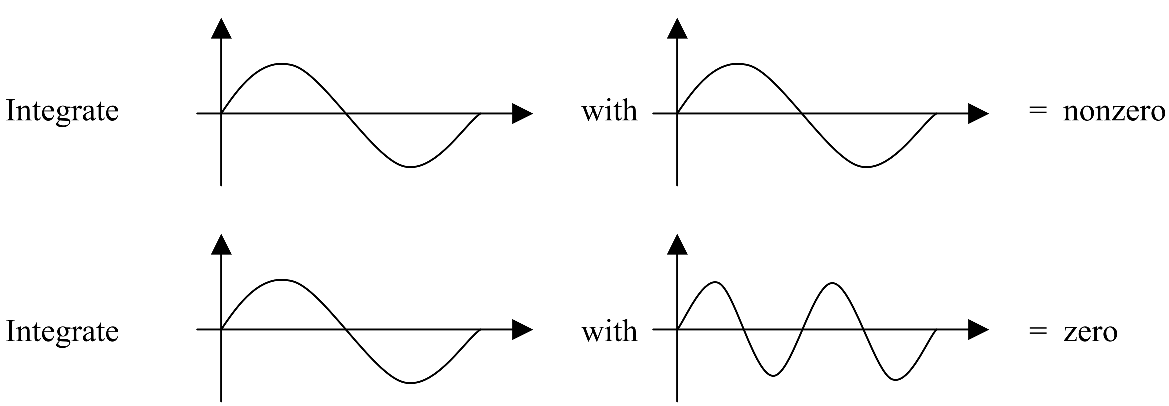
$$p(t) = \sum_{j=-\infty}^{\infty} P_j e^{i\omega_j t} \tag{11}$$

where $\omega_j = j\omega_0 = j\dfrac{2\pi}{T_0}$

The coefficient P_j in Eq.(11), called Fourier coefficient, is evaluated by the integration shown in Eq.(12). Due to the orthogonality of harmonic functions, such an integral eliminates all the components with other frequencies in p(t) except the one with frequency ω_j.

$$P_j = \frac{1}{T_0}\int_{t_1}^{t_1+T_0} p(t)e^{-i(\omega_j t)}dt \tag{12}$$

Concept of orthogonality:



As an illustration of the idea, the response to the j^{th} term in (11) is given below:

In frequency domain:

Displacement: $$U(\omega_j) = H(\omega_j)P_j \tag{13}$$

Velocity: $$\dot{U}(\omega_j) = i\omega_j H(\omega_j)P_j \tag{14}$$

In time domain:

Displacement: $$u_{sj}(t) = U(\omega_j)e^{i\omega_j t} \tag{15}$$

Velocity: $$\dot{u}_{sj}(t) = \dot{U}(\omega_j)e^{i\omega_j t} \tag{16}$$

The steady-state response to p(t) is the sum of responses to individual terms:

Displacement: $$u_s(t) = \sum_{j=-\infty}^{\infty} U(\omega_j)e^{i\omega_j t} \tag{17}$$

Velocity: $$\dot{u}_s(t) = \sum_{j=-\infty}^{\infty} \dot{U}(\omega_j)e^{i\omega_j t} \tag{18}$$

The rocedures for the frequency domain analysis of response to periodic excitation are summarized in Figure 1.

