1) $$\begin{bmatrix} 4 & 1 & 2 \\ 2 & 4 & -1 \\ 1 & 1 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ -5 \\ -9 \end{bmatrix}$$

normalized floating point representation

Using normalization of form ~~1.xxx × 10^x~~ $1.xxx \times 10^y$ - normalization doesnt change result

$$\begin{bmatrix} 4\times10^0 & 1\times10^0 & 2\times10^0 \\ 2\times10^0 & 4\times10^0 & -1\times10^0 \\ 1\times10^0 & 1\times10^0 & -3\times10^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9\times10^0 \\ -5\times10^0 \\ -9\times10^0 \end{bmatrix}$$

we were normalizing as $0.xxx \times 10^x$ in class by the normalization shouldnt change answer - wikipedia

taking $R_1/4$ =>

$$\begin{bmatrix} 1\times10^0 & 2.5\times10^{-1} & 5\times10^{-1} \\ 2\times10^0 & 4\times10^0 & -1\times10^0 \\ 1\times10^0 & 1\times10^0 & -3\times10^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^0 \\ -5\times10^0 \\ -9\times10^0 \end{bmatrix}$$

taking $R_2 - 2R_1$

$$\begin{bmatrix} 1\times10^0 & 2.5\times10^{-1} & 5\times10^{-1} \\ 0 & 3.5\times10^0 & -2\times10^0 \\ 1\times10^0 & 1\times10^0 & -3\times10^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^0 \\ -9.6\times10^0 \\ -9\times10^0 \end{bmatrix}$$

taking $R_3 - R_1$

$$\begin{bmatrix} 1\times10^0 & 2.5\times10^{-1} & 5\times10^{-1} \\ 0 & 3.5\times10^0 & -2\times10^0 \\ 0 & 7.5\times10^{-1} & -3.5\times10^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^0 \\ -9.6\times10^0 \\ -1.1\times10^1 \end{bmatrix}$$

taking $R_2 / 3.5\times10^0$

$$\begin{bmatrix} 1\times10^0 & 2.5\times10^{-1} & 5\times10^{-1} \\ 0 & 1\times10^0 & -5.7\times10^{-1} \\ 0 & 7.5\times10^{-1} & -3.5\times10^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^0 \\ -2.7\times10^0 \\ -1.1\times10^1 \end{bmatrix}$$

$R_3 - (7.5\times10^{-1})(R_2)$

$$\begin{bmatrix} 1\times10^{0} & 2.5\times10^{-1} & 5\times10^{-1} \\ 0 & 1\times10^{0} & -5.7\times10^{-1} \\ 0 & 0 & \begin{bmatrix} -3.5\times10^{0} \\ +0.42\times10^{0} \end{bmatrix} = -3.1\times10^{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^{0} \\ -2.7\times10^{0} \\ [2.0\times10^{0} + 1.1\times10^{0}] \\ [-8.97] \\ = -9.0\times10^{0} \end{bmatrix}$$

$R_3 / 3.1$

$$\begin{bmatrix} 1\times10^{0} & 2.5\times10^{-1} & 5\times10^{-1} \\ 0 & 1\times10^{0} & -5.7\times10^{-1} \\ 0 & 0 & 1\times10^{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2.3\times10^{0} \\ -2.7\times10^{0} \\ 2.9\times10^{0} \end{bmatrix}$$

from back substitution

$x_3 = 2.9\times10^{0}$

$x_2 = -2.7\times10^{0} + (0.57\times10^{-1}\times 2.9\times10^{0})$

$= -2.7\times10^{0} + 1.7\times10^{0}$

$= -1\times10^{0}$

$x_1 = 2.3\times10^{0} - (5\times10^{-1}\times 2.9\times10^{0}) - (2.5\times10^{-1}\times -1\times10^{0})$

$2.3\times10^{0} - 1.5\times10^{0} + 0.25\times10^{0}$

$= 1.05\times10^{0}$

$= 1.1\times10^{0}$

$\therefore$ Solution $=$ $x_1 = 1.1$

$x_2 = -1$

$x_3 = 2.9$

2) library

a) C- FFTW Python - numpy.fft (module)
function fftw_plan_dft_1d - numpy.fft.fft
fftw_execute

b) library C- GSL - python scipy.linalg
function gsl_linalg_QR_decomp() - scipy.linalg.qr()

c) Python - numpy.random.lognormal([mean, sigma, size])

d) Python - scipy.integrate.solve_ivp(... , method='DOP853')
by setting method to 'DOP853'

e) Python - numpy.linalg.svd(a)

f) Python - emcee. EnsembleSampler
- run_mcmc

g) Python - scipy.integrate.solve_ivp()
by Default has adaptive step-size control)

h) Python - mcint - library
function mcint.integrate

i) Python; scipy.integrate.solve_bvp()
solve_bvp can solve any number of coupled ODEs

j) Python, numpy.linalg.eig(a)

3)

$$\underbrace{\begin{bmatrix} a_1 & b_1 & & & & \\ c_1 & a_2 & b_2 & & & \\ 0 & c_2 & \cdot & & & \\ 0 & 0 & \cdot & & & 0 \\ & \cdot & \cdot & c_{n-2} & a_{n-1} & b_n \\ 0 & & 0 & 0 & c_{n-1} & a_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}}_{b}$$

~~The~~ Tridiagonal matrix is a band matrix that has non-zero elements on the main ~~Dia~~ diagonal, the first Diagonal below this, and the first diagonal above

Solving using Gaussian Elimination.

