Thinking about Problems

Ryan Greenup & James Guerra

August 27, 2020

Contents

§ 1 Fibonacci Sequence

This pops up all the time in natural sequences and fractals so we'll deal with it now

¶ 1.1 Computational Approach

The *Fibonacci* Numbers are given by:

$$F_n = F_{n-1} + F_{n-2} \tag{1}$$

This type of recursive relation can be expressed in *Python* by using recursion, as shown in listing 1, however using this function will reveal that it is extraordinarily slow, as shown in listing 2, this is because the results of the function are not cached and every time the function is called every value is recalculated[1], meaning that the workload scales in exponential as opposed to polynomial time.

The `functools` library for python includes the `@functools.lru_cache` decorator which will modify a defined function to cache results in memory [9], this means that the recursive function will only need to calculate each result once and it will hence scale in polynomial time, this is implemented in listing 3.

[1] Dr. Hazrat mentions something similar in his book with respect to *Mathematica*® [12, Ch. 13]

```
def rec_fib(k):
    if type(k) is not int:
        print("Error: Require integer values")
        return 0
    elif k == 0:
        return 0
    elif k <= 2:
        return 1
    return rec_fib(k-1) + rec_fib(k-2)
```

Listing 1: Defining the *Fibonacci Sequence* (1) using Recursion

```
  start = time.time()
  rec_fib(35)
  print(str(round(time.time() - start, 3)) + "seconds")

## 2.245seconds
```

Listing 2: Using the function from listing 1 is quite slow.

```
  from functools import lru_cache

  @lru_cache(maxsize=9999)
  def rec_fib(k):
      if type(k) is not int:
          print("Error: Require Integer Values")
          return 0
      elif k == 0:
          return 0
      elif k <= 2:
          return 1
      return rec_fib(k-1) + rec_fib(k-2)


start = time.time()
rec_fib(35)
print(str(round(time.time() - start, 3)) + "seconds")
## 0.0seconds
```

Listing 3: Caching the results of the function previously defined 2

```
  start = time.time()
  rec_fib(6000)
  print(str(round(time.time() - start, 9)) + "seconds")

## 8.3923e-05seconds
```

Restructuring the problem to use iteration will allow for even greater performance as demonstrated by finding F_{10^6} in listing 4. Using a compiled language such as *Julia* however would be thousands of times faster still, as demonstrated in listing 5.

```
  def my_it_fib(k):
      if k == 0:
          return k
      elif type(k) is not int:
          print("ERROR: Integer Required")
          return 0
      # Hence k must be a positive integer

      i  = 1
      n1 = 1
      n2 = 1

      # if k <=2:
      #     return 1

      while i < k:
         no = n1
         n1 = n2
         n2 = no + n2
         i = i + 1
      return (n1)

  start = time.time()
  my_it_fib(10**6)
  print(str(round(time.time() - start, 9)) + "seconds")

 ## 6.975890398seconds
```

Listing 4: Using Iteration to Solve the Fibonacci Sequence

In this case however an analytic solution can be found by relating discrete mathematical problems to continuous ones as discussed below at section .

```julia
function my_it_fib(k)
    if k == 0
        return k
    elseif typeof(k) != Int
        print("ERROR: Integer Required")
        return 0
    end
    # Hence k must be a positive integer

    i  = 1
    n1 = 1
    n2 = 1

    # if k <=2:
    #     return 1
    while i < k
       no = n1
       n1 = n2
       n2 = no + n2
       i = i + 1
    end
    return (n1)
end

@time my_it_fib(10^6)

##  my_it_fib (generic function with 1 method)
##    0.000450 seconds
```

Listing 5: Using Julia with an iterative approach to solve the 1 millionth fibonacci number

