Homework 2

1

Given that a linear system in the unknowns x_1, x_2, x_3, x_4 has general solution $(x_2 + 3x_4 + 4, x_2, 2 - x_4, x_4)$ for free variables x_2, x_4, find a minimal reduced row echelon for this system.

Solution:

Exercise 2.1.13

We know that there are at least two equations in the system (since we have two constraints in the general solution.) We can write these two constraints as:

$$R = \begin{bmatrix} 1 & -1 & 0 & -3 & 4 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

We could also have written

$$R = \begin{bmatrix} 1 & -1 & 0 & -3 & 4 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$, which has the same solutions.

Just for kicks, checking in Sympy:

```
from sympy import print_latex
def my_print(x, *args, **kwargs):
    print_latex(x, itex=False, mode='equation', *args, **kwargs)
```

```
from sympy import Matrix
R=Matrix([[1, -1, 0, -3], [0, 0, 1, 1]])
rhs = Matrix([4,2])
R.gauss_jordan_solve(rhs)
my_print(R.gauss_jordan_solve(rhs))
```

```
\begin{equation}\left( \left[\begin{matrix}\tau_{0} + 3 \tau_{1} + 4\\\tau_{0}\\2 - \tau_{1
```

```
from sympy import Matrix, print_latex
R=Matrix([[1, -1, 0, -3], [0, 0, 1, 1], [0, 0, 0, 0], [0, 0, 0, 0]])
rhs = Matrix([4,2,0,0])
my_print(R.gauss_jordan_solve(rhs))
```

$$\left(\begin{bmatrix} \tau_0 + 3\tau_1 + 4 \\ \tau_0 \\ 2 - \tau_1 \\ \tau_1 \end{bmatrix}, \begin{bmatrix} \tau_0 \\ \tau_1 \end{bmatrix} \right) \tag{1}$$

2

Use the technique of Example 2.10 in your textbook to balance the following chemical equation:

$$C_8H_{18} + O_2 \rightarrow CO_2 + H_2O.$$

Solution:

Exercise 2.2.23

With vectors indicating amount of C, H and O, variables the number of molecules of each compound occurring, system represented is

$$x_1 \begin{bmatrix} 8 \\ 18 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = x_3 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} + x_4 \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$$

resulting in coefficient matrix

$$\begin{bmatrix} 8 & 0 & -1 & 0 \\ 18 & 0 & 0 & -2 \\ 2 & 2 & -2 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & \frac{1}{9} \\ 0 & 1 & 0 & \frac{-23}{18} \\ 0 & 0 & 1 & \frac{-8}{9} \end{bmatrix}$$

The smallest integer solution is $x_1 = 2, x_2 = 23, x_3 = 16, x_4 = 18$.

3

Express the following functions, if linear, as matrix operators. (If not linear, explain why.)

(a) $T((x_1, x_2)) = (x_1 + x_2, 2x_1, 4x_2 - x_1)$

(b) $T((x_1, x_2)) = (x_1 + x_2, 2x_1x_2)$

(c) $T((x_1, x_2, x_3)) = (2x_3, -x_1)$

(d) $T((x_1, x_2, x_3)) = (x_2 - x_1, x_3, x_2 + x_3)$

Solution:

Exercise 2.3.3

Operator is T_A with:

(a) $A = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 4 & -1 \end{bmatrix}$

(b) nonlinear

(c) $\begin{bmatrix} 0 & 0 & 2 \\ -1 & 0 & 0 \end{bmatrix}$(

d) $A = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

4

A *fixed-point* of a linear operator T_A is a vector $\mathbf{x}$ such that $T_A(\mathbf{x}) = \mathbf{x}$. Find all fixed points, if any, of the linear operators in the previous exercise.

