MATH 231: Differential Equations with Linear Algebra

Hand-Checked Assignment #3, due date: Tues., Mar. 23, 2021

Write up, carefully and legibly, your solutions to the following problems. While you do not need to present one problem per page, please do not put problems side-by-side (i.e., no two-column format). To submit your work it must be

- scanned (all pages) to a single .pdf file (one multi-page file containing all graded problems).
- submitted to `https://www.gradescope.com` as **hc03**.

$\star$23 **A Basis for the Null Space of the 3-by-7 Hamming Matrix**. Consider the set $\mathbb{Z}_2^n$. The objects in this set are n-by-1 matrices (in that respect they are like the objects in $\mathbb{R}^n$), with entries that are *all zeros or ones*; each object in $\mathbb{Z}_2^n$ can be thought of as an n-bit binary word.

We wish to define what it means to *add* objects in $\mathbb{Z}_2^n$, and how to multiply these objects by a reduced list of scalars—namely 0 and 1. When we add vectors from $\mathbb{Z}_2^n$, we do so componentwise (as in $\mathbb{R}^n$), but with each sum calculated mod 2.[1] Scalar multiplication is done mod 2 as well. For instance, in $\mathbb{Z}_2^3$ we have

$$\begin{bmatrix}1\\0\\1\end{bmatrix} + \begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\end{bmatrix}, \qquad 0\begin{bmatrix}1\\0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix} \qquad \text{and} \qquad 1\begin{bmatrix}1\\0\\1\end{bmatrix} = \begin{bmatrix}1\\0\\1\end{bmatrix}.$$

Note that, when operations are performed mod 2, an m-by-n matrix times a vector in $\mathbb{Z}_2^n$ produces a vector in $\mathbb{Z}_2^m$. For instance

$$\begin{bmatrix}3 & 1 & 2\\1 & 6 & 0\\2 & 0 & 1\\7 & 2 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}, \quad \text{and is equivalent to} \quad \begin{bmatrix}1 & 1 & 0\\1 & 0 & 0\\0 & 0 & 1\\1 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}.$$

Consider the matrix

$$\mathbf{H} := \begin{bmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1\\0 & 1 & 1 & 0 & 0 & 1 & 1\\1 & 0 & 1 & 0 & 1 & 0 & 1\end{bmatrix}.$$

[1] Modular arithmetic is the type of *integer* arithmetic we use with clocks. For a standard clock, the *modulus* is 12, resulting in statements like “It is now 8 o’clock; in 7 hours it will be 3 o’clock” (i.e., “8 + 7 = 3”). In mod 2 arithmetic, the modulus is 2, and we act as if the only numbers on our “clock” are 0 and 1.

An easy way to remember this matrix, known as the **Hamming Matrix**, is through noting that beginning from its left column you have, in sequence, the 3-bit binary representations of the integers 1 through 7. Find a basis for null $(\mathbf{H})$, where the matrix product $\mathbf{Hx}$ is to be interpreted mod 2 as described above.

A couple of observations may be helpful. First, if you had a 2-by-5 matrix with entries from $\mathbb{Z}_2$ such as this one

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

the next step in Gaussian elimination would be to zero out the rest of column 2 under the pivot. You can do this by adding row 1 to row 2—that is:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

At this point, this 2-by-5 matrix has reached echelon form (not quite RREF, yet).

Secondly (but related), in $\mathbb{Z}_2$ each of the two possible numbers (0 and 1) are their own additive inverses. That is,

$$0 + 0 = 0 \qquad \text{and} \qquad 1 + 1 = 0.$$

This means that, when you have a variable x that represents a number in $\mathbb{Z}_2$, then $x + x = 0$. So, if you have a $\mathbb{Z}_2$ equation which says

$$x_1 + x_3 + x_4 = 0,$$

you can *add* x_3 and x_4 to both sides to get

$$x_1 = x_3 + x_4.$$

Bizarre, yet kinda cool, too.

