

CPSC 303: RECURRENCE RELATIONS AND FINITE PRECISION

JOEL FRIEDMAN

CONTENTS

1. Basic Facts About Finite Precision	2
2. Cancellations	2
3. Some Two- and Three-Term Recurrence Relations	3
3.1. Two-Term Recurrence Relations	3
3.2. A Simple Three-Term Recurrence Relation	4
3.3. Solution of (2)	7
3.4. Fibonacci Recurrence	7
3.5. Some Other Three-Term Recurrence Relations	8
3.6. Matrix Form of a Three-Term (Constant Coefficient) Recurrence	9
3.7. More General Recurrence Relations	9
4. MATLAB code for the Fibonacci Recurrence	10
4.1. The Matrix Form of the Fibonacci Recurrence	11
5. MATLAB Experiments on the Nature of Double Precision	12
5.1. Some Recursive Relations with Constant Solutions	12
5.2. The Case $r = 1/3$	12
5.3. The Case $r = 1/2$	13
5.4. MATLAB Experiment for $r = 1/2$	13
5.5. MATLAB Experiment for $r = 1/3, 1/5, 1/10$	13
5.6. Similar MATLAB Experiments in Matrix Form	14
6. The Fibonacci Lecture	15
7. Conceptual Examples From ODE's and PDE's	15
8. Exercises	16
8.1. Some MATLAB Experiments	16
8.2. Some Exercises	17
Appendix A. Linear Recurrence Relations with Constant Coefficients	18
Appendix B. Diagonalization	19
B.1. Non-Diagonalizable Matrices and Multiple Eigenvalues/Roots	21
Appendix C. The Shift Operator	21

Copyright: Copyright Joel Friedman 2020. Not to be copied, used, or revised without explicit written permission from the copyright owner.

Research supported in part by an NSERC grant.

Disclaimer: The material may sketchy and/or contain errors, which I will elaborate upon and/or correct in class. For those not in CPSC 303: use this material at your own risk. . .

Acknowledgement: We wish to thank Will Evans for discussions and ideas for course material.

The main goals of this article are to

- (1) make some remarks on errors in finite precision computations;
- (2) introduce some MATLAB statements;
- (3) explain some properties of three-term recursion relations (which are a simpler analog of second-order differential equations);
- (4) use (1)–(3) above to test the finite precision of your MATLAB (or other) software on your machine;
- (5) make some remarks on differential equations and how (4) relates to *stiff* differential equations.

Technicalities of finite precision are covered in much greater detail in Chapter 2 of the course textbook [A&G] (by Ascher and Greif); in CPSC 303 this year, the course begins by covering this article instead. Some finite precision issues will arise later in CPSC 303, but these issues tend to be specific to each application and less dependent on the technicalities of Chapter 2.

1. BASIC FACTS ABOUT FINITE PRECISION

Here are the basic facts about finite precision that will suffice for our current needs in CPSC 303.

- (1) *Double precision*, which is the MATLAB default, is a 64-bit representation of a floating point numbers (using base 2 scientific notation), which gives you roughly 16 decimal digits of precision (with numbers in the range of roughly $\pm 10^{\pm 300}$); *single precision*, a 32-bit representation, gives roughly 7 decimal digits.
- (2) On some modern devices, such as a calculator, you can't be sure how numbers are represented and how much precision you have in any calculation; this includes how floating point numbers are stored, but also operations such as $+$, $/$, dividing by zero (which should give some sort of error message), rounding, trig functions like sine and cosine, etc.
- (3) Most modern computers follow some form of *IEEE 754 Standard* for floating-point computations single and double precision (and some others in bases 2 and 10). The original version was published in 1985, and revised in 2008 and 2019.
- (4) There are a number of good reasons to have all computers follow some specific standard.

In class we might mention some real-world problems arising from floating point imprecision; we may also mention some early Intel Pentium chips that produced errors (although fairly rarely) after the 5th or 6th significant digit.

2. CANCELLATIONS

Of course, in any finite precision computation, errors can accumulate over a large computation (e.g., solving a differential equation over a large time interval).

However, the most dramatic errors occur in computations involving “cancellations,” such as

$$(2^{100} + 1) - 2^{100}$$

which in floating point arithmetic is probably 0 unless you are working with more than 100 bits of precision (or, similarly, $(10^2 + 1) - 10^2$ if you are working with 2 decimal digits of precision). No one willingly introduces such cancellations into their algorithms. In practice these cancellations can arise, although it tends to depend on the particular situation.

For the rest of this article, we will give some examples. We briefly describe a toy example.

Example 2.1. A (mostly) toy example arises in the quadratic equation $ax^2 + bx + c = 0$, or equivalently $c(1/x)^2 + b(1/x) + a = 0$, whose two solutions are given by

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

If ac is “much smaller” than b^2 , and $b > 0$, then the $+$ solution on the left-hand-side can have cancellation problems; in this case the corresponding right-hand-side solution (with the $-$) remedies the problem.

Most problems with finite precision are not as simple to identify and fix.

3. SOME TWO- AND THREE-TERM RECURRENCE RELATIONS

In this section we will discuss sequences of real numbers

$$\{x_n\}_{n \in \mathbb{Z}} = \{\dots, x_{-1}, x_0, x_1, x_2, \dots\}$$

that satisfy a (constant coefficient) three-term recurrence relation. We will uncover some numerical issues of working with such equations that arise in numerically solving differential equations (Chapter 16 of [A&G]).

3.1. Two-Term Recurrence Relations. Two-term recurrence relations are much simpler than three-term recurrences. Two-term recurrences illustrate some of the principles of recurrence relations, although they are deceptively.