First we have to create a upper tiriangular matrix

① So ~~So to~~ steps - first Divide R_1 (Row) by a_1 &

so 2- steps $\left(\cancel{a_1}, \frac{b_1}{a_1}, \frac{d_1}{a_1}\right)$

② ~~Deleting~~ & Substating $R_2 - c_1 R_1$ - we have 3 - multiplications and 3 - substrations

∴ 6 - steps.

∴ Repeating the step ① for R_2 - $\left(\text{Dividing by} - \left(a_2 - \frac{b_1 c_1}{a_1}\right)\right)$

2-steps

, Repeating the step ② for $R_3 - R_2$ - 6- 4 steps

∴ for total Decomposition - we have N - steps of type ①

an $N-1$ - steps of type ②

∴ Total number of steps = $2N + 3(N-1)$

$= 5N - 3$ steps

③ For Back substitution - To find solutions. (b_n' and d_n' are after Decompcition)

for finding x_{n-1} - we have $d_n' - b_n'(x_n)$

which involves - 2-steps

we have ~~A - ste~~ $n-1$ steps of the form ③

$\therefore$ ~~Tota~~ Number of steps in Back-substitution $= 2N-2$

$\therefore$ Total number of steps

$= 5N-3 + 2N-2$

$\sim 7N-5$ - steps

$\therefore$ Order (n)

$\therefore$ solving a $n \times n$ tridiagonal matrix is Order (n) - $O(n)$

5) The main factors to consider while choosing a library are

1) Computation time (speed): Different ~~algorythms~~ libraries may use Different algorythms for computation. So Some libraries are faster than others., If you are going to be using the function multiple times or is a very time intensive, It ~~m~~ is better to choose faster library

2) Required Accuracy: Some libraries might be fast but may not be able to provide the accuracy required ~~by~~ for your physics. In that case it is required that we choose the slower but more accurate library. ~~A~~ The accuracy Depends on The algorythm and how it handles round-off errors.

3) Memory: Based on the memory (RAM) available to you; ~~there~~ Some libraries wont be able to work. So you have to choose the library which can work within your ~~f~~ memory tresholds for the given physics problem

Generally you have to choose a best comprimize between speed; accuracy and memory ~~, as increasing memory we would be able to get better accur~~

Other factors while choosing libraries. are:

a) licensing- Some librarys are open-source -while others are paid.

b) Exception-handling- How the libraries handle exceptions. ~~b~~ If they are coded well to handle exceptions.

c) Documentation:- A well documented library will help you considerably when you use it.

d) Community- Number of people using and working on the library.

7) Linear Congruential Method:

$$X_{i+1} = (aX_i + c) \bmod m,$$

a = The multiplier
c = Increment
m = modulus
x_0 = seed

The selection of values of a, c, m and X_0 drastically affects cycle length. (length after which the cycle repeats)

~~[illegible]~~ $X_0 < m$

Repeating sequence.
take $a=13$, $c=0$, $m=64$ and $x_0=4$

$x_0 = 4$
$x_1 = (13 \times 4 + 0) \% 64 = (52 \% 64) = 52$
$x_3 = (52 \times 13 + 0) \% 64 = (676) \% 64 = 36$
$x_4 = (36 \times 13 + 0) \% 64 = (468) \% 64 = 20$
$x_5 = (20 \times 13 + 0) \% 64 = 260 \% 64 = 4$

$\therefore$ Series = 4, 52, 36, 20, 4.

a much easier choice of repeating ~~[illegible]~~ LCG woulb be to choose:
$a=1$, $c=0$, $m=2$, ~~x_0~~

$\therefore$ $x_0 = 1$
$x_1 = 1 \% 2 = 1$
$x_2 = 1 \% 2 = 1$

$\therefore$ ~~$x_0 = 0$~~ ~~x_1~~

$\therefore$ It repeats imidiately. we can see that maximum length without repeating is m - (as after that we ~~[illegible]~~ will have no more intigers and we will repeat)

Seed never appears.

One simple example would be to

choose $m = 16,\ a = 2,\ c = 0 \quad x_0 = 8$

$\therefore\ x_0 = 8$

$x_1 = (8 \times 2 + 0)\ \%\ 16 = 0$

$x_3 = 0$

$x_4 = 0$

$\vdots$

$\therefore$ The seed Dosent appear again, but the sequence is also not random.

~~The maximum number of random numbers occurs when the seed repeats after m iterations or seed occurs~~ ~~cycle length~~

The maximum cycle length without any ~~repet~~ repetition or ~~large~~ period is m.

to get the maxium period we have to choose

m = as large as possible $\Rightarrow 2^b = 2^{32}$ (maximum 32 bit number)

c = relative prime to $m \Rightarrow$ odd number

$a = 1 + 4k$ - (k is any intiger)

4) b) The maximum frequency is given by nyquist frequency.

$$f_{max} = \frac{n-1}{n} 2\pi \left(\frac{1}{2\Delta}\right) = \frac{n-1}{n\Delta}\pi \quad \text{for our case} = 3.13$$

$$f_{min} = -2\pi \left(\frac{1}{2\Delta}\right) = -\frac{\pi}{\Delta} \quad \text{for our case} = -3.14$$

We are assuming a sampling rate of 1Hz as it is not specified in the problem, based on our experiment we will know the sampling rate from which we can find Δt as (1/sampling rate). It Doesnt Depend on n

e) As our input is a uniform distributed - the power spectrum that we expect is a fourier transform of uniform function -
we know that the fourier transform of a uniform function is a Delta function. so our result for power spectrum which is in form of Delta-function should be correct.