$\star$24 **Error-Correcting Codes: The Hamming (7,4) Code**. In this problem, we wish to look at a method for transmitting the 16 possible 4-bit **binary words**

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

in such a way that if, for whatever reason (perhaps electrostatic interference), some digit is reversed in transmission (a 0 becomes a 1 or vice versa), then the error is *both* detected and corrected.

In the previous problem, you found a basis for the null space of the Hamming matrix

$$\mathbf{H} := \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

In most instances, a vector space has numerous different bases, and null$(\mathbf{H})$ is no exception. Though it is likely a different collection of basis vectors than you found in the previous problem, going forward we will make use of the basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$, where $\mathbf{u}_1 = (1,0,0,0,0,1,1)$, $\mathbf{u}_2 = (0,1,0,0,1,0,1)$, $\mathbf{u}_3 = (0,0,1,0,1,1,0)$, and $\mathbf{u}_4 = (0,0,0,1,1,1,1)$.

Transmitting a 4-bit Word

Let (c_1, c_2, c_3, c_4) be a 4-bit word (i.e., each c_i is 0 or 1), one we wish to transmit. We could do so as is, but if an error occurred in transmission, there would be no automatic indicator of this. Instead, we use the values $c_1, \ldots, c_4$ to generate a 7-bit word which will be the one we transmit. This 7-bit word is a linear combination (mod 2) of $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$. To be precise, instead of the original 4-bit word, we transmit the 7-bit word

$$\mathbf{v} := c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + c_3\mathbf{u}_3 + c_4\mathbf{u}_4 = (c_1, c_2, c_3, c_4, c_2 + c_3 + c_4, c_1 + c_3 + c_4, c_1 + c_2 + c_4).$$

This $\mathbf{v}$ is in both $\mathbb{Z}_2^7$ and null$(\mathbf{H})$. (Do you see why it is an element of the latter?)

(a) Suppose we wish to transmit the 4-bit binary word 1101. What 7-bit word corresponding to this one will actually be transmitted?

Error Detection and Correction

Suppose a 7-bit word $\tilde{\mathbf{v}}$ is received. It may be the same as the transmitted $\mathbf{v}$, or it may be a corrupted version of $\mathbf{v}$. Suppose that at most one binary digit of $\tilde{\mathbf{v}}$ is in error. Then the matrix product $\mathbf{H}\tilde{\mathbf{v}}$ tells us what we need to know. To see this, consider two cases:

- There are no errors (that is, $\tilde{\mathbf{v}} = \mathbf{v}$).
  In this case, $\mathbf{H}\tilde{\mathbf{v}} = \mathbf{H}\mathbf{v} = \mathbf{0}$, and the receiver, who takes this as an indication that the word arrived uncorrupted, throws out the final 3 bits and keeps the first 4 (entries) of $\mathbf{v}$ as the 4-bit word originally intended.
- There is an error in position i (so $\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{e}_i$, where $\mathbf{e}_i$ is a vector of zeros except in its i^{th} position, where it has a 1).
  In this case, $\mathbf{H}\tilde{\mathbf{v}} = \mathbf{H}(\mathbf{v} + \mathbf{e}_i) = \mathbf{H}\mathbf{v} + \mathbf{H}\mathbf{e}_i = \mathbf{0} + \mathbf{H}\mathbf{e}_i = \mathbf{H}\mathbf{e}_i = i$th column of $\mathbf{H}$. Thus, $\mathbf{H}\tilde{\mathbf{v}} \neq \mathbf{0}$ in this case. Moreover, by inspecting which column of $\mathbf{H}$ is equal to $\mathbf{H}\tilde{\mathbf{v}}$, we learn which of $\tilde{\mathbf{v}}$'s digits is different from those of $\mathbf{v}$. The receiver may

correct that bit in $\tilde{\mathbf{v}}$, and once again take the first 4 bits of this (newly-corrected) $\tilde{\mathbf{v}}$ as the intended word.