¶ 1.2 Exponential Generating Functions

1.2.1 Motivation

Consider the *Fibonacci Sequence* from (1):

$$\begin{aligned} a_n &= a_{n-1} + a_{n-2} \\ \iff a_{n+2} &= a_{n+1} + a_n \end{aligned} \tag{2}$$

from observation, this appears similar in structure to the following *ordinary differential equation*, which would be fairly easy to deal with:

$$f''(x) - f'(x) - f(x) = 0$$

This would imply that $f(x) \propto e^{mx}, \quad \exists m \in \mathbb{Z}$ because $\frac{\mathrm{d}(e^x)}{\mathrm{d}x} = e^x$, and so by using a power series it's quite feasable to move between discrete and continuous problems:

$$f(x) = e^{rx} = \sum_{n=0}^{\infty} \left[r \frac{x^n}{n!} \right]$$

1.2.2 Example

Consider using the following generating function, (the derivative of the generating function as in (4) and (5) is provided in section 1.2.3)

$$f(x) = \sum_{n=0}^{\infty} \left[a_n \cdot \frac{x^n}{n!} \right] \quad = e^x \tag{3}$$

$$f'(x) = \sum_{n=0}^{\infty} \left[a_{n+1} \cdot \frac{x^n}{n!} \right] = e^x \tag{4}$$

$$f''(x) = \sum_{n=0}^{\infty} \left[a_{n+2} \cdot \frac{x^n}{n!} \right] = e^x \tag{5}$$

So the recursive relation from (2) could be expressed :

$$\begin{aligned} a_{n+2} &= a_{n+1} + a_n \\ \frac{x^n}{n!} a_{n+2} &= \frac{x^n}{n!} (a_{n+1} + a_n) \\ \sum_{n=0}^{\infty} \left[\frac{x^n}{n!} a_{n+2} \right] &= \sum_{n=0}^{\infty} \left[\frac{x^n}{n!} a_{n+1} \right] + \sum_{n=0}^{\infty} \left[\frac{x^n}{n!} a_n \right] \\ f''(x) &= f'(x) + f(x) \end{aligned}$$

Using the theory of higher order linear differential equations with constant coefficients it can be shown:

$$f(x) = c_1 \cdot \exp\left[\left(\frac{1 - \sqrt{5}}{2} \right) x \right] + c_2 \cdot \exp\left[\left(\frac{1 + \sqrt{5}}{2} \right) \right]$$

By equating this to the power series:

$$f\left(x\right)=\sum_{n=0}^{\infty}\left[\left(c_1\left(\frac{1-\sqrt{5}}{2}\right)^n+c_2\cdot\left(\frac{1+\sqrt{5}}{2}\right)^n\right)\cdot\frac{x^n}{n}\right]$$

Now given that:

$$f\left(x\right)=\sum_{n=0}^{\infty}\left[a_n\frac{x^n}{n!}\right]$$

We can conclude that:

$$a_n=c_1\cdot\left(\frac{1-\sqrt{5}}{2}\right)^n+c_2\cdot\left(\frac{1+\sqrt{5}}{2}\right)$$

By applying the initial conditions:

$$\begin{aligned} a_0=c_1+c_2 &\implies c_1=-c_2 \\ a_1=c_1\left(\frac{1+\sqrt{5}}{2}\right)-c_1\frac{1-\sqrt{5}}{2} &\implies c_1=\frac{1}{\sqrt{5}} \end{aligned}$$

And so finally we have the solution to the *Fibonacci Sequence* 2:

$$\begin{aligned} a_n &= \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right] \\ &= \frac{\varphi^n-\psi^n}{\sqrt{5}} \\ &= \frac{\varphi^n-\psi^n}{\varphi-\psi} \end{aligned} \tag{6}$$

where:

- $\varphi=\frac{1+\sqrt{5}}{2}\approx 1.61\ldots$
- $\psi=1-\varphi=\frac{1-\sqrt{5}}{2}\approx 0.61\ldots$

1.2.3 Derivative of the Exponential Generating Function

Differentiating the exponential generating function has the effect of shifting the sequence to the backward: [16]

$$\begin{aligned}
f\left(x\right) &= \sum_{n=0}^{\infty}\left[a_n\frac{x^n}{n!}\right] \qquad (7)\\
f'\left(x\right)) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\sum_{n=0}^{\infty}\left[a_n\frac{x^n}{n!}\right]\right)\\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left(a_0\frac{x^0}{0!}+a_1\frac{x^1}{1!}+a_2\frac{x^2}{2!}+a_3\frac{x^3}{3!}+\dots\frac{x^k}{k!}\right)\\
&= \sum_{n=0}^{\infty}\left[\frac{\mathrm{d}}{\mathrm{d}x}\left(a_n\frac{x^n}{n!}\right)\right]\\
&= \sum_{n=0}^{\infty}\left[\frac{a_n}{(n-1)!}x^{n-1}\right]\\
\implies f'(x) &= \sum_{n=0}^{\infty}\left[\frac{x^n}{n!}a_{n+1}\right] \qquad (8)
\end{aligned}$$