Solution:

Exercise 2.3.9

We require our matrix to be square (i.e., have the same number of rows and columns) to have a fixed point. For the matrices in the previous exercise, only the matrix in part (d) is square. The fixed points are the solutions to the system of equations

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

First, we subtract the right-hand side from the left-hand side to get

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

This simplifies to

$$\begin{bmatrix} -2 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Putting our matrix into reduced row echelon form, we have

$$\begin{bmatrix} -2 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This is the identity matrix, which means that the system of equations is nonsingular and has a unique solution. Moreover, the right-hand side of the equation is the zero vector, which means that the fixed point is the zero vector: $(x_1, x_2, x_3) = (0, 0, 0)$.

```
from sympy import eye
#| echo: false
#| output: none

A = Matrix([[-1, 1, 0], [0, 0, 1], [0, 1, 1]])
(A-eye(3))
# append a zero vector to the right of the matrix
R = (A-eye(3)).row_join(Matrix([0,0,0]))
R.rref()
```

```
(Matrix([
 [1, 0, 0, 0],
 [0, 1, 0, 0],
```

```
  [0, 0, 1, 0]]),
  (0, 1, 2))
```

5

A linear operator on $\mathbb{R}^2$ is defined by first applying a scaling operator with scale factors of 2 in the x-direction and 4 in the y-direction, followed by a counterclockwise rotation about the origin of $\pi/6$ radians. Express this operator and the operator that results from reversing the order of the scaling and rotation as matrix operators.

Solution:

Exercise 2.3.5 The scaling operator is given by the matrix

$$\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

To create a counterclockwise rotation by θ radians, we use the matrix

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Here, we want to rotate by $\pi/6$ radians, so we have

$$\begin{bmatrix} \cos\frac{\pi}{6} & -\sin\frac{\pi}{6} \\ \sin\frac{\pi}{6} & \cos\frac{\pi}{6} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

The operator for the scaling followed by the rotation is the product of these two matrices:

$$T_A, A = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} = \begin{bmatrix} \sqrt{3} & -2 \\ 1 & 2\sqrt{3} \end{bmatrix}$$

The operator for the rotation followed by the scaling is the product of the matrices in reverse order:

$$T_B, B = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \sqrt{3} & -1 \\ 2 & 2\sqrt{3} \end{bmatrix}$$

6

Find a scaling operator S and shearing operator H such that the concatenation $S \circ H$ maps the points $(1, 0)$ to $(2, 0)$ and $(0, 1)$ to $(4, 3)$.

Solution:

Exercise 2.3.7

We know that the shearing operator must add a multiple of the x_2 component to the x_1 component. For now, we can write this as

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

Then we can apply the shearing operator to our two points to get

$$\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ 1 \end{bmatrix}$$

We know our scaling operator will be of the form

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix}$$

Applying this to our two points, we get

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} s \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} \alpha \\ 1 \end{bmatrix} = \begin{bmatrix} s\alpha \\ t \end{bmatrix}$$

We can now set up the equations

$$s = 2, 0 = 0$$
$$\alpha s = 4, t = 3$$

It is clear that the solutions to these equations are $s = 2, t = 3, \alpha = 2$. Thus, our scaling operator is

$$S = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

and our shearing operator is

$$H = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

Checking:

```
S = Matrix([[2, 0], [0, 3]])
H = Matrix([[1, 2], [0, 1]])
S*H*Matrix([1,0]), S*H*Matrix([0,1])
```

```
(Matrix([
 [2],
 [0]]),
 Matrix([
 [4],
 [3]]))
```

7

Given transition matrices for discrete dynamical systems

(a) $\begin{bmatrix} .1 & .3 & 0 \\ 0 & .4 & 1 \\ .9 & .3 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} .5 & .3 & 0 \\ 0 & .4 & 0 \\ .5 & .3 & 1 \end{bmatrix}$ (d) $\begin{bmatrix} 0 & 0 & 0.9 \\ 0.5 & 0 & 0 \\ 0 & 0.5 & 0.1 \end{bmatrix}$ and initial state vector $\mathbf{x}^{(0)} = \frac{1}{2}(1, 1, 0)$, calculate the first and second state vector for each system and determine whether it is a Markov chain.

Solution:

Exercise 2.3.11

We know going in that (d) is not a stochastic matrix, because the neither the first or second column sums to 1. The others are stochastic.