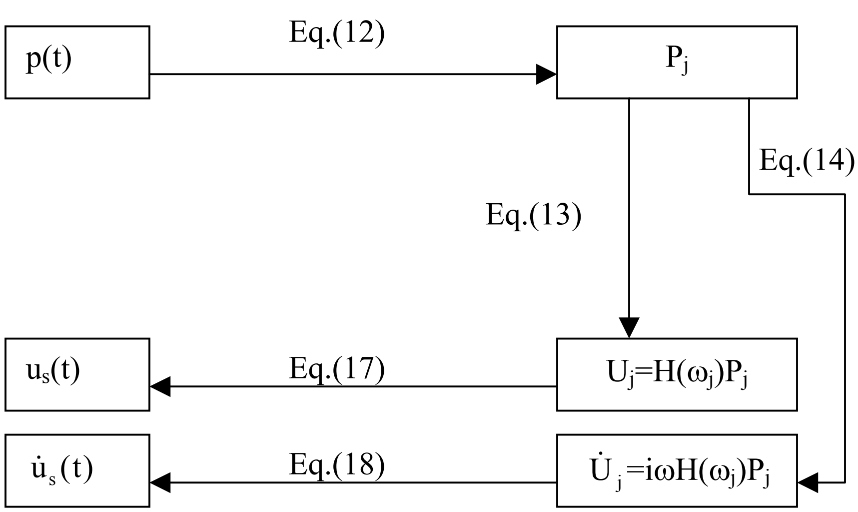


Figure 1 Flow chart for frequency domain analysis of response to periodic excitation.

2.4 Steady-state response to aperiodic excitation

Similar to the periodic excitation case, an aperiodic excitation can also be expressed as the sum of complex function $P(\omega)e^{i\omega t}$ at different frequencies ω (Eq.(19)):

$$p(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} P(\omega)e^{i\omega t}d\omega \qquad (19)$$

where

$$P(\omega) = \int_{-\infty}^{+\infty} p(t)e^{-i\omega t}dt \qquad (20)$$

Eq.(19) is known as the inverse Fourier transform (IFT) of P(ω) and Eq.((20) is known as the Fourier transform (FT) of p(t). P(ω) will be evaluated numerically as discussed in the next section (Section 2.5).

As in the periodic excitation case, the steady-state responses of an SDOF system to each excitation term $P(\omega)e^{i\omega t}$ are:

In frequency domain:

Displacement:
$$U(\omega) = H(\omega)P(\omega) \qquad (21)$$

Velocity:
$$\dot{U}(\omega) = i\omega H(\omega)P(\omega) \qquad (22)$$

In time domain:

Displacement:
$$u_{s\omega}(t) = U(\omega)e^{i\omega t} \qquad (23)$$

Velocity:
$$\dot{u}_{s\omega}(t) = \dot{U}(\omega)e^{i\omega t} \qquad (24)$$

The steady-state responses to p(t) in Eq.(19) are the integral of (23) and (24) over frequency as below:

Displacement:
$$u_s(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} U(\omega)e^{i\omega t}d\omega \qquad (25)$$

Velocity:
$$\dot{u}_s(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \dot{U}(\omega)e^{i\omega t}d\omega \qquad (26)$$

It is clear that Eqs.(25) and (26) are actually the IFT of $U(\omega)$ and $\dot{U}(\omega)$. These IFTs will also be evaluated numerically as discussed in Section 2.5.

The procedures for the frequency domain analysis of response to aperiodic excitation are summarized in Figure 2.

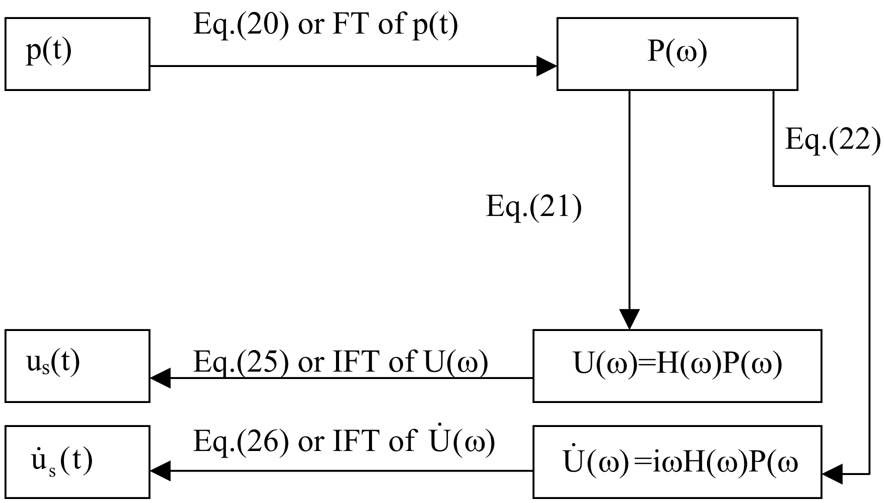


Figure 2 Flow chart for frequency domain analysis of response to aperiodic excitation.