If $f\left(x\right)=\sum_{n=0}^{\infty}\left[a_n\frac{x^n}{n!}\right]$ can it be shown by induction that $\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left(f\left(x\right)\right)=f^k\left(x\right)\sum_{n=0}^{\infty}\left[x^n\frac{a_{n+k}}{n!}\right]$

1.2.4 Homogeneous Proof

An equation of the form:

$$\sum_{n=0}^{\infty}\left[c_i\cdot f^{(n)}(x)\right]=0 \qquad (9)$$

is said to be a homogenous linear ODE: [26, Ch. 2]

Linear because the equation is linear with respect to $f(x)$

Ordinary because there are no partial derivatives (e.g. $\frac{\partial}{\partial x}(f(x))$)

Differential because the derivates of the function are concerned

Homogenous because the ***RHS*** is 0

- A non-homogeous equation would have a non-zero RHS

There will be k solutions to a k^{th} order linear ODE, each may be summed to produce a superposition which will also be a solution to the equation, [26, Ch. 4] this will be considered as the desired complete solution (and this will be shown to be the only solution for the recurrence relation (10)). These k solutions will be in one of two forms:

1. $f(x)=c_i\cdot e^{m_ix}$
2. $f(x)=c_i\cdot x^j\cdot e^{m_ix}$

where:

- $\sum_{i=0}^{k}\left[c_im^{k-i}\right]=0$
  - This is referred to the characteristic equation of the recurrence relation or ODE [17]
- $\exists i,j\in\mathbb{Z}^+\cap[0,k]$
  - These is often referred to as repeated roots [17, 27] with a multiplicity corresponding to the number of repetitions of that root [20, §3.2]

Unique Roots of Characteristic Equation

Example An example of a recurrence relation with all unique roots is the fibonacci sequence, as described in section 1.2.2 .

Proof Consider the linear recurrence relation (10):

$$\sum_{i=0}^{n} [c_i \cdot a_i] = 0, \quad \exists c \in \mathbb{R}, \ \forall i < k \in \mathbb{Z}^+$$

By implementing the exponential generating function as shown in (3), this provides:

$$\sum_{i=0}^{k} [c_i \cdot a_n] = 0$$

By Multiplying through and summing:

$$\begin{aligned} \implies \sum_{i=0}^{k} \left[\sum_{n=0}^{\infty} \left[c_i a_n \frac{x^n}{n!} \right] \right] &= 0 \\ \sum_{i=0}^{k} \left[c_i \sum_{n=0}^{\infty} \left[a_n \frac{x^n}{n!} \right] \right] &= 0 \end{aligned} \tag{10}$$

Recall from (3) the generating function $f(x)$:

$$\sum_{i=0}^{k} \left[c_i f^{(k)}(x) \right] = 0 \tag{11}$$

Now assume that the solution exists and all roots of the characteristic polynomial are unique (i.e. the solution is of the form $f(x) \propto e^{m_i x} : \quad m_i \neq m_j \forall i \neq j$), this implies that [26, Ch. 4] :

$$f(x) = \sum_{i=0}^{k} [k_i e^{m_i x}], \quad \exists m, k \in \mathbb{C}$$

This can be re-expressed in terms of the exponential power series, in order to relate the solution of the function $f(x)$ back to a solution of the sequence a_n, (see section for a derivation of the exponential power series):

$$
\begin{aligned}
\sum_{i=0}^{k}\left[k_i e^{m_i x}\right] &= \sum_{i=0}^{k}\left[k_i \sum_{n=0}^{\infty}\frac{(m_i x)^n}{n!}\right] \\
&= \sum_{i=0}^{k}\sum_{n=0}^{\infty} k_i m_i^n \frac{x^n}{n!} \\
&= \sum_{n=0}^{\infty}\sum_{i=0}^{k} k_i m_i^n \frac{x^n}{n!} \\
&= \sum_{n=0}^{\infty}\left[\frac{x^n}{n!}\sum_{i=0}^{k}\left[k_i m_i^n\right]\right], \quad \exists k_i \in \mathbb{C},\ \ \forall i \in \mathbb{Z}^+ \cap [1,k] \qquad (12)
\end{aligned}
$$