First and second states are (a) $(0.2, 0.2, 0.6)$, $(0.08, 0.68, 0.24)$ (b) $\frac{1}{2}(0, 1, 1)$, $\frac{1}{2}(1, 1, 0)$ (c) $(0.4, 0.3, 0.4)$, $(0.26, 0.08, 0.66)$ (d) $(0, 0.25, 0.25)$, $(0.225, 0, 0.15)$

As expected, the states sum to 1 for a-c, but not for d. Therefore a-c are Markov chains.

8

For each of the dynamical systems of the previous exercise, determine by calculation whether the system tends to a limiting steady-state vector. If so, what is it?

Solution:

Exercise 2.3.12

We can do this in Python by multiplying the initial state vector by each matrix a large number of times and seeing if the result converges.

```
from sympy import Matrix
A1 = Matrix([[0.1, 0.3, 0], [0, 0.4, 1], [0.9, 0.3, 0]])
A2 = Matrix([[0, 0, 1], [0, 1, 0], [1, 0, 0]])
A3 = Matrix([[0.5, 0.3, 0], [0, 0.4, 0], [0.5, 0.3, 1]])
A4 = Matrix([[0, 0, 0.9], [0.5, 0, 0], [0, 0.5, 0.1]])
x = Matrix([1/2, 1/2, 0])

def limit(A, x0, n):
    print(x0)
    for i in range(n):
        x0 = A*x0
        # This was initially confusing -- printing only every 10 iterations, I missed the i
        if i % 10 == 0 or i % 10 == 1:
            print(x0)

print("A1")
limit(A1, x, 100)
print("A2")
limit(A2, x, 100)
print("A3")
limit(A3, x, 100)
print("A4")
limit(A4, x, 100)
```

```
A1
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0.200000000000000], [0.200000000000000], [0.600000000000000]])
Matrix([[0.0800000000000000], [0.680000000000000], [0.240000000000000]])
Matrix([[0.172271278520000], [0.517519069520000], [0.310209651960000]])
Matrix([[0.172482848708000], [0.517217279768000], [0.310299871524000]])
Matrix([[0.172413911708915], [0.517241315268424], [0.310344773022661]])
Matrix([[0.172413785751419], [0.517241299130031], [0.310344915118550]])
Matrix([[0.172413793080544], [0.517241379195438], [0.310344827724019]])
Matrix([[0.172413793066686], [0.517241379402194], [0.310344827531121]])
Matrix([[0.172413793103395], [0.517241379310495], [0.310344827586110]])
Matrix([[0.172413793103488], [0.517241379310308], [0.310344827586204]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
Matrix([[0.172413793103448], [0.517241379310345], [0.310344827586207]])
A2
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
```

```
Matrix([[0], [0.500000000000000], [0.500000000000000]])
Matrix([[0.500000000000000], [0.500000000000000], [0]])
A3
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0.400000000000000], [0.200000000000000], [0.400000000000000]])
Matrix([[0.260000000000000], [0.0800000000000000], [0.660000000000000]])
Matrix([[0.000913647940000000], [2.09715200000000e-5], [0.999065380540000]])
Matrix([[0.000463115426000000], [8.38860800000000e-6], [0.999528495966000]])
Matrix([[9.47077246639594e-7], [2.19902325555200e-9], [0.999999050723730]])
Matrix([[4.74198330296463e-7], [8.79609302220801e-10], [0.999999524922060]])
Matrix([[9.30630821712715e-10], [2.30584300921370e-13], [0.999999999069139]])
Matrix([[4.65384586146634e-10], [9.22337203685479e-14], [0.999999999534523]])
Matrix([[9.09422166223752e-13], [2.41785163922926e-17], [0.999999999999090]])
Matrix([[4.54718336666794e-13], [9.67140655691706e-18], [0.999999999999545]])
Matrix([[8.88170813796524e-16], [2.53530120045647e-21], [0.999999999999999]])
Matrix([[4.44086167488622e-16], [1.01412048018259e-21], [0.999999999999999]])
Matrix([[8.67360940451606e-19], [2.65845599156984e-25], [1.00000000000000]])
Matrix([[4.33680549979483e-19], [1.06338239662794e-25], [1.00000000000000]])
Matrix([[8.47032863626506e-22], [2.78759314981634e-29], [1.00000000000000]])
Matrix([[4.23516440176032e-22], [1.11503725992654e-29], [1.00000000000000]])
Matrix([[8.27180603784018e-25], [2.92300327466182e-33], [1.00000000000000]])
Matrix([[4.13590302768910e-25], [1.16920130986473e-33], [1.00000000000000]])
Matrix([[8.07793566026819e-28], [3.06499108173179e-37], [1.00000000000000]])
Matrix([[4.03896783105359e-28], [1.22599643269272e-37], [1.00000000000000]])
A4
Matrix([[0.500000000000000], [0.500000000000000], [0]])
Matrix([[0], [0.250000000000000], [0.250000000000000]])
Matrix([[0.225000000000000], [0], [0.150000000000000]])
Matrix([[0.00302039010000000], [0.00120386925000000], [0.00264539827500000]])
Matrix([[0.00238085844750000], [0.00151019505000000], [0.000866474452500000]])
Matrix([[4.03791461577412e-5], [2.52845782290541e-5], [2.17203569987220e-5]])
Matrix([[1.95483212988498e-5], [2.01895730788706e-5], [1.48143248143993e-5]])
Matrix([[4.06315437745616e-7], [3.50928578393455e-7], [2.81839630258029e-7]])
Matrix([[2.53655667232226e-7], [2.03157718872808e-7], [2.03648252222531e-7]])
Matrix([[4.89866853485548e-9], [3.95031105574765e-9], [3.70325613478943e-9]])
Matrix([[3.33293052131049e-9], [2.44933426742774e-9], [2.34548114135277e-9]])
Matrix([[6.19440844858878e-11], [4.70794262956052e-11], [4.43926066707180e-11]])
Matrix([[3.99533460036462e-11], [3.09720422429439e-11], [2.79789738148744e-11]])
Matrix([[7.53264744484970e-13], [5.82090739342239e-13], [5.32977001565985e-13]])
Matrix([[4.79679301409387e-13], [3.76632372242485e-13], [3.44343069827718e-13]])
Matrix([[9.09975559486833e-15], [7.10204455266123e-15], [6.50648903042589e-15]])
Matrix([[5.85584012738330e-15], [4.54987779743417e-15], [4.20167117937320e-15]])
```

```
Matrix([[1.10752324227116e-16], [8.61303873249460e-17], [7.93329748585034e-17]])
Matrix([[7.13996773726531e-17], [5.53761621135579e-17], [5.09984911483233e-17]])
Matrix([[1.34906603357799e-18], [1.04742159016943e-18], [9.64385211129635e-19]])
Matrix([[8.67946690016671e-19], [6.74533016788995e-19], [6.20149316197677e-19]])
```

