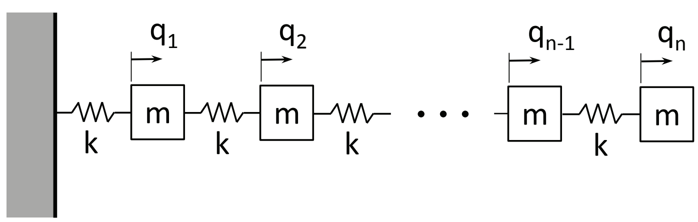
Assignment 7 2.086 Fall 2015

Due: *10 November 2015 at 5 PM.*

Upload your solution folder to Stellar as a zip file named "`YOURNAME_ASSIGNMENT_7.zip`"; the folder must include the script for each question, in the proper format, as well as any other MATLAB functions that your scripts call. **Remember to verify your upload by downloading it, and running `grade_o_matic`, before the assignment is due**. Full instructions are given in the "Preparing Assignments for `Grade_o_matic`" document.

1. **Simulating Chained Masses and Springs.** Consider the system below, containing a large number n of identical masses m, connected to each other by identical springs k, and affixed to a rigid wall at the left end. The q_i's here are taken as *dynamic* deflections from an equilibrium condition. The model represents compliant structures of many types, and exemplifies *traveling wave* behavior: a disturbance will travel across the structure with finite speed. You can observe traveling waves directly in the lateral dynamics of a rope or cable; also, in an appendix below we give some commentary about the acoustic wave equation, connecting our mass-spring drawing with the propagation of sound through a dense medium such as air or water. In this question you will simulate traveling wave behavior, by solving a large set of coupled ODE IVP's.



The first step is to write a dynamic force balance on the mass-spring system, i.e., "f = ma" for each of the masses. For instance, $m\ddot{q}_1 = -kq_1 + k(q_2 - q_1)$, and $m\ddot{q}_n = k(q_{n-1} - q_n)$. Note that any static external forces (such as mg) have already been accounted for in the equilibrium condition, and there are no dynamic external forces in this question. Assemble all of your equations into the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{0},$$

and construct the associated state-space system, for simulation:

$$\frac{d}{dt}\begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{q} \end{pmatrix} = \begin{bmatrix} \mathbf{0} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I}_{n\times n} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{q} \end{pmatrix}.$$

($\mathbf{M}^{-1}$ is the evil inverse(!), but $\mathbf{M}$ is diagonal and we only need to compute this inverse once.)

Please develop a simulation for the system, using the Matlab function `ode45` with its default settings. The five `grade_o_matic` input parameters are all scalar: the simulation final time $t_f \leftrightarrow$ `tFinal`; the number of masses `n`; the initial velocity at node n, $\dot{q}_n(t=0) \leftrightarrow$ `qdot0n`; the mass value `m`; and the spring value `k`. For the instances, assume $0 < t_f \leq 1000$, $2 \leq n \leq 200$, $0 \leq \dot{q}_n(t=0) \leq 100$, $0.001 \leq m \leq 1000$, and $0.001 \leq k \leq 1000$; all units are MKS. Note that all the initial conditions except for $\dot{q}_n(t=0)$ — which is the disturbance — should be set to zero. There are two scalar outputs: $q_1(t_f) \leftrightarrow$ `q1Final`, and $q_n(t_f) \leftrightarrow$ `qnFinal`.

2. **In More Detail.** Using your program from Question 1, please answer the following three questions in the file `A7Q2.m`; there are no `grade_o_matic` input parameters. In these computations, use the built-in Matlab ODE solver `ode45`, and fix $n = 100$ and $\dot{q}_n(t=0) = 10$m/s.

   (i) Try out several different values of k and m, and check out the pulse traveling across the structure, and reflecting off the wall. If we say that the spacing between the masses at equilibrium is Δ meters, what is the approximate wave speed, in meters per second?

   (a) $\Delta\sqrt{k/m}$
   (b) k/m
   (c) Δ
   (d) $\sqrt{m/k\Delta}$
   (e) $\sqrt{k/m}$

   (ii) Now fix $m = 1$kg, $k = 1$N/m. `ode45` adapts its stepsize to maintain a relative tolerance (e.g., $|\text{error in } y|/|y| \leq$ `RelTol`) or an absolute tolerance (e.g., $|\text{error in y}| \leq$ `AbsTol`), whichever allows the larger stepsize. You can set both `RelTol` and `AbsTol` with the `odeset` command. Fixing `RelTol` at 10^{-12}, what is the largest value of `AbsTol` for which, when the pulse arrives back at at mass n (after about 200 seconds), the peak negative displacement is accurate to better than 0.1%?

   (a) 10^{-1}
   (b) 10^{-2}
   (c) 10^{-3}
   (d) 10^{-4}

   (iii) Alternatively, fixing `AbsTol` at 10^{-12}, what is the largest value of `RelTol` for which, when the pulse arrives back at at mass n, the peak negative displacement is accurate to better than 0.1%?