Recall the definition of the generating function from 11, by relating this to (12):

$$
\begin{aligned}
f(x) &= \sum_{n=0}^{\infty}\left[\frac{x^n}{n!}a_n\right] \\
&= \sum_{n=0}^{\infty}\left[\frac{x^n}{n!}\sum_{i=0}^{k}\left[k_i m_i^n\right]\right] \\
\implies a_n &= \sum_{n=0}^{k}\left[k_i m_i^n\right]
\end{aligned}
$$

□

This can be verified by the fibonacci sequence as shown in section 1.2.2 , the solution to the characteristic equation is $m_1 = \varphi, m_2 = (1-\varphi)$ and the corresponding solution to the linear ODE and recursive relation are:

$$
\begin{aligned}
f(x) =& c_1 e^{\varphi x} + c_2 e^{(1-\varphi)x}, \quad \exists c_1, c_2 \in \mathbb{R} \subset \mathbb{C} \\
\iff a_n =& k_1 n^{\varphi} + k_2 n^{1-\varphi}, \qquad \exists k_1, k_2 \in \mathbb{R} \subset \mathbb{C}
\end{aligned}
$$

Repeated Roots of Characteristic Equation

Example Consider the following recurrence relation:

$$
\begin{aligned}
a_n - 10a_{n+1} + 25a_{n+2} = 0 \qquad (13) \\
\implies \sum_{n=0}^{\infty}\left[a_n \frac{x^n}{n!}\right] - 10\sum_{n=0}^{\infty}\left[\frac{x^n}{n!}+\right] + 25\sum_{n=0}^{\infty}\left[a_{n+2}\frac{x^n}{n!}\right] = 0
\end{aligned}
$$

By applying the definition of the exponential generating function at (3) :

$$
f''(x) - 10f'(x) + 25f(x) = 0
$$

By implementing the already well-established theory of linear ODE's, the characteristic equation for (**??**) can be expressed as:

$$\begin{aligned} m^2 - 10m + 25 &= 0 \\ (m-5)^2 &= 0 \\ m &= 5 \end{aligned} \tag{14}$$

Herein lies a complexity, in order to solve this, the solution produced from (14) can be used with the *Reduction of Order* technique to produce a solution that will be of the form [27, §4.3].

$$f(x) = c_1 e^{5x} + c_2 x e^{5x} \tag{15}$$

(15) can be expressed in terms of the exponential power series in order to try and relate the solution for the function back to the generating function, observe however the following power series identity (TODO Prove this in section):

$$x^k e^x = \sum_{n=0}^{\infty} \left[\frac{x^n}{(n-k)!} \right], \quad \exists k \in \mathbb{Z}^+ \tag{16}$$

by applying identity (16) to equation (15)

$$\begin{aligned} \implies f(x) &= \sum_{n=0}^{\infty} \left[c_1 \frac{(5x)^n}{n!} \right] + \sum_{n=0}^{\infty} \left[c_2 n \frac{(5x^n)}{n(n-1)!} \right] \\ &= \sum_{n=0}^{\infty} \left[\frac{x^n}{n!} (c_1 5^n + c_2 n 5^n) \right] \end{aligned}$$

Given the defenition of the exponential generating function from (3)

$$\begin{aligned} f(x) &= \sum_{n=0}^{\infty} \left[a_n \frac{x^n}{n!} \right] \\ \iff a_n &= c_{15}^n + c_2 n_5^n \end{aligned}$$

□

Generalised Example

Proof In order to prove the the solution for a k^{th} order recurrence relation with k repeated Consider a recurrence relation of the form:

$$\begin{aligned} \sum_{n=0}^{k} [c_i a_n] &= 0 \\ \implies \sum_{n=0}^{\infty} \sum_{i=0}^{k} c_i a_n \frac{x^n}{n!} &= 0 \\ \sum_{i=0}^{k} \sum_{n=0}^{\infty} c_i a_n \frac{x^n}{n!} & \end{aligned}$$

