Lecture 2: Row reduction

Using the “Replace”, “Swap”, and “Scale” row-operations, we shall now discuss how to put any matrix in a particularly nice form:

Reduced Row-Echelon Form (RREF)

A matrix $A$ is in RREF if all of the following hold:

(i) the first nonzero entry of each row is 1, called a “leading 1”;

(ii) when a column contains a leading 1, all other entries in that column are 0 (this is called a “pivot column”); and

(iii) when a row contains a leading 1, each row above it contains a leading 1 to the left.

The weaker notion of Row-Echelon Form (REF) is obtained by dropping (i) and weakening (ii) and (iii):

(ii') when a column contains the first nonzero entry of some row, all the entries of the column
below it are 0; and

(iii') the leading nonzero entry of a row occurs further to the right than all the leading entries in the rows above it.

Ex 1/ If "*" stands for "arbitrary numbers", then

\[
\begin{pmatrix}
* & 0 & * & * \\
0 & 0 & \ast & \ast \\
0 & 0 & 0 & 0
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
1 & 0 & * & * \\
0 & 1 & \ast & \ast \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

are in reduced row-echelon form (RREF), while (if "*" stands for "arbitrary nonzero number")

\[
\begin{pmatrix}
* & * & * & * \\
0 & 0 & \ast & * \\
0 & 0 & 0 & 0
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
* & * & * & * \\
* & * & * & * \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

are in row-echelon form (REF)

Row reduction means "some procedure for associating an RREF matrix to a given matrix A":

\[ A \rightarrow \text{rref}(A). \]

Since the procedure simply applies row operations to A, the new matrix is row-equivalent to A.

**FACT**: There is exactly one RREF matrix row-equivalent to a given matrix A.
CONSEQUENCE: \( \text{ref}(A) \) is independent of the procedure used!

The book has a 2-stage row-reduction algorithm:
- convert \( A \) to a REF matrix
- convert the REF matrix to a RREF one.
(See the "appendix" below.) Here's a simpler version:

Row-reduction algorithm

(a) "Cursor" starts at upper left-hand entry of matrix;
(b) move cursor to right if the cursor entry and all entries below it are 0; repeat until this is no longer the case;
(c) if cursor entry = 0, swap cursor row with the first row below it having nonzero entry in the cursor column;
(d) divide cursor row by cursor entry;
(e) eliminate all other entries in the cursor column by adding suitable multiples of the cursor row to other rows;
(f) if cursor is at bottom right, stop. Otherwise, move down & to the right, and go back to (b).
Let's illustrate with an example: \[ \square = \text{cursor} \]

Ex 2: 

\[ A = \begin{pmatrix} 0 & 1 & -1 & -1 \\ 2 & 4 & 2 & 4 & 2 \\ 2 & 4 & 3 & 3 & 3 \\ 3 & 6 & 6 & 3 & 6 \end{pmatrix} \overset{(c)}{\rightarrow} \begin{pmatrix} 2 & 4 & 4 & 2 \\ 0 & 0 & 1 & -1 & -1 \\ 2 & 4 & 3 & 3 & 3 \\ 3 & 6 & 6 & 3 & 6 \end{pmatrix} \overset{(d)}{\rightarrow} \begin{pmatrix} 2 & 1 & 2 & 1 \\ 0 & 0 & 1 & -1 & -1 \\ 2 & 4 & 3 & 3 & 3 \\ 3 & 6 & 6 & 3 & 5 & 6 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 2 & 1 & 2 & 1 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 3 & -3 & 3 \end{pmatrix} \overset{(e)}{\rightarrow} \begin{pmatrix} 1 & 2 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \overset{(f)}{\rightarrow} \begin{pmatrix} 1 & 2 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 2 & 0 & 3 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \text{rref}(A). \]

Now, how do we use this to solve a linear system?

**Step 1:** Convert the system to an augmented matrix 

\[ M = [A \mid \mathbf{b}] \]

**Step 2:** Apply the above algorithm to compute 

\[ \text{rref}(M). \]

**Step 3:** Convert back to a linear system and find the tuples \((x_1, \ldots, x_n)\) solving it. (More precisely: use the non-first variables to parameterize the solution set. This is sometimes called "back substitution".)
Ex 3/ \[
\begin{align*}
3x_1 - 6x_2 + 2x_3 - x_4 &= 1 \\
-2x_1 + 4x_2 + x_3 + 3x_4 &= 4 \\
x_3 + x_4 &= 2 \\
x_1 - 2x_2 + x_3 &= 1
\end{align*}
\]