Answers:

(a) Tends to steady state $(0.172414, 0.517241, 0.310345)$.
(b) No steady state (alternates).
(c) Yes, steady state $(0, 0, 1)$.
(d) Yes, steady state $(0, 0, 0)$.

9

A population is modeled with two states, immature and mature, and the resulting structured population model transition matrix is $\begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & 0 \end{bmatrix}$.

(a) Explain what this matrix says about the two states.
(b) Starting with a population of $(30, 100)$, does the population stabilize, increase or decrease over time? If it stabilizes, to what distribution?

Solution:

```
from sympy import Matrix
A = Matrix([[1/2, 1], [1/2, 0]])
x = Matrix([30, 100])
for i in range(100):
    x = A*x
    if i % 10 == 0:
        print(x)
```

```
Matrix([[115.000000000000], [15.0000000000000]])
Matrix([[86.6943359375000], [43.3056640625000]])
Matrix([[86.6666936874390], [43.3333063125610]])
Matrix([[86.6666666930541], [43.3333333069459]])
Matrix([[86.6666666666924], [43.3333333333076]])
Matrix([[86.6666666666667], [43.3333333333333]])
Matrix([[86.6666666666667], [43.3333333333333]])
Matrix([[86.6666666666667], [43.3333333333333]])
```

```
Matrix([[86.6666666666667], [43.3333333333333]])
Matrix([[86.6666666666667], [43.3333333333333]])
```

Exercise 2.3.13

Solution. (a) The first column says that 50% of the immature become mature and 50% of the immature remain immature in one time period. The second column says that none of the mature survive, but each mature individual produces one immature in one time period. (b) The total populations after $0, 3, 6, 9, 18$ time periods is a constant 130 , and populations tend to approximately $(86.667, 43.333)$.