Consider the recurrence relation

$$(1) \quad x_{n+1} = 3x_n \quad \forall n \in \mathbb{Z}$$

with the additional constraint that $x_0 = 10$. The equation above implies that

$$x_1 = 3x_0 = 30, \quad x_2 = 3x_1 = 90, \quad x_3 = 3x_2 = 270;$$

continuing in this fashion, we have

$$x_m = 3^m 10 \quad \forall m \geq 4.$$

The recurrence (1) can also be used to determine x_{-1}, x_{-2}, \dots by writing

$$x_{n+1} = 3x_n \quad \iff \quad x_{n+1} - 3x_n = 0 \quad \iff \quad x_n = x_{n+1}/3,$$

and hence

$$x_{-1} = x_0/3 = (1/3)10, \quad x_{-2} = x_{-1}/3 = (1/9)10, \quad x_{-3} = (1/27)10, \quad \dots$$

Hence we conclude that (1) and the “initial condition” $x_0 = 10$ implies that

$$x_m = 3^m 10 \quad \forall m \in \mathbb{Z}.$$

Conversely, we see that the sequence $x_m = 3^m 10$ satisfies (1) and the initial condition $x_0 = 10$.

By the same principle, we see that a sequence $\{x_n\}$ satisfies (1) iff $x_n = 3^n x_0$ for all n .

In the above, the condition $x_0 = 10$ is called an “initial condition” because we think x_n representing the value of some “system” at time n ; hence x_0 is the value of the system at time 0, x_1 the value at time 1, etc.

Notice that if we specify that $x_5 = 23$ in (1), then this condition also determines x_n uniquely as $x_n = 3^{n-5} 23$. Similarly specifying any particular value of x_n gives rise to a unique sequence $\{x_n\}$ satisfying (1).

More generally, if $r \in \mathbb{R}$ is any fixed real number, then the recurrence equation

$$x_{n+1} = rx_n \quad \forall n \in \mathbb{Z}$$

holds iff $x_n = r^n x_0$ for all $n \in \mathbb{Z}$, and specifying any initial condition determines x_n uniquely.

The recurrence equations in this section are related to the differential equation for a function $x = x(t)$ (thinking of t as representing time),

$$\frac{dx}{dt} = Cx,$$

whose unique solution is $x(t) = e^{Ct} x(0)$ (differential equations will be discussed in more detail when we get to Chapter 16 of [A&G]). Writing $r = e^C$, we have $x(t) = r^t x(0)$, which resembles the solution to our two-term recurrence relations. However, $x = x(t)$ is defined for all $t \in \mathbb{R}$, whereas in recurrence relations $x = x_n$ is defined only for $n \in \mathbb{Z}$.

3.2. A Simple Three-Term Recurrence Relation. Consider the three-term recurrence relation

$$(2) \quad x_{n+2} - 5x_{n+1} + 6x_n = 0 \quad \forall n \in \mathbb{Z}.$$

If we specify (the “initial conditions”) $x_0 = 4$ and $x_1 = 7$, we can write the recurrence as

$$x_{n+2} = 5x_{n+1} - 6x_n$$

to determine x_2, x_3, x_4, \dots

Let’s use MATLAB code to determine the first few values; unfortunately MATLAB insists on positive integer indices, so the code below will take $x_1 = 4$ and $x_2 = 7$ instead:

```
clear
x{1} = 4
x{2} = 7
for i=3:20,
    i
    x{i}=5*x{i-1}-6*x{i-2},
end
```

[You may have to cut and paste this code line-by-line if your cutting and pasting method ignores spaces and new lines.] The output should give you the sequence

(MATLAB will call this a “cell array”)

4, 7, 11, 13, -1, -83, -409, -1547, -5281, ..., -386109769, -1.1596e+09

Can you guess what is the formula for x_n ?

You can also determine x_{-1}, x_{-2}, \dots by writing the above recurrence as

$$x_n = \frac{-x_{n+2} + 5x_{n+1}}{6}.$$

```
% Note that the symbol % is used for comments in MATLAB
% Note that a semicolon (;) suppresses the output
%
clear                % this command clears all the variables
x{1} = 7;  x{2} = 4 ;

for i=3:20,
    x{i}=( -x{i-2} + 5 * x{i-1} )/6 ;
end

x                    % this line prints out x
```

which gives the sequence

1.9071e-05, ..., 0.5880, 1.1389, 2.1667, 4, 7

Combining the sequences gives the numerical results

..., $x_{-2} = 1.1389, x_{-1} = 2.1667, x_0 = 4, x_1 = 7, x_2 = 11, x_3 = 13, x_4 = -1, \dots$

Next we derive a formula for x_n .

Try to guess (or determine without guessing) a formula for x_n before turning the page.

Did you really try?

3.3. Solution of (2). There are many solutions to (2). First let us determine some “special solutions:” let us see if there are any nonzero $r \in \mathbb{R}$ such that $x_n = r^n$, i.e.,

$$\dots, x_{-2} = r^{-2}, x_{-1} = r^{-1}, x_0 = 1, x_1 = r, x_2 = r^2, \dots$$

is a solution to (2). This amounts to the equation

$$r^{n+2} - 5r^{n+1} + 6r^n = 0$$

for all n , which (since we are looking for $r \neq 0$) amounts to

$$r^2 - 5r + 6 = 0$$

and hence $r = 2, 3$. It follows that

- (1) $x_n = 2^n$ is a solution to (2);
- (2) $x_n = 3^n$ is a solution to (2);
- (3) as a result (by the “linearity” of (2)) for any $C_1, C_2 \in \mathbb{R}$,

$$(3) \quad x_n = C_1 2^n + C_2 3^n$$

satisfies (2).

[It turns out that this formula for x_n represents all possible solutions to (2), although we do not need this result here.]