   (a) 10^{-1}
   (b) 10^{-2}
   (c) 10^{-3}
   (d) 10^{-4}

3. **Integral Errors in a Second-Order System.** In feedback control systems, we often wish to tune the response of a dynamical system to initial conditions. One family of metrics, the topic of this question, is the *integral error*. Consider the classic second-order ODE IVP

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n\frac{dx}{dt} + \omega_n^2 x = 0, \text{ with i.c.'s } \dot{x}_0 \equiv \frac{dx}{dt}(t=0),\ x_0 \equiv x(t=0).$$

We are writing with the standard form, wherein ω_n is the undamped natural frequency (radians per second), and ζ is the damping ratio (dimensionless). Variable $x(t)$ (which could be a position, a voltage, a pressure, etc.) is interpreted as an error signal, and the controller's objective is to drive it to zero.

Please answer the following three questions, providing the requested values in the file `A7Q3.m`. In the computations, set $x_0 = 1$, $\dot{x}_0 = 0$, and $\omega_n = 1$.

(a) To two significant digits, what positive value of ζ minimizes $\int_{t=0}^{\infty} |x(\tau)| d\tau$ (known as the integral of absolute error, IAE)? Call this output `zeta1`.

(b) To two significant digits, what positive value of ζ minimizes $\int_{t=0}^{\infty} x^2(\tau) d\tau$ (the integral of squared error, ISE, which penalizes large deviations) ? Call this output `zeta2`.

(c) To two significant digits, what positive value of ζ minimizes $\int_{t=0}^{\infty} t x^2(\tau) d\tau$ (the integral of time times squared error, ITSE, which penalizes large deviations at large times) ? Call this output `zeta3`.

We recommend that, for a set of ζ's, you solve the ODE IVP's numerically, view the results, and calculate the integrals of interest. You are responsible for selecting a viable ODE solver, ensuring that its errors are sufficiently small to provide the optimal ζ's to two significant digits, and that making sure the simulation times are long enough (note that the upper limits in our integrals are infinity).

4. **Flying Through a Tornado.** The trajectory of an inertial object moving through a complex flow field is relevant to a number of industrial systems, including separators and vacuums. A useful dynamic model is not difficult to write:

$$\frac{d\vec{v}}{dt} = \alpha \left[\vec{u}(\vec{r}) - \vec{v} \,\right] ||\vec{u}(\vec{r}) - \vec{v}\,||$$
$$\frac{d\vec{r}}{dt} = \vec{v},$$

where $\vec{v}$ ($= [v_x \; v_y \; v_z]^{\mathrm{T}}$) is the three-vector of the particle velocity through Cartesian space and α is a scalar constant that scales drag versus inertia. $\vec{u}(\vec{r})$ is the (vector) *flow* velocity at the particle's location $\vec{r}$, so that $\vec{u}(\vec{r}) - \vec{v}$ is the flow velocity *as seen by the particle*. Note that the particular drag model here involves the square of relative velocity, and may not be valid for all flow conditions and length scales. Similarly, the model assumes a spherical (non-directional) particle; an airplane, for instance, would not respond to wind gusts in such a simple way.

In this question, we ask you to simulate the sixth-order ODE system above (with state vector $[\ \vec{v}\ ;\ \vec{r}\ ]$), for given initial conditions and a given flow field, and report the location of the particle at a particular time. For the flow field, we specify a fearsome vortex that generates flow velocity $\vec{u}$ depending on $\vec{r}$ as follows:

$$u_x(\ \vec{r}\ ) = 10\ (r_z - r_y)\ /\ ||\ \vec{p}\ ||^2$$
$$u_y(\ \vec{r}\ ) = 10\ (r_x - r_z)\ /\ ||\ \vec{p}\ ||^2$$
$$u_z(\ \vec{r}\ ) = 10\ (r_y - r_x)\ /\ ||\ \vec{p}\ ||^2,$$

where

$$\vec{p} \equiv \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \vec{r}.$$

This vortex is a spinning structure, whose center axis passes through $\vec{r} = \vec{0}$, and is parallel to the vector $[1, 1, 1]$ in Cartesian space — you can think of it as a tilted tornado. More specifically, our flow model is an *irrotational vortex*, for which you can easily look up the technical conditions if you like. $|| \, \vec{p} \, ||$ is the distance from the vortex centerline out to $\vec{r}$, and Yes, velocity $\vec{u}$ is undefined on the centerline!

There are four `grade_o_matic` input parameters: the initial position of the particle $\vec{r}(t = 0) \leftrightarrow$ `r0` (a 3×1 vector); the initial velocity of the particle $\vec{v}(t = 0) \leftrightarrow$ `v0` (a 3×1 vector); the drag-to-inertia scale constant $\alpha \leftrightarrow$ `alpha`; and the final simulation time $t_f \leftrightarrow$ `tFinal`. For the instances, assume $||\vec{r}(t = 0)|| \leq 100$, $||\vec{v}(t = 0)|| \leq 100$, $0 < t_f \leq 100$, and $0 \leq \alpha \leq 1$. The single output variable is position of the particle at the end of the simulation, $\vec{r}(t_f) \leftrightarrow$ `rFinal` (a 3×1 vector). Use `ode45` with its default tolerances.