2.5 Steady-state response by discrete Fourier transform

Due to the difficulty in analytically evaluating the integrals in the FT (Eq.(20)) and the IFT (Eqs.(25)-(26)), numerical Fourier transform or discrete Fourier transform (DFT) is needed. This section discusses how computer programs do the DFT and the inverse DFT.

Step 1 Analysis duration T_0

Suppose an actual excitation has duration t_d. As the DFT and inverse DFT are equivalent to approximating an aperiodic function by a periodic function, analysis duration T_0 should be t_d plus a free vibration duration t_f to avoid the effects of other cycles on responses. t_f should be long enough to let free vibration (after t_d) be damped out. Obviously low damping requires longer t_f. Therefore, the excitation turns into a forcing function with duration T_0 (Figure 3).

Step 2 Discretize excitation p(t)

The forcing function p(t) over the time duration T_0 is sampled at N equally spaced time instants, numbered from 0 to N (Figure 3). The sampling interval is denoted by Δt. Δt depends on the maximum frequency to obtain ($f_{max} = \frac{1}{2\Delta t}$ or $\omega_{max} = \frac{\pi}{\Delta t}$). The forcing function p(t) is then is defined by a set of discrete points (t_n, p_n).

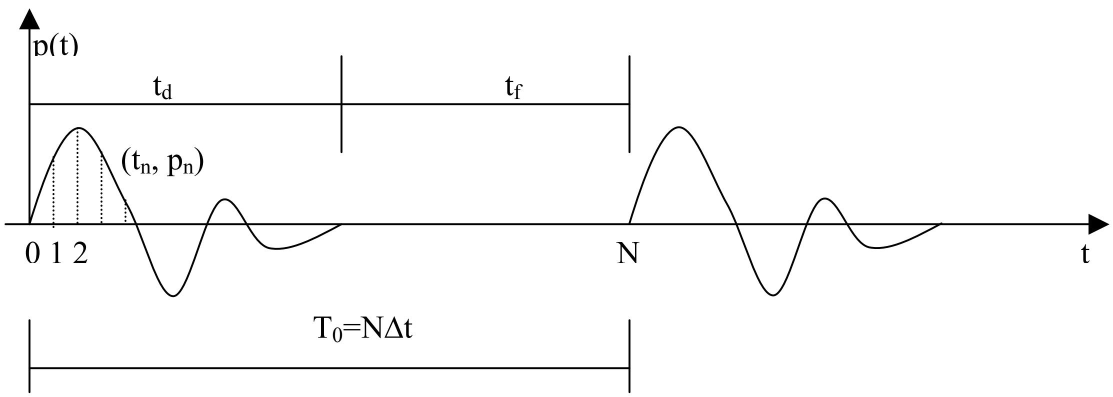


Figure 3 Excitation p(t) and its discretization.

Step 3 Do DFT of p(t)

The Fourier transform and the inverse Fourier transform in discrete form are given by:

Inverse Fourier Transform
$$p_n = \sum_{j=0}^{N-1} P_j e^{i(2\pi nj/N)} \quad (27)$$

Fourier Transform
$$P_j = \frac{1}{N}\sum_{n=0}^{N-1} p_n e^{-i(2\pi nj/N)} \quad (28)$$

It is seen that the array P_j is the DFT of the excitation sequence p_n and the array p_n is the inverse DFT of the sequence P_j.