Now let us find C_1, C_2 in (3) for which $x_0 = 4$ and $x_1 = 7$, which amounts to

$$x_0 = 4 = C_1 + C_2, \quad x_1 = 7 = C_1 2 + C_2 3,$$

which gives $C_2 = -1$ and $C_1 = 5$. It follows that

$$x_n = (5)2^n + (-1)3^n$$

satisfies both (2) and $x_0 = 4$ and $x_1 = 7$. This formula also shows that for $n \rightarrow +\infty$, x_n is roughly $(-1)3^n$, and for $n \rightarrow -\infty$, x_n is roughly $(5)2^n$; i.e., the $(-1)3^n$ term dominates for n tending to positive infinity, and the $(5)2^n$ dominates for n tending to negative infinity.

3.4. Fibonacci Recurrence. The Fibonacci numbers

$$\dots, F_0 = 0, F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, F_7 = 13, \dots$$

satisfy the recurrence equation

$$(4) \quad x_{n+2} = x_{n+1} + x_n \quad \forall n \in \mathbb{Z}$$

(in the case where $x_n = F_n$). The following are not hard to see:

- (1) If x_0, x_1 are given, then (4) determines x_2, x_3, \dots and also x_{-1}, x_{-2}, \dots
- (2) The Fibonacci numbers are determined by (4) and the (“initial”) conditions $x_0 = 0$ and $x_1 = 1$ (or, equivalently, $x_1 = 1$ and $x_2 = 1$).
- (3) It is not hard to show that a sequence $\{x_n\}$ satisfies (4) iff x_n is given by a formula

$$(5) \quad x_n = C_1 \left(\frac{1 + \sqrt{5}}{2} \right)^n + C_2 \left(\frac{1 - \sqrt{5}}{2} \right)^n$$

for constants C_1, C_2 .

- (4) The above numbers $(1 \pm \sqrt{5})/2$, called the *golden ratio* $(1 + \sqrt{5})/2 = 1.6180339\dots$ and *the conjugate of the golden ratio* $(1 - \sqrt{5})/2 = -0.6180339\dots$, arise from looking for a solution $x_n = r^n$ to (4) (as in the previous subsection), which leads to equation

$$r^2 - r - 1 = 0, \quad \text{hence } r = \frac{1 \pm \sqrt{5}}{2}.$$

- (5) The Fibonacci numbers are given by the formula

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n,$$

which is (5) in the special case $C_1 = 1/\sqrt{5}$, $C_2 = -1/\sqrt{5}$.

3.5. Some Other Three-Term Recurrence Relations. We will be interested in the following three-term (constant coefficient) recurrence relations: we will take two distinct real numbers, r_1, r_2 , and consider the relation

$$(6) \quad x_{n+2} = (r_1 + r_2)x_{n+1} - (r_1 r_2)x_n.$$

The Fibonacci recurrence relation (4) is the special case where

$$r_1, r_2 = \frac{1 \pm \sqrt{5}}{2}.$$

Like the Fibonacci recurrence relation, we can see that for any constants $r_1 \neq r_2$, the general solution to the above recurrence relation is of the form

$$(7) \quad x_n = C_1 r_1^n + C_2 r_2^n$$

for constants C_1, C_2 . In particular, for $C_1 = 1$ and $C_2 = 0$ we have that $x_n = r_1^n$ is one solution to the recurrence relation, and similarly $x_n = r_2^n$ is another.

Example 3.1. Let $r_1 = 1$ and $r_2 = 1/2$. We get the recurrence relation

$$(8) \quad x_{n+2} = (3/2)x_{n+1} - (1/2)x_n.$$

We see that

- (1) if $x_0 = 1$ and $x_1 = 1/2$, then we have $x_n = (1/2)^n$ for all $n \in \mathbb{Z}$;
- (2) similarly if $x_0 = 1$ and $x_1 = 1$, then we have $x_n = 1$ for all $n \in \mathbb{Z}$;
- (3) more generally, the general solution to (8) is

$$x_n = C_1 + C_2(1/2)^n.$$

Example 3.2. Let $r_1 = 1$ and $r_2 = 1/3$. We get the recurrence relation

$$x_{n+2} = (4/3)x_{n+1} - (1/3)x_n.$$

The general solution is

$$x_n = C_1 + C_2(1/3)^n.$$

In what follows we will also use examples like $r_1 = 1$ and $r_2 = 1/4, 1/5, 1/10$; these all follow a similar pattern as the above examples.

3.6. Matrix Form of a Three-Term (Constant Coefficient) Recurrence.

The Fibonacci recurrence $x_{n+2} = x_{n+1} + x_n$ can be written in “matrix form”

$$\begin{bmatrix} x_{n+2} \\ x_{n+1} \end{bmatrix} = A \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

From this vector/matrix formula, it follows that

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = A \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} = A^2 \begin{bmatrix} x_{n-1} \\ x_{n-2} \end{bmatrix} = \dots = A^n \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}.$$

In particular, the Fibonacci numbers satisfy

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = A^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = A^n \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

This formula can be used to compute the Fibonacci numbers (see Section ??).

Similarly, any of the three-term relations can be written in matrix form; for example $x_{n+2} = 5x_{n+1} - 6x_n$ can be written as

$$\begin{bmatrix} x_{n+2} \\ x_{n+1} \end{bmatrix} = A \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 5 & -6 \\ 1 & 0 \end{bmatrix}.$$

One can also solve recurrences by diagonalizing the above matrix A (see Appendix B).

3.7. More General Recurrence Relations. One generalization of the two- and three-term recurrence relations we have discussed so far is a relation

$$x_{n+k} + a_1 x_{n+k-1} + \dots + a_{k-1} x_{n+1} + a_k x_n = 0, \quad \forall n \in \mathbb{Z}$$

where a_1, \dots, a_k are fixed real or complex numbers. See Appendix A for some remarks these recurrences. Their general solution can be found by solving the equation

$$r^k + a_1 r^{k-1} + \dots + a_{k-1} r + a_k = 0.$$

The only new phenomenon that he haven't described is what happens when the above equation has repeated roots (e.g., $r^2 - 2r + 1 = 0$, corresponding to the recurrence $x_{n+2} = 2x_{n+1} - x_n$).