**Step 1**

\[
M = \begin{bmatrix}
3 & -6 & 2 & -1 \\
-2 & 4 & 1 & 3 \\
0 & 0 & 1 & 1 \\
1 & -2 & 1 & 0
\end{bmatrix}
\text{scale } \begin{bmatrix} -2 & 2/3 & -1/3 \\
4 & 1 & 3 \\
2 & 0 & 1 \\
1 & -2 & 1 & 0
\end{bmatrix}
\]

**Step 2**

\[
\begin{bmatrix}
1 & -2 & 2/3 & -1/3 \\
0 & 0 & 2/3 & 2/3 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1/3 & 1/3
\end{bmatrix}
\text{scale } \begin{bmatrix} 1 & -2 & 2/3 & -1/3 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1/3 & 1/3 \\
0 & 0 & 1/3 & 1/3
\end{bmatrix}
\]

**Step 3**

\[
\text{ref } (M)
\]

Solve for the variables corresponding to pivot columns, called basic variables:

\[
\begin{align*}
-x_1 - 2x_2 & = 1 \\
x_3 - x_4 & = 2
\end{align*}
\]

\[
\begin{align*}
x_1 = 2x_2 + x_4 - 1 \\
x_3 = 2 - x_4
\end{align*}
\]
The free variables are the non-prime ones, and are so named because they can be freely chosen. They parameterize the solution set of the linear system:

\[ S = \{(2x_2 + x_4 - 1, x_2, 2 - x_4, x_4) \mid x_1, x_4 \in \mathbb{R}\} \]

Ex 4

\[ \begin{align*}
3x_1 - 6x_2 + 2x_3 - x_4 &= 1 \\
-2x_1 + 4x_2 + x_3 + 3x_4 &= 4 \\
x_3 + x_4 &= 2 \\
x_1 - 2x_2 + x_3 &= 0
\end{align*} \]

Some by Ex. 3 except for two

\[
\begin{bmatrix}
3 & -6 & 2 & -1 & 1 \\
-2 & 4 & 1 & 3 & 4 \\
0 & 0 & 1 & 1 & 2 \\
1 & -2 & 1 & 0 & 0
\end{bmatrix}
\]

Steps from Ex. 3

\[
\begin{bmatrix}
1 & -2 & 0 & -1 & -1 \\
0 & 0 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Scale \( x_3 \rightarrow -1 \)

\[
\begin{bmatrix}
1 & -2 & 0 & -1 & 0 \\
0 & 0 & 1 & 1 & 2 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Replace \( x_1 - 2x_2 + x_3 \rightarrow x_1 - 2x_2 + x_4 = 0 \)

\[
\begin{bmatrix}
1 & -2 & 0 & -1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Solution set \( S = \emptyset \)

System is consistent!
From the last example, we notice that
• If the last column of the augmented matrix is a pivot column, then the system is inconsistent.

Otherwise, steps 1–3 tell us how to solve the system, and so:
• If the last column is non-pivot, the system is consistent; it has a unique solution if there are no free variables, i.e. if all but the last column are pivots.

So far we have looked at \( m \) equations in \( n \) unknowns, with \( m = n \). What about the other cases?
• If \( m > n \), the system is called overdetermined.
• If \( m < n \), the system is called under-determined.

**Q:** Can an overdetermined system be consistent?
**A:** YES. But some equations will have to be “linear combinations” of others.

**Q:** Can an underdetermined system have a unique solution?
**A:** NO. Think about intersection of planes in space: two planes in 3-space will never intersect in a point.

A more mathematical answer can be formulated as follows: when you reduce an augmented matrix of the form
to RREF, there must be non-pivot columns, hence free variables. This is because each pivot column contains a leading 1 for some row, and there are only m rows hence \( \leq n \) leading 1's. In general, the number of pivots is at most the smaller of m & n.

APPENDIX (The book's row-reduction algorithm)

**PART I** Produce REF matrix:

I(a): “cursor” starts at upper left-hand entry.

I(b): if necessary, move cursor to right until you reach a nonzero column.

I(c): if cursor entry = 0, exchange cursor row with 1st row below if having nonzero entry in cursor column.

\( \text{SWAP} \)

I(d): eliminate entries below the cursor (in cursor column) by adding multiples of cursor row to rows below.

\( \text{REPLACE} \)

I(e): move the cursor down & to the right, hide all rows above it & columns to the left of it, go back to I(b).
PART II  REF to RREF:

II (a): "Cursor" starts at right most leading entry

II (b): divide cursor row by cursor entry

II (c): eliminate entries above cursor (by the "replace" operation)

II (d): move cursor left & up to the next leading entry,  
go back to II (b).

Again, stop when you exit the matrix.