(b) For practice (i.e., **don't hand in this first bit of work**), take your answer from part (a)—call this $\mathbf{v}$—and switch/corrupt the 1^{st} entry (binary digit), calling this new 7-bit word $\tilde{\mathbf{v}}$. Calculate $\mathbf{H}\tilde{\mathbf{v}}$, and use the procedure outlined above to convince yourself that the corrupted bit can be detected and corrected. Repeat this several times, corrupting some other bit of $\mathbf{v}$ to form $\tilde{\mathbf{v}}$.

Now, for something **to write up**: Suppose that the 7-bit word $(1, 0, 1, 1, 1, 0, 0)$ is received. Assuming that this was originally a 4-bit word that was sent using the Hamming (7,4) error-correcting code, and assuming at most one binary digit becomes corrupted during transmission, what was the original 4-bit word?

(c) {This part is optional.} What happens if more than one bit of the 7-bit (transmitted) word is corrupted? Investigate this question, and see if the procedure outlined above can be relied upon to *detect* and, if so, *correct* two corrupted bits in the transmitted word. Report on your findings.

$\star$25 From Section 1.12, pp. 82–84, do Exercise 1.12.3, parts (b) and (c).

$\star$26 Write down a differential equation of the form $dy/dx = ay + b$ whose solutions approach $y = 2/3$ as $x \to \infty$. Demonstrate that the solutions, indeed, do what they are supposed to do.

$\star$27 From Section 1.6, pp. 37–40, do Exercise 1.6.8 parts (a) and (b).

$\star$28 Consider the IVP: $2y' - y = e^{t/3}$, subject to $y(0) = a$.

(a) Look at a direction field for the differential equation. What appears to be the *end behavior* of solutions as t (the independent variable) becomes large? Do solutions go to $+\infty$ as $t \to +\infty$? Do they approach a limit? Does the behavior depend on the choice of the initial value a?

(b) Let a_0 be the value of a for which the transition from one type of end behavior to another occurs. Estimate a_0.

(c) Solve the IVP and confirm or revise your estimate of a_0.

(d) Describe the behavior of the solution as $t \to \infty$ when the initial condition is $y(0) = a_0$.

$\star$29 (a) Rewrite the initial value problem in vector form $\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x}$, where $\mathbf{x}(t) = (y_1(t), y_2(t), y_3(t))$. That is, determine the correct entries of the matrix $\mathbf{A}$, and state the initial condition $\mathbf{x}(0)$.

$$\begin{aligned} y_1' &= 6y_1 + 4y_2 + 4y_3 & \qquad y_1(0) &= 3 \\ y_2' &= -7y_1 - 2y_2 - y_3 & \qquad y_2(0) &= -6 \\ y_3' &= 7y_1 + 4y_2 + 3y_3 & \qquad y_3(0) &= 4 \end{aligned}$$

(b) Verify that the vector function

$$\mathbf{x}(t) = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} e^{6t} + 2 \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} e^{2t} + \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} e^{-t}$$

is a solution of your system from part (a). (You must show both that this vector function satisfies the DE system, and the IC.)

★30 Solve the given problem. When you rely on technology, you should indicate what task you assigned to software, and what results were returned to you.

(a) $\dfrac{d}{dt}\mathbf{x} = \begin{bmatrix} -2 & 1 \\ -1 & -4 \end{bmatrix} \mathbf{x}$
(This is Exercise 3.5.11, p. 174.)

(b) $\dfrac{d}{dt}\mathbf{x} = \begin{bmatrix} 6 & 4 & 8 \\ 4 & 0 & 4 \\ 8 & 4 & 6 \end{bmatrix} \mathbf{x}$

(c) $\dfrac{d}{dt}\mathbf{x} = \begin{bmatrix} 5 & 2 & -1 \\ -3 & 2 & 2 \\ 1 & 3 & 2 \end{bmatrix} \mathbf{x}$, subject to $\mathbf{x}(0) = \begin{bmatrix} 4 \\ 0 \\ 6 \end{bmatrix}$.

(d) $\dfrac{d}{dt}\mathbf{x} = \begin{bmatrix} 5.5 & -1.5 & 1.5 \\ 0.5 & -0.5 & -1.5 \\ -1 & -1 & 2 \end{bmatrix} \mathbf{x}$