The above recurrences can be written in matrix form involving a $k \times k$ (square) matrix analogous to the 2×2 matrix form for three-term (constant coefficient) recurrences.

Another generalization of the three-term recurrence relations we have discussed is the relation

$$(9) \quad x_{n+2} = f(n)x_{n+1} + g(n)x_n$$

where f, g are functions $\mathbb{Z} \rightarrow \mathbb{R}$ (or $\mathbb{Z} \rightarrow \mathbb{C}$, where \mathbb{C} denotes the complex numbers). This equation is still *linear*, in that if y_n and z_n are two particular solutions to this equation, then so is $C_1 y_n + C_2 z_n$ for any constants C_1, C_2 . A typical second-order ODE (ordinary differential equation) is more closely analogous to this type of equation; however, a lot of properties of (and problems arising in) numerical methods can be understood by looking at the recurrences (or ODE's) where f, g are constants independent of n .

We call a linear recurrence relations (9) a *linear recurrence with varying coefficients*; we will return to them when we study ODE's in Chapter 16 of [A&G].

4. MATLAB CODE FOR THE FIBONACCI RECURRENCE

The following MATLAB code will print out the first few Fibonacci numbers:

```
clear
Fib{1}=1
Fib{2}=1

for i=3:20, i, Fib{i}=Fib{i-1}+Fib{i-2}, end
```

It is a bit nicer to suppress the printout of the (cell array) Fib until the last step.

```
clear
Fib{1}=1
Fib{2}=1

for i=3:20, Fib{i}=Fib{i-1}+Fib{i-2}; end
```

Fib

To test how large floating point numbers can be, just print out more Fibonacci numbers:

```
clear
Fib{1}=1
Fib{2}=1

for i=3:1500, Fib{i}=Fib{i-1}+Fib{i-2}; end
```

Fib

Here is MATLAB code for a sequence satisfying the Fibonacci recurrence but with different x_1, x_2 :

```
clear
x{1}=3
x{2}=7

for i=3:20, x{i}=x{i-1}+x{i-2}; end
```

x

By taking different values for $x\{1\}, x\{2\}$ we get different sequences.

The following demonstrates the affects of rounding. If $x_1 = 1$ and $x_2 = r$ where

$$r = \frac{1 \pm \sqrt{5}}{2} = 1.618033\dots, -0.618033\dots$$

then $x_3 = r^2$, $x_4 = r^3$, etc. Taking the negative value of r , we can generate the sequence x_n numerically as follows:

```
clear
x{1}=1
x{2}=(1-sqrt(5))/2

for i=3:20, x{i}=x{i-1}+x{i-2}; end

x
```

However, if we generate more values of x_n , numerically things go awry:

```
clear
x{1}=1
x{2}=(1-sqrt(5))/2

for i=3:120, x{i}=x{i-1}+x{i-2}; end

x
```

In the next section we study this phenomenon by using simpler 3-term recurrences.

4.1. The Matrix Form of the Fibonacci Recurrence. Another way to write (4) is in the matrix form

$$\begin{bmatrix} x_{n+2} \\ x_{n+1} \end{bmatrix} = A \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

For given x_0, x_1 , we have

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = A^n \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}.$$

We can also derive (5) by diagonalizing the matrix A .

MATLAB was designed with matrix operations in mind. Here is some code that uses the matrix A and its powers.

```
% Any statement beginning with a % is a comment
```

```
clear
A = [1, 1; 1, 0]
u = [ 1 ; 1 ]

for i = 1:15, i, u = A * u , end
```

Here is the problematic case of $x_0 = 1$ and $x_1 = (1 - \sqrt{5})/2$.

```
clear
A = [1, 1; 1, 0]
```

```

u = [ (1-sqrt(5))/2 ; 1 ]

%      for i = a:b:c iterates i from a to c by steps of size b

for i = 5:5:120,    i, u = A^5 * u , end

```

5. MATLAB EXPERIMENTS ON THE NATURE OF DOUBLE PRECISION

We will now give some MATLAB experiments that give some insight into double precision arithmetic; for example, our experiments should convince you that double precision is implemented in scientific notation in base 2 (rather than in base 10). These experiments also show the problems in finite precision when solving recursive relations; these relations will also arise in solving differential equations.

5.1. Some Recursive Relations with Constant Solutions. According to Subsection 3.5, for any nonzero real $r \in \mathbb{R}$, by taking $r_1 = 1$ and $r_2 = r$ in (6) and (7), we see that the recurrence equation

$$x_{n+2} = (1+r)x_{n+1} - r x_n$$

has a general solution

$$x_n = C_1 + C_2 r^n.$$

In particular, if we set $x_0 = 1$ and $x_1 = r$, and use the above recurrence to compute x_n , then the exact solution to the above recurrence is $x_n = r^n$ (which is the case $C_1 = 0$ and $C_2 = 1$ above). Hence the calculation

$x_0 = 1, x_1 = r, x_2 = (1+r)x_1 - r x_0, x_3 = (1+r)x_2 - r x_1, x_4 = (1+r)x_3 - r x_2, \dots$
yields (in exact arithmetic) the sequence

$$x_0 = 1, x_1 = r, x_2 = r^2, x_3 = r^3, \dots$$

Hence if $|r| < 1$ then we have $x_n \rightarrow 0$ as $n \rightarrow \infty$.

However, we (meaning you) will make MATLAB experiments on the homework to see that if $|r| < 1$, then the roundoff error can give you a very different result once x_n is less than 10^{-18} or so; however, when $r = 1/2$ or $r = 1/4$, the numerical result will agree until x_n reaches 10^{-324} or so, because only division by powers of 2 are involved in the recurrence, and things go wrong only when you reach x_n that are close to the smallest nonzero real in double precision.