Step 4 Calculate response in frequency domain

In frequency domain, the responses of the SDOF to each excitation term in (27) are:

$$\text{Displacement:} \quad U(\omega_j) = H(\omega_j)P_j \tag{29}$$

$$\text{Velocity:} \quad \dot{U}(\omega_j) = i\omega_j H(\omega_j)P_j \tag{30}$$

$$\text{ATTENTION:} \quad \omega_j = \begin{cases} j\omega_0 & 0 \le j \le N/2 \\ -(N-j)\omega_0 & N/2 < j \le N-1 \end{cases} \tag{31}$$

Step 5 Do Inverse DFT of U *and* $\dot{U}$ *to get steady-state response in time domain.*

The steady-state responses to excitation sequence p_n are the inverse DFT of the sequence U and $\dot{U}$.

$$\text{Displacement:} \quad u_{sn} = \sum_{j=0}^{N-1} U_j e^{i(2\pi nj/N)} \tag{32}$$

$$\text{Velocity:} \quad \dot{u}_{sn} = \sum_{j=0}^{N-1} \dot{U}_j e^{i(2\pi nj/N)} \tag{33}$$

The procedures for the frequency domain analysis of response using discrete Fourier transform are summarized in Figure 4.

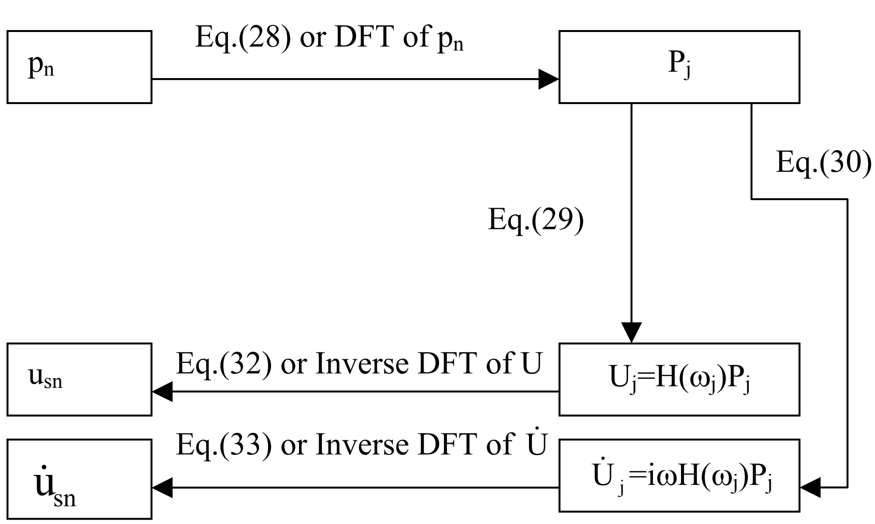


Figure 4 Flow chart for frequency domain analysis of response using discrete Fourier transform.

3 Transient response of an SDOF system

For an SDOF system with viscous damping, its transient response or free vibration is given by:

Displacement: $$u_t(t) = e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t) \tag{34}$$

Velocity: $$\begin{aligned}&\dot{u}_t(t) = \\ &e^{-\xi\omega_n t}(-\omega_D A\sin\omega_D t + \omega_D B\cos\omega_D t) \\ &- e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)\xi\omega_n\end{aligned} \tag{35}$$

Velocity: $$\begin{aligned}&\ddot{u}_t(t) = \\ &e^{-\xi\omega_n t}(-\omega_D A\sin\omega_D t + \omega_D B\cos\omega_D t) \\ &- e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)\xi\omega_n\end{aligned} \tag{36}$$

where

$\omega_n = \sqrt{\frac{k}{m}}$ is system natural frequency,

$\xi = \frac{c}{2m\omega_n}$ is damping ratio,

$\omega_D = \omega_n\sqrt{1-\xi^2}$,

A and B are constants determined from initial displacement u(0) and velocity $\dot{u}(0)$.