By substituting for the value of the generating function (from (3)):

$$\sum_{i=0}^{k}\left[c_i f^{(k)}(x)\right] \tag{17}$$

Assume that (17) corresponds to a charecteristic polynomial with only 1 root of multiplicity k, the solution would hence be of the form:

$$\begin{align}
&\sum_{i=0}^{k}\left[c_i m^i\right]=0\wedge m=B,\ \ \exists! B\in\mathbb{C} \nonumber\\
\implies\ & f(x)=\sum_{i=0}^{k}\left[x^i A_i e^{mx}\right],\quad \exists A\in\mathbb{C}^{+},\ \ \forall i\in[1,k]\cap\mathbb{N} \tag{18}\\
& \tag{19}
\end{align}$$

Recall the following power series identity (proved in section xxx):

$$x^k e^x=\sum_{n=0}^{\infty}\left[\frac{x^n}{(n-k)!}\right]$$

By applying this to (18) :

$$\begin{align}
f(x)&=\sum_{i=0}^{k}\left[A_i\sum_{n=0}^{\infty}\left[\frac{(xm)^n}{(n-i)!}\right]\right] \nonumber\\
&=\sum_{n=0}^{\infty}\left[\sum_{i=0}^{k}\left[\frac{x^n}{n!}\frac{n!}{(n-i)}A_i m^n\right]\right] \tag{20}\\
&=\sum_{n=0}^{\infty}\left[\frac{x^n}{n!}\sum_{i=0}^{k}\left[\frac{n!}{(n-i)}A_i m^n\right]\right] \tag{21}
\end{align}$$

Recall the generating function that was used to get 17:

$$\begin{align}
f(x)&=\sum_{n=0}^{\infty}\left[a_n\frac{x^n}{n!}\right] \nonumber\\
\implies\ a_n&=\sum_{i=0}^{k}\left[A_i\frac{n!}{(n-i)!}m^n\right] \nonumber\\
&=\sum_{i=0}^{k}\left[m^n A_i\prod_{0}^{k}[n-(i-1)]\right] \tag{22}
\end{align}$$

$\because\ \ i\leq k$

$$=\sum_{i=0}^{k}\left[A_i^{*}m^n n^i\right],\quad \exists A_i\in\mathbb{C},\ \ \forall i\in\mathbb{Z}^{+}$$

$\square$

General Proof

In sections 1.2.4 and 1.2.4 it was shown that a recurrence relation can be related to an ODE and then that solution can be transformed to provide a solution for the recurrence relation, when the charecteristic polynomial has either complex roots or 1 repeated root. Generally the solution to a linear ODE will be a superposition of solutions for each root, repeated or unique and so here it will be shown that these two can be combined and that the solution will still hold.

Consider a Recursive relation with constant coefficients:

$$\sum_{n=0}^{\infty} \left[c_i \cdot a_n\right] = 0, \quad \exists c \in \mathbb{R}, \ \forall i < k \in \mathbb{Z}^+$$

This can be expressed in terms of the exponential generating function:

$$\sum_{n=0}^{\infty} \left[c_i \cdot a_n\right] = 0 \implies \sum_{n=0}^{\infty} \left[\sum_{n=0}^{\infty} \left[c_i \cdot a_n\right]\right] = 0$$

- Use the Generating function to get an ODE
- The ODE will have a solution that is a combination of the above two forms
- The solution will translate back to a combination of both above forms

¶ 1.3 Fibonacci Sequence and the Golden Ratio

The *Fibonacci Sequence* is actually very interesting, observe that the ratios of the terms converge to the *Golden Ratio*:

$$\begin{aligned}
F_n &= \frac{\varphi^n - \psi^n}{\varphi - \psi} = \frac{\varphi^n - \psi^n}{\sqrt{5}} \\
\iff \frac{F_{n+1}}{F_n} &= \frac{\varphi^{n+1} - \psi^{n+1}}{\varphi^n - \psi^n} \\
\iff \lim_{n\to\infty}\left[\frac{F_{n+1}}{F_n}\right] &= \lim_{n\to\infty}\left[\frac{\varphi^{n+1} - \psi^{n+1}}{\varphi^n - \psi^n}\right] \\
&= \frac{\varphi^{n+1} - \lim_{n\to\infty}\left[\psi^{n+1}\right]}{\varphi^n - \lim_{n\to\infty}\left[\psi^n\right]} \\
\text{because } \mid \psi \mid < 0 \ n \to \infty \implies \psi^n \to 0: & \\
&= \frac{\varphi^{n+1} - 0}{\varphi^n - 0} \\
&= \varphi
\end{aligned}$$

We'll come back to this later on when looking at spirals and fractals.