[Some more details will be given on the homework and/or the exercises at the end of this article.]

5.2. The Case $r = 1/3$. The case $r = 1/3$ gives the recurrence relation $x_n = (4/3)x_{n-1} - (1/3)x_{n-2}$, whose general solution is

$$x_n = C_1 + C_2(1/3)^n$$

where $C_1, C_2 \in \mathbb{R}$ are fixed real numbers. In particular, $C_1 = 0$ and $C_2 = 1$, corresponds to $x_n = (1/3)^n$. however, we claim that if you set $x_0 = 1, x_1 = 1/3$, and

$$x_n = (4/3)x_{n-1} - (1/3)x_{n-2} \quad \text{for } n \geq 2,$$

then in exact arithmetic you get $x_2 = 1/9, x_3 = 1/27$, etc. However, we claim that in MATLAB there will be roundoff error, due to the division by 3, and as $n \rightarrow \infty$ you get x_n tends to -1.7385e-18. This means—roughly speaking—that the

numerical solution resembles the solution where C_1 is a tiny negative number (and whatever C_2 is, the C_1 term is much larger than $C_2(1/3)^n$). The actual truth is more complicated, because when you transition from a dominant $C_2(1/3)^n$ term to a dominant C_1 term, the transition is a bit more complicated.

5.3. The Case $r = 1/2$. By contrast, the case $r = 1/2$ gives the recurrence $x_n = (3/2)x_{n-1} - (1/2)x_{n-2}$, whose general solution is

$$x_n = C_1 + C_2(1/2)^n;$$

however, setting $x_0 = 1$ and $x_1 = 1/2$, and using the recurrence to compute x_i for $i = 2, 3, \dots$, the divisions by 2 are computed essentially exactly (because double precision works in base 2). However, at i around 1075, when you reach the smallest possible value of double precision, the x_i 's begin to cycle in a curious way. (What happens and why?)

[Some MATLAB code for this will be given in the exercises at the end of this article and/or the homework assignments.]

5.4. MATLAB Experiment for $r = 1/2$. This code has division by 2; but since double precision works in base 2, the result sequence $1, 1/2, 1/4, 1/8, \dots$ is computed exactly.

```
clear
x{1} = 1;
x{2} = 1/2;
for i=3:100, x{i} = (3/2)*x{i-1}-(1/2)*x{i-2}; end
x
```

If we iterate the recurrence enough times, we obtain x_n that are as small as roughly 4.94×10^{-324} , around the 1075-th term, but then something weird happens.

```
% continuing from above... (The % is used for commenting in MATLAB)
%
for i=3:1100, x{i} = (3/2)*x{i-1}-(1/2)*x{i-2}; end
x
```

[Exercise: explain this weird cycling.]

5.5. MATLAB Experiment for $r = 1/3, 1/5, 1/10$. This code has division by 3; since double precision works in base 2, the result sequence $1, 1/3, 1/9, 1/27, \dots$ is not computed exactly, and the resulting output looks more like

$$C_1 + (1/3)^n C_2$$

where C_1 is (close to) but not exactly zero

```
clear
x{1} = 1;
x{2} = 1/3;
for i=3:100, x{i} = (4/3)*x{i-1}-(1/3)*x{i-2}; end
x
```

In this case x_n seems to converge to roughly $-1.7385e-18$.

This code has division by 5 in its recurrence equation.

```
clear
x{1} = 1;
x{2} = 1/5;
```

```
for i=3:100, x{i} = (6/5)*x{i-1}-(1/5)*x{i-2}; end
x
```

In this case x_n seems to converge to roughly $-4.0416e-17$.
This code has division by 10 in its recurrence equation.

```
clear
x{1} = 1;
x{2} = 1/10;
for i=3:100, x{i} = (11/10)*x{i-1}-(1/10)*x{i-2}; end
x
```

In this case x_n seems to converge to roughly $9.8263e-18$.

5.6. Similar MATLAB Experiments in Matrix Form. We may write $x_{n+2} = (1+r)x_{n+1} - rx_n$ in “matrix form”

$$\begin{bmatrix} x_{n+2} \\ x_{n+1} \end{bmatrix} = A \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 1+r & -r \\ 1 & 0 \end{bmatrix}.$$

For given x_0, x_1 , we have

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = A^n \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}.$$

Hence the case $x_0 = 1, x_1 = 1$ corresponds to the matrix power

$$\begin{bmatrix} 1+r & -r \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} r \\ 1 \end{bmatrix}.$$

Here is some matrix code that is likely produces similar results as the previous MATLAB code.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% division by 2
A = [ 3/2, -1/2 ; 1, 0 ]
u = [ 1/2; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% division by 3
A = [ 4/3, -1/3 ; 1, 0 ]
u = [ 1/3; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% division by 4
A = [ 5/4, -1/4 ; 1, 0 ]
u = [ 1/4; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% division by 10
A = [ 11/10, -1/10 ; 1, 0 ]
u = [ 1/10; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

6. THE FIBONACCI LECTURE

One way to begin any course on (any kind of) algorithms is to give the venerable “Fibonacci lecture,” meaning that the entire lecture is devoted to given many algorithms related to computing the n -th Fibonacci lecture.

As an example, one can discuss the formula

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = A^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = A^n \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and mention how one might exploit repeated squaring and Strassen’s method to speed things up.

When I was a student, Prof. John Reif gave such a lecture (in 1983 or so); I believe the tradition goes back some years (or decades) before this (in the prehistory of the internet...), and I do not know its history. I would be curious to know.

7. CONCEPTUAL EXAMPLES FROM ODE’S AND PDE’S

Some real-world examples of cancellation are easy to understand when you the properties of the solution of the equations; often you don’t even have to understand much about the particular ODE or PDE.