Imposing initial conditions to the complete solution Eq.(3) and (4) gives:

$$\begin{cases} u(0) = u_t(0) + u_s(0) \\ \dot{u}(0) = \dot{u}_t(0) + \dot{u}_s(0) \end{cases} \tag{37}$$

or

$$\begin{cases} u(0) = A + u_s(0) \\ \dot{u}(0) = -\xi\omega_n A + B\omega_D + \dot{u}_s(0) \end{cases} \tag{38}$$

yields

$$A = u(0) - u_s(0) \tag{39}$$

$$B = \{\dot{u}(0) + \xi\omega_n[u(0) - u_s(0)] - \dot{u}_s(0)\}/\omega_D \tag{40}$$

where $u_s(0)$ and $\dot{u}_s(0)$ are obtained by frequency domain method shown in Section 2.

Note: Eqs.(5) and (6) give

$$\eta = 2\xi \frac{\omega}{\omega_n}$$

At natural frequency $\eta = 2\xi$, so for an SDOF system with rate independent damping, its transient response can be obtained by replacing ξ with $\eta/2$.

A flow chart for computing the total response of the SDOF are shown in Figure 5.

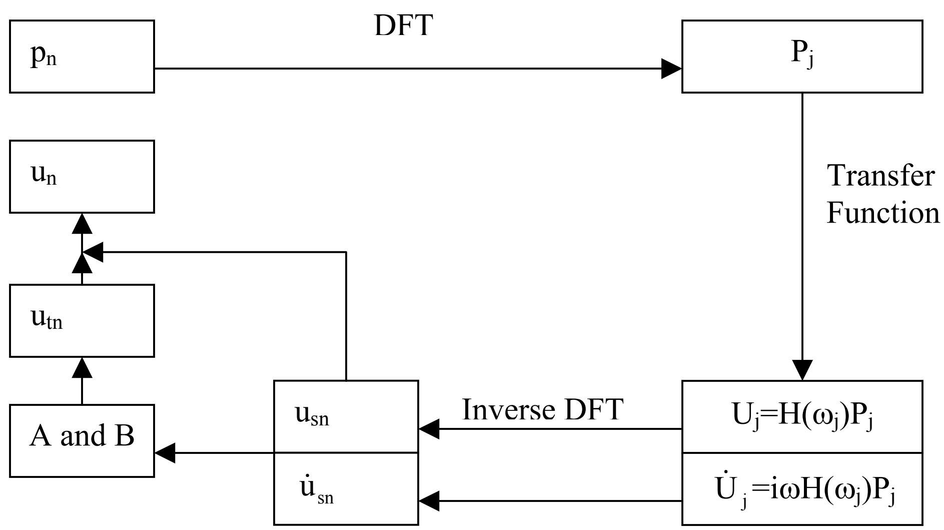


Figure 5 Flow chart for analysis of response using discrete Fourier transform.

4 Matlab Implementation

DFT and inverse DFT are carried out by calling functions FFT and IFFT respectively.

Step 1 Discretize excitation p(t)

1.1 Determine the maximum frequency f_{max} or ω_{max} you want to catch. Use the relationship $f_{max} = \frac{1}{2\Delta t}$ or $\omega_{max} = \frac{\pi}{\Delta t}$ to find time interval Δt. Quite often, discretized p(t) is given at known Δt (i.e., f_{max} or ω_{max} is known)

1.2 Determine the analysis duration T_0 or number of excitation data points N.

The number of excitation data points should be an odd number. If not, pad zeroes to it.

$$T_0 = N\,\Delta t$$

$$\omega_0 = 2\pi/ T_0$$

Note: For earthquake engineering, p(n) = -m*a(n), where a(n) is ground acceleration.