(Ungraded) Sidelight: Acoustics Interpretation of Question 1

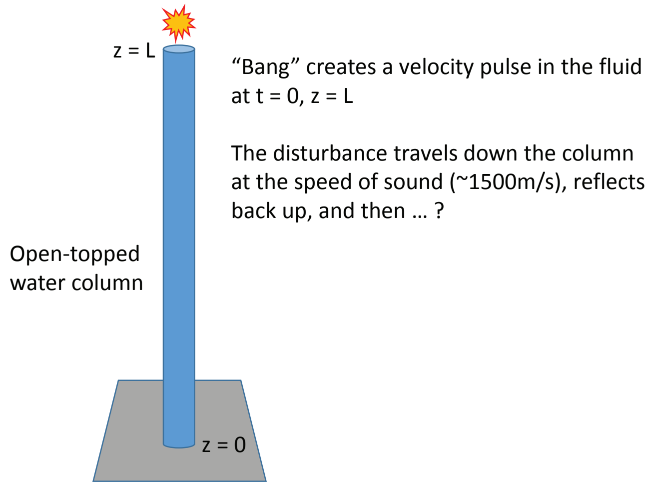
Compressibility of a dense medium – be it air, water, earth or steel – gives rise to pressure waves that are at the heart of acoustics, that is, sound. The acoustic wave equation in one dimension is a PDE:

$$\rho \, \frac{\partial^2 q}{\partial t^2} = E \, \frac{\partial^2 q}{\partial z^2} \; ,$$

where z is the spatial dimension, t is time, $q(t, z)$ is the (typically small) displacement of a particle, ρ is density of the medium, and E the bulk modulus (Young's modulus in elastic solids). The acoustic equation is a specialization of the wave equation, which is extremely interesting from a mathematical point of view, and is far-reaching in applications well beyond compressible media. You can solve the acoustic equation numerically, by applying standard differentiation formulas in the spatial domain, and then solving the consequent system of ODE's in the time domain.

First, we need to define the physical scenario, and in particular what is happening at the boundaries within which the acoustic equation applies. Our description below mimics that of Question 1. It is helpful perhaps to visualize a long vertical column of water, with a closed bottom and an open top.

- At the bottom end, $z = 0$, we take the particle position at all times to be zero, i.e., $q(t \geq 0, z = 0) = 0$. For water in a column, this implies a hard wall at the lower end.
- At the upper end, $z = L$, we impose the condition of zero pressure at all positive times. Since the fluid is compressible, this is the same as having zero spatial derivative in q at $x = L$, i.e., $q\,'(t > 0, x = L) = 0$. It means the upper water interface is unrestricted and open to the air.



- Now for the initial conditions. At time zero, we set all $q(t = 0, 0 \leq z \leq L) = 0$, i.e., the system is at its equilibrium position. We set the velocity to zero everywhere too, EXCEPT at $z = L$, where the initial velocity is $\dot{q}_{0,L}$. This description captures a system that is completely at rest, but just before time zero an impulsive force — i.e., a bang! — is applied at the upper end, giving the local velocity $\dot{q}_{0,L}$. What provides the bang? A transducer, an explosion, a diver, etc. The initial velocity conditions are thus $\dot{q}(t = 0, 0 \leq z < L) = 0$ and $\dot{q}(t = 0, z = L) = \dot{q}_{0,L}$.

Our next step is a spatial discretization into n nodes, at $z_i = i\Delta z$, $i = 1, \cdots, n$, where $\Delta z = L/n$. Apply the centered-difference formula for the second spatial derivative at nodes 1 through $n - 1$, and then the backward-difference formula for the first derivative at node n (the top). Recalling that E and Δz encode a stiffness, the result from these spatial discretizations is very familiar from Assignment 4.

Following the above process will yield a set of equations of the same form as in Question 1 of this assignment. From the point of view of the acoustic equation, oscillatory behavior you observe in the simulation output is an *artifact* of the spatial discretization; the oscillations would shrink if you increased n, and they would not appear in an exact solution. You may be interested to know that for seawater, $\rho \approx 1030\text{kg/m}^3$ and $E \approx 2.2 \times 10^9\text{Pa}$. For the air we breathe, $\rho \approx 1.2\text{kg/m}^3$ and $E \approx 1.4 \times 10^5\text{Pa}$; for steel, $\rho \approx 7600\text{kg/m}^3$ and $E \approx 200 \times 10^9\text{Pa}$. The speed of sound in the acoustic equation is $\sqrt{E/\rho}$.