- (1) If two bodies collide in the “ n -body problem” under Newton’s law, then their differential equation(s) become “singular” at the collision (i.e., their distance becomes zero, the potential energy of the system becomes infinite, and therefore so does their kinetic energy, some of their speeds, etc.). In this case the equations themselves break down.
- (2) Similarly, if two bodies nearly collide, then the corresponding equations are close to a “singularity” and numerical methods require care (for various reasons, including cancellation problems).
- (3) If the solution to an equation (e.g., the classical heat or wave equation) depends significantly on data that your discrete approximation does not use, then this is clearly problematic; what may not be clear at this point is that this can happen if you are not careful... Often such discretizations scheme produces “garbage;” at times this can be viewed as a cancellation problem in the discrete approximation, although there is really something more fundamental that is wrong.
- (4) If you are solving a differential equation whose general solution is of the form $f(t) = C_1 e^t + C_2 e^{-t}$ (here C_1, C_2 are constants and t is a variable), then when exact initial conditions dictate that $C_1 = 0$ and $C_2 = 1$ you may have problems with cancellation when t is large. (For small t there is little difference between $C_1 = 0$ and $C_1 = 10^{-20}$; for large t there is...) This is analogous to the examples involving three-term recursive relations.

Other times discrete approximations to real-world problems have cancellation problems—such as instabilities in high order Runge-Kutta schemes—for reasons that are not apparent from the solutions of the equation itself (but that can be understood by analyzing the discrete approximation). We will see some examples in Chapters 16–18 of the course textbook.

[We also remark that (Newton’s law for celestial mechanics in) the 2-body problem violates relativity when two bodies get “very close,” since then their potential

energy can become arbitrarily negative and therefore the kinetic energy arbitrarily large. One such situation—the *precession of the perihelion of Mercury*—was remarkably (corrected and) modeled by Einstein’s general relativity.]

[Similarly the classical heat equation also violates relativity; however, the influence of one point in space over another of distance Δx is roughly proportional to $e^{-C|\Delta x|^2/(\Delta t)}$ in time Δt , for a moderate constant C . Hence relativity is violated, but by an immeasurably small amount in typical applications.]

8. EXERCISES

[THIS IS A WORK IN PROGRESS. I WILL LIKELY EDIT THE MATERIAL HERE AND ADD MORE MATERIAL.]

In CPSC 303 this term, we might assign some of the following problems as homework. The upshot is that you can get strong evidence that MATLAB finite precision is working in base 2 by running the code below.

8.1. Some MATLAB Experiments. SOME OF THIS MATLAB CODE AND EXPERIMENTS CAN BE FOUND IN EARLIER SECTIONS.

This code has division by 2; but since double precision works in base 2, the result sequence $1, 1/2, 1/4, 1/8, \dots$ is computed exactly.

```
clear
x{1} = 1;
x{2} = 1/2;
for i=3:100, x{i} = (3/2)*x{i-1}-(1/2)*x{i-2}; end
x
```

If we iterate the recurrence enough times, we obtain x_n that are as small as roughly 4.94×10^{-324} , around the 1075-th term, but then something weird happens.

```
% continuing from above... (The % is used for commenting in MATLAB)
%
for i=3:1100, x{i} = (3/2)*x{i-1}-(1/2)*x{i-2}; end
x
```

[Exercise: explain this weird cycling.]

This code has division by 3; since double precision works in base 2, the result sequence $1, 1/3, 1/9, 1/27, \dots$ is not computed exactly, and the resulting output looks more like

$$C_1 + (1/3)^n C_2$$

where C_1 is (close to) but not exactly zero

```
clear
x{1} = 1;
x{2} = 1/3;
for i=3:100, x{i} = (4/3)*x{i-1}-(1/3)*x{i-2}; end
x
```

In this case x_n seems to converge to roughly $-1.7385e-18$.

This code has division by 5 in its recurrence equation.

```
clear
x{1} = 1;
x{2} = 1/5;
for i=3:100, x{i} = (6/5)*x{i-1}-(1/5)*x{i-2}; end
```


x

In this case x_n seems to converge to roughly $-4.0416e-17$.

This code has division by 10 in its recurrence equation.

```
clear
x{1} = 1;
x{2} = 1/10;
for i=3:100, x{i} = (11/10)*x{i-1}-(1/10)*x{i-2}; end
x
```

In this case x_n seems to converge to roughly $9.8263e-18$.

Here are some matrix computations that do similar things.

```
%%%%%%%%% division by 2
A = [ 3/2, -1/2 ; 1, 0 ]
u = [ 1/2; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%% division by 3
A = [ 4/3, -1/3 ; 1, 0 ]
u = [ 1/3; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%% division by 4
A = [ 5/4, -1/4 ; 1, 0 ]
u = [ 1/4; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