Step 2 Do DFT of array p(n)

```
P=fft(p);
```

Step 3 Compute steady-state response in frequency domain

```
for j=0:N-1
   if       j>=0 & j<=N/2
      ωj =j*ω0;
   elseif   j>N/2 & j<=N-1
      ωj =-(N-j)*ω0;
   end
   hwj=1/k*1/((1-(ωj/ ωn)^2)+i*(2*ξ* ωj /ωn));
   uw(j+1)=hwj*P(j+1); % In Matlab, array index starts from 1 instead of 0
   vw(j+1)=i*ωj*hwj*P(j+1);
end
```

Step 4 Do Inverse DFT of array uw and array vw to get steady-state response in time domain.

us=ifft(uw);

vs=ifft(vw);

Step 5 Add transient response to steady-state response to get total response

Note: array index starts from 1 in Matlab.

$A = u0 - us(1)$

$B = \{v(0) + \xi\omega_n[u0 - us(1)] - vs(1)\} / \omega_D$

for j=0:N-1

t=j*Δt

u(j+1)=us(j+1)+ $e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)$

v(j+1)=vs(j+1)+ $e^{-\xi\omega_n t}(-\omega_D A\sin\omega_D t + \omega_D B\cos\omega_D t) - e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)\xi\omega_n$

end

5 Fortran Implementation

DFT and inverse DFT are carried out by subroutine FFT by John F. Hall (1982). "An FFT algorithm for structural dynamics", Earthquake Engineering and Structural Dynamics, vol.10, pp.797-811.

Step 1 Discretize excitation p(t)

1.1 Determine the maximum frequency f_{max} or ω_{max} you want to catch. Use the relationship $f_{max} = \frac{1}{2\Delta t}$ or $\omega_{max} = \frac{\pi}{\Delta t}$ to find time interval Δt. Quite often, discretized p(t) is given at known Δt (i.e., f_{max} or ω_{max} is known)

1.2 Determine the analysis duration T_0 or number of excitation data points N.

This Fortran subroutine FFT requires the number of excitation data points be power of two. If not, pad zeroes to it up to the next power of two.

$T_0 = N\,\Delta t$

$\omega_0 = 2\pi / T_0$

Note: For earthquake engineering, p(n) = -m*a(n), where a(n) is ground acceleration.

Step 2 Do DFT of array p(n)

```
        CALL FFT(p, L-1,2,0,0, Δt)
```

Step 3 Compute steady-state response in frequency domain

```
C       real part of H(ω = 0):
        hwr  = 1.0/k
c       imaginary part of H(ω = 0):
        hwi  = 0
c       Compute U(ω) = P(ω)*H(ω) @ freqency ω = 0
        uw(1) = p(1)*hwr
        vw(1)=0;
c       Compute U(ω) = P(ω)*H(ω) @ intermediate frequency ω
        do j=1, j<=N/2-1
                ωj =j*ω0;
                denominator= k*1/((1-(ωj/ ωn)**2)**2+(2*ξ* ωj /ωn))**2;
c               transfer function: real part
                hwr=(1-(ωj/ ωn))**2)/denominator
c               transfer function: imaginary part
                hwi= -2*ξ*((ωj/ ωn))/denominator
c               displacement: real part
                uw(2*j+1) = p(2*j+1)*hwr-p(2*j+2)*hwi
c               displacement: imaginary part
                uw(2*j+2) = p(2*j+1)*hwi+p(2*j+2)*hwr
c               velocity: real part
                vw(2*j+1) =  −ωj*uw (2*j+2)
c               velocity: imaginary part
                vw(2*j+2) =  ωj*uw (2*j+1)
        enddo
```

c Compute real part of $H(\omega)$ @ ω = max. freqency

denominator= $k*1/((1-(N/2*\omega_0/\omega_n)**2)**2+(2*\xi* N/2*\omega_0/\omega_n))**2$;

hwr = $(1-(N/2*\omega_0/\omega_n)**2)$/denominator

c Compute $U(\omega) = P(\omega)*H(\omega)$ @ ω = max. freqency

uw(2) = p(2)*hwr

vw(2) = 0;

Step 4 Do Inverse DFT of array uw and array vw to get steady-state response in time domain.