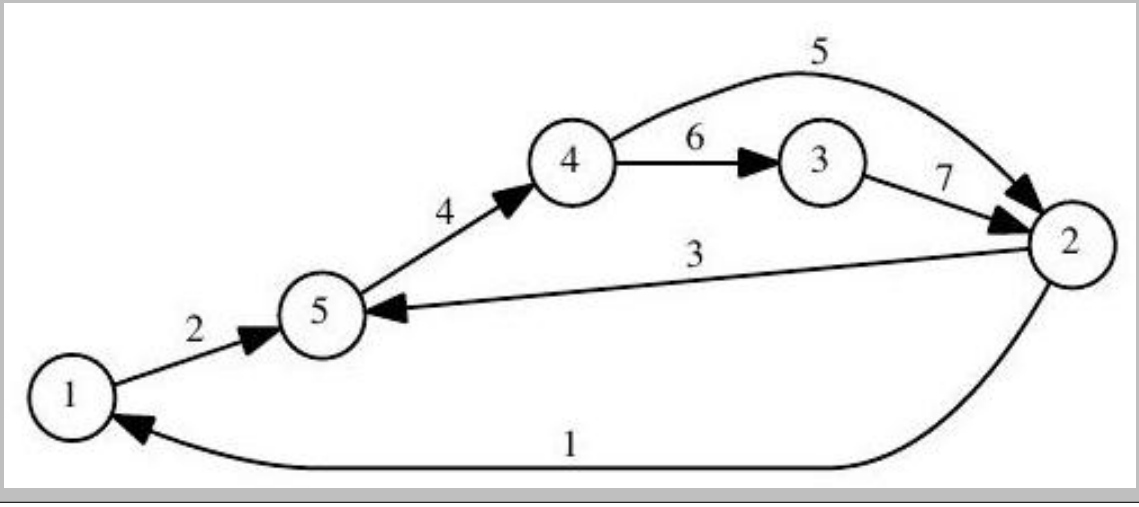
10

A digraph G has vertex set $V = \{1, 2, 3, 4, 5\}$ and edge set $E = \{(2, 1), (1, 5), (2, 5), (5, 4), (4, 2), (4, 3), (3, 2)\}$. Sketch a picture of the graph G and find its adjacency matrix. Use this to find the power of each vertex of the graph and determine whether this graph is dominance-directed.

Solution:

Exercise 2.3.15

Powers of vertices $1 - 5$ are $2, 4, 3, 5, 3$, respectively. Graph is dominance directed (there are no bi-directional edges between any pairs of vertices), adjacency matrix is $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$. Picture:



11

Consider the linear difference $y_{k+2} - y_{k+1} - y_k = 0$.

(a) Express this difference in matrix form.

(b) Find the first ten terms of the solution to this difference given the initial conditions $y_0 = 0, y_1 = 1$. (This is the well-known Fibonacci sequence.)

Solution:

Exercise 2.3.19

(a) $\begin{bmatrix} y_{k+1} \\ y_{k+2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y_k \\ y_{k+1} \end{bmatrix}$

(b) The first ten terms are $0, 1, 1, 2, 3, 5, 8, 13, 21, 34$.

12

Suppose that in Example 2.27 you invest $\$1,000$ initially (the zeroth year) and no further amounts. Make a table of the value of your investment for years 0 to 12. Also include a column that calculates the annual interest rate that your investment is earning each year, based on the current and previous year's values. What conclusions do you draw? You will need a technology tool for this exercise.

Solution:

Problem 2.3.24 Solution.

After a sizable third year earning of 24%, the annual rate appears to settle down to 14.41%. Partial table:

Year	Value	Interest Rate (%)
0	1000	-
1	1000	0
3	1240	24
6	1859	14.1
9	2785	14.41
12	4171	14.41