```
%%%%%%%%% division by 10
A = [ 11/10, -1/10 ; 1, 0 ]
u = [ 1/10; 1 ]
for i = 10:10:150, i, u = A^10 * u , end
```

8.2. Some Exercises. Possible exercises [THIS IS A WORK IN PROGRESS]

- (1) (a) Show that if $r = 1$ or $r = 1/2$, then the sequence $x_n = r^n$ (where n ranges over \mathbb{Z} , the integers) satisfies the recurrence equation

$$(10) \quad x_n = (3/2)x_{n-1} - (1/2)x_{n-2}.$$

- (b) Show that if x_n satisfies (10), then for any $C \in \mathbb{R}$, the sequence $x'_n = Cx_n$ also satisfies (10) (i.e., $x'_n = (3/2)x'_{n-1} - (1/2)x'_{n-2}$).
- (c) Show that if x'_n and x''_n are two sequences satisfies (10), then also $x'''_n = x'_n + x''_n$ satisfies (10).
- (d) Conclude that for any constants $C_1, C_2 \in \mathbb{R}$, the sequence

$$x_n = C_1 + C_2(1/2)^n$$

satisfies (10).

- (e) Etc.

APPENDIX A. LINEAR RECURRENCE RELATIONS WITH CONSTANT
COEFFICIENTS

In this section we gather some facts about linear recurrence relations with constant coefficients. These are analogous to an ODE with constant coefficients.

A *linear recurrence relation* is a relation

$$(11) \quad x_{n+k} + a_1 x_{n+k-1} + \cdots + a_{k-1} x_{n+1} + a_k x_n = 0, \quad \forall n \in \mathbb{Z}$$

(i.e., for all integers n), where a_1, \dots, a_k are real or complex; for simplicity we assume that $a_k \neq 0$. a sequence

$$\{x_m\}_{m \in \mathbb{Z}} = \{\dots, x_{-1}, x_0, x_1, x_2, \dots\}$$

satisfies the above relation if this equation holds for all $n \in \mathbb{Z}$.

Example A.1. The Fibonacci numbers satisfy the recurrence relation

$$x_{n+2} - x_{n+1} - x_n = 0.$$

Notice that the solution to the equation $z^2 - z - 1 = 0$ has two roots, r_1, r_2 , given by

$$\frac{1 \pm \sqrt{5}}{2},$$

and that if r is one of those roots, then the sequence $x_n = r^n$ satisfies the above recurrence. It is known that a sequence $\{x_n\}$ is a solution of the above recurrence iff it is of the form

$$x_n = C_1 r_1^n + C_2 r_2^n$$

for some fixed constants C_1, C_2 .

There is a standard way to describe all solutions of (11): namely, set

$$p(r) = r^k + a_1 r^{k-1} + \cdots + a_k,$$

and let the (complex) roots of p be r_1, \dots, r_s , so that

$$p(r) = (r - r_1)^{m_1} \cdots (r - r_s)^{m_s}$$

for some positive integers m_1, \dots, m_s (we are using the fundamental theorem of algebra). Then x_n satisfies (11) iff

$$x_n = q_1(n)r_1^n + \cdots + q_s(n)r_s^n$$

where each q_j is a polynomial of degree at most $m_j - 1$.

Example A.2. Let r_1, r_2 be distinct and nonzero (either real or complex numbers). Then since

$$(z - r_1)(z - r_2) = z^2 - (r_1 + r_2)z + (r_1 r_2),$$

we have that solutions to the recurrence equation

$$x_{n+2} - (r_1 + r_2)x_{n+1} + (r_1 r_2)x_n = 0,$$

or, equivalently

$$x_{n+2} = (r_1 + r_2)x_{n+1} - (r_1 r_2)x_n$$

are given by all sequences $\{x_n\}$ of the form

$$x_n = C_1 r_1^n + C_2 r_2^n$$

for constants C_1, C_2 .

Example A.3. Taking $r_1 = 1$ and $r_2 = 1/3$ in the above example shows that the recurrence

$$x_{n+2} = (4/3)x_{n+1} - (1/3)x_n$$

(for all $n \in \mathbb{Z}$) is satisfied iff for some constants C_1, C_2 we have

$$x_n = C_1 1^n + C_2 (1/3)^n = C_1 + C_2 (1/3)^n.$$

Example A.4. Consider the recurrence relation

$$x_{n+2} = 2x_{n+1} - x_n, \quad \text{i.e., } x_{n+2} - 2x_{n+1} + x_n = 0.$$

The corresponding polynomial is $p(z) = z^2 - 2z + 1 = (z - 1)^2$, whose roots are 1, 1 (i.e., 1 with multiplicity 2). The general solution of this equation is

$$x_n = C_1 + C_2 n$$

where C_1, C_2 are constants, or equivalently $x_n = q(n)$ where q is a polynomial of degree at most 1. Similarly the general solution to the recurrence relation

$$x_{n+3} - 3x_{n+2} + 3x_{n+1} - x_n = 0$$

is

$$x_n = C_1 + C_2 n + C_3 n^2,$$

and the general solution to the recurrence relation

$$x_{n+4} - 4x_{n+3} + 6x_{n+2} - 4x_{n+1} + x_n = 0$$

is

$$x_n = C_1 + C_2 n + C_3 n^2 + C_4 n^3.$$

These recurrences can be useful in many topics covered in CPSC 303.

[It is sometime useful to interpret the above recurrence using the “shift operator,” meaning that we view the above relation as saying that $p(\sigma)x_{n-k} = 0$ for all n , where σ is the (*upward*) *shift operator* given by $\sigma x_m = x_{m+1}$; this is equivalent to requiring that $p(\sigma)x_n = 0$ for all $n \in \mathbb{Z}$.]

APPENDIX B. DIAGONALIZATION

One can solve the three-term (constant coefficient) recurrences in this article by writing them in matrix form and diagonalizing the matrix. In this appendix we review the ideas.

It is easy to take the power of a diagonal matrix: for example if

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

then we easily see that for any $n = 1, 2, 3, \dots$

$$D^n = \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix}.$$

It follows that if A is a matrix that can be written as MDM^{-1} for some invertible matrix, M , then

$$A^2 = (MDM^{-1})(MDM^{-1}) = MDM^{-1}MDM^{-1} = MD^2M^{-1} = M \begin{bmatrix} 2^2 & 0 \\ 0 & 3^2 \end{bmatrix} M^{-1}$$

and similarly

$$A^n = M \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix} M^{-1}.$$

Now consider the first three-term recurrence we considered:

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = A^n \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 5 & -6 \\ 1 & 0 \end{bmatrix}.$$

Computing the eigenvectors and eigenvalues of A see that A has eigenvalues 2, 3, with eigenvectors (which are unique up to scalar multiple) given by

$$A \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad A \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3 \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