CALL FFT(uw,L-1,2,0,1,f0)

CALL FFT(vw,L-1,2,0,1,f0)

Step 5 Add transient response to steady-state response to get total response

Note: array index starts from 1 in Matlab.

$A = u0 - us(1)$

$B = \{v(0) + \xi\omega_n[u0 - us(1)] - vs(1)\} / \omega_D$

do j=0,N-1

t=j* Δt

u(j+1)=uw(j+1)+ $e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)$

v(j+1)=vw(j+1)+ $e^{-\xi\omega_n t}(-\omega_D A\sin\omega_D t + \omega_D B\cos\omega_D t) - e^{-\xi\omega_n t}(A\cos\omega_D t + B\sin\omega_D t)\xi\omega_n$

enddo

Note: Taking complex conjugate into account, only terms corresponding to $0 \le j \le N/2$ are needed.

6 Computer Program (FrequencyDomainMethod.exe)

This program computes total response (transient and steady-state response) of a single degree of freedom system to earthquake excitation governed by the following equation of motion (EOM):

EOM of an SDOF with viscous damping	EOM of an SDOF with rate independent damping
$m\ddot{u} + c\dot{u} + ku = -ma_g$ or $\ddot{u} + 2\xi\omega_n\dot{u} + \omega_n^2 u = -a_g$	$m\ddot{u} + k(1+i\eta)u = -ma_g$ or $\ddot{u} + \omega_n^2(1+i\eta)u = -a_g$

where:

$\omega_n = \sqrt{\frac{k}{m}}$ is system natural frequency,

$\xi = \frac{c}{2m\omega_n}$ or η is damping ratio,

a_g is ground acceleration.

The steady-state response is first obtained by frequency domain method. Transient response is then computed and added to obtain total response (relative displacement, velocity, and acceleration).

This program used a subroutine FFT(…..) which is from John F. Hall (1982). "An FFT Algorithm for Structural Dynamics", Earthquake Engineering and Structural Dynamics, Vol.10, pp.797-811. This subroutine computes the forward and inverse Fourier transform by:

$$P_j = \Delta t \sum_{n=0}^{N-1} p_n e^{-i(2\pi nj/N)} \tag{41}$$

$$p_n = \frac{1}{T_0} \sum_{j=0}^{N-1} P_j e^{i(2\pi nj/N)} \tag{42}$$

There are 2 input files (input.txt and motion.txt) required. All data are in free format. Note: time unit should be in second; length unit should be the same for displacement, velocity and acceleration.

6.1

There is one line input in input.txt. They are the following six parameters in free format:
system natural Frequency in Hz,
damping,
damping type flag,
= 1, rate independent damping; = 0, viscous damping
initial displacement,
initial velocity,
motion time step Δt.

6.2

The data in motion.txt is one column data of acceleration. If the number of data points in motion.txt is not power of two, the program will automatically pad zeroes to it up to the next power of two.

6.3

MotionFFT.txt stores the Fourier Transform of the input motion. There are six columns of data corresponding to the frequency, amplitude, phase angle, real part, and imaginary part of the input motion.

6.4

Response.txt stores the relative displacement, velocity, and acceleration responses of the SDOF system to the excitation. It has 4 columns of data, the first column for time, the second column for displacement, the third for velocity, and the fourth for acceleration.

7 Computer Program (FFT.exe)

This program computes the forward Fourier Transform by:

$$P_j = \Delta t \sum_{n=0}^{N-1} p_n e^{-i(2\pi nj/N)} \tag{43}$$

Input and Ouput files:

Input.txt has two columns of data. The first column is time and the second column is function value at each time point. If number of data points is not power of two, the program will pad the data series with zeroes up to the next power of two.

Output.txt has six columns of data. They are frequency amplitude, phase angle, and real and imaginary parts of the input data.