(It is no coincidence that the eigenvector corresponding to $r = 2, 3$ is $[r \ 1]^T$.) [We may review some of this linear algebra, or we may use it as a black box and/or assign some homework on this.] Here we have

$$A = M \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} M^{-1}, \quad \text{where } M = \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix}.$$

And therefore

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = A^n \begin{bmatrix} x_1 \\ x_0 \end{bmatrix} = M \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix} M^{-1} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}$$

from which we can derive an expression for x_n in terms of x_1, x_0 . In more detail

$$\begin{aligned} M \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix} M^{-1} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix} &= \left(\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2^n & 0 \\ 0 & 3^n \end{bmatrix} \right) \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \cdot 2^n & 3 \cdot 3^n \\ 2^n & 3^n \end{bmatrix} \frac{1}{2 \cdot 1 - 3 \cdot 1} \begin{bmatrix} 1 & -3 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix} = \begin{bmatrix} 2 \cdot 2^n & 3 \cdot 3^n \\ 2^n & 3^n \end{bmatrix} (-1) \begin{bmatrix} x_1 - 3x_0 \\ -x_1 + 2x_0 \end{bmatrix} \\ &= \begin{bmatrix} \text{whatever} \\ -2^n(x_1 - 3x_0) - 3^n(-x_1 + 2x_0) \end{bmatrix} = \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} \end{aligned}$$

and so

$$x_n = -2^n(x_1 - 3x_0) - 3^n(-x_1 + 2x_0).$$

Note that the term

$$\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}$$

appearing above is equivalent to solving the equation

$$\begin{bmatrix} x_1 \\ x_0 \end{bmatrix} = C_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + C_2 \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Note also that the term that we called “whatever” is just the expression for x_n with n replaced by $n + 1$, in view of the fact that

$$\begin{bmatrix} 2 \cdot 2^n & 3 \cdot 3^n \\ 2^n & 3^n \end{bmatrix} = \begin{bmatrix} 2^{n+1} & 3^{n+1} \\ 2^n & 3^n \end{bmatrix}.$$

B.1. Non-Diagonalizable Matrices and Multiple Eigenvalues/Roots. We remark that not all matrices are diagonalizable, even those occurring in recurrence relations. For example, the recurrence

$$x_{n+2} - 2x_{n+1} + x_n = 0$$

whose general solution is

$$x_n = C_1 + C_2n$$

can be written as

$$\begin{bmatrix} x_{n+2} \\ x_{n+1} \end{bmatrix} = A \begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}.$$

The matrix A has eigenvalues 1, 1, i.e., the only eigenvalue is 1 but it has multiplicity two. This is also the case for the root $r = 1$ corresponding polynomial equation

$$r^2 - 2r + 1 = 0.$$

However, any matrix, including the above matrix A , can be written as MJM^{-1} where J is a *block Jordan matrix* (which is *Jordan canonical form* of A), roughly meaning that it is close to a diagonal matrix. In this case

$$A = MJM^{-1}$$

for an invertible matrix M and the matrix

$$J = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

In this case J^n still has simple formula:

$$J = I + N, \quad \text{where } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Since I is the identity and $N^2 = 0$ (so N is *nilpotent*), we have

$$J^n = (I + N)^n = I^n + nN + \binom{n}{2}N^2 + \cdots = I^n + nN = \begin{bmatrix} 1 & n \\ 0 & 1 \end{bmatrix}.$$

APPENDIX C. THE SHIFT OPERATOR

In this section we make some remarks on the *shift operator*. This operator is useful in a number of areas of study, including interpolation and differential equations (which we will study in CPSC 303); the shift operator part of the foundations of some areas of study, such as symbolic dynamics and time series.

If $\{x_n\}_{n \in \mathbb{Z}}$ is a two-sided infinite sequence of real numbers (or complex numbers, or rational numbers, etc.) indexed on n varying over the integers, \mathbb{Z} (or over the positive integers, or over the non-negative integers, etc.), the *shift operator*, σ , is naively defined by the formula

$$\sigma x_n = x_{n+1};$$

it is worth thinking about what this means precisely.

Usually one thinks of $\mathbf{x} = \{x_n\}_{n \in \mathbb{Z}}$ as a sequence, and $\sigma \mathbf{x}$ as the new sequence whose n -th term is x_{n+1} . Hence σ is a map from sequences to sequences; one easily verifies that it is a *linear operator*, meaning that if \mathbf{x} and \mathbf{y} are two sequences and C_1, C_2 are constants, then

$$\sigma(C_1\mathbf{x} + C_2\mathbf{y}) = C_1\sigma\mathbf{x} + C_2\sigma\mathbf{y}.$$

We often write $x = \{x_n\}_{n \in \mathbb{Z}}$ instead of $\mathbf{x} = \{x_n\}_{n \in \mathbb{Z}}$ when confusion is unlikely to occur. We often write $(\sigma x)_n$ instead of σx_n when we want to emphasize that σ is an operator, although σx_n is less cumbersome to write.

The operator σ has a large number of important properties. Here are some.

- (1) $\sigma^2 x_n = x_{n+2}$, and for any integer $k \geq 3$, $\sigma^k x_n = x_{n+k}$;
- (2) σ has an inverse (on sequences indexed over \mathbb{Z}), $\sigma^{-1} x_n = x_{n-1}$;
- (3) for any polynomial, $p(z)$, one can define the operator $p(\sigma)$;
- (4) we can write the recurrence

$$x_{n+2} - 5x_{n+1} + 6x_n = 0$$

as

$$(\sigma^2 - 5\sigma + 6)x_n = 0,$$

where the 6 really refers to the identity operator (which equals σ^0);

- (5) etc. etc. etc.

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, BC V6T 1Z4, CANADA.

E-mail address: `jf@cs.ubc.ca`

URL: `http://www.cs.ubc.ca/~jf`