This can also be shown by using analysis, let \$L$=\lim_{\mathsf{n} \to \infty}\left[\frac{\{}{F}\mathsf{n+1}\}\{F_{\mathsf{n}}\}\right], then:$

$$L\& = \frac{\{}{F}_{\mathsf{n+1}}\}\{f_{\mathsf{n}}\} \tag{23}$$

$$\& = \frac{\{}{F}_{\mathsf{n}} + F_{\mathsf{n-1}}\}\{F_{\mathsf{n}}\} \tag{24}$$

$$\& = 1 + \frac{\{}{F}_{\mathsf{n-1}}\}\{F_{\mathsf{n}}\} \tag{25}$$

$$\tag{26}$$

Se This Tutorial

References

[1] *5. Data Structures Python 3.8.5 Documentation*. URL: https://docs.python.org/3/tutorial/datastructures.html (visited on 08/24/2020).

[2] A. C Benander, B. A Benander, and Janche Sang. "An Empirical Analysis of Debugging Performance Differences between Iterative and Recursive Constructs". In: *Journal of Systems and Software* 54.1 (Sept. 30, 2000), pp. 17–28. ISSN: 0164-1212. DOI: 10.1016/S0164-1212(00)00023-6. URL: http://www.sciencedirect.com/science/article/pii/S0164121200000236 (visited on 08/24/2020).

[3] Corrado Böhm. "Reducing Recursion to Iteration by Algebraic Extension: Extended Abstract". In: *ESOP 86*. Ed. by Bernard Robinet and Reinhard Wilhelm. Red. by G. Goos et al. Vol. 213. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 111–118. ISBN: 978-3-540-16442-5 978-3-540-39782-3. DOI: 10.1007/3-540-16442-1_8. URL: http://link.springer.com/10.1007/3-540-16442-1_8 (visited on 08/24/2020).

[4] Corrado Böhm. "Reducing Recursion to Iteration by Means of Pairs and N-Tuples". In: *Foundations of Logic and Functional Programming*. Ed. by Mauro Boscarol, Luigia Carlucci Aiello, and Giorgio Levi. Red. by G. Goos et al. Vol. 306. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 58–66. ISBN: 978-3-540-19129-2 978-3-540-39126-5. DOI: 10.1007/3-540-19129-1_3. URL: http://link.springer.com/10.1007/3-540-19129-1_3 (visited on 08/24/2020).

[5] Anne M. Burns. ""Persian" Recursion". In: *Mathematics Magazine* 70.3 (1997), pp. 196–199. ISSN: 0025-570X. DOI: 10.2307/2691259. JSTOR: 2691259.

[6] *Emergence How Stupid Things Become Smart Together*. Nov. 16, 2017. URL: https://www.youtube.com/watch?v=16W7c0mb-rE (visited on 08/25/2020).

[7] Peter Farrell. *Math Adventures with Python: An Illustrated Guide to Exploring Math with Code*. San Francisco: No Starch Press, 2019. 276 pp. ISBN: 978-1-59327-867-0.

[8] Michael Fried. "Mathematics as the Science of Patterns - Mathematics as the Science of Patterns | Mathematical Association of America". In: *Convergence* (Aug. 2010). URL: https://www.maa.org/press/periodicals/convergence/mathematics-as-the-science-of-patterns-mathematics-as-the-science-of-patterns (visited on 08/24/2020).

[9] *Functools Higher-Order Functions and Operations on Callable Objects Python 3.8.5 Documentation*. URL: https://docs.python.org/3/library/functools.html (visited on 08/25/2020) (cit. on p. 1).

[10] *Gnuplot / Fractal / Mandelbrot (E)*. URL: http://folk.uio.no/inf3330/scripting/doc/gnuplot/Kawano/fractal/mandelbrot-e.html (visited on 08/25/2020).

[11] G. H Hardy. *A Mathematician's Apology*. OCLC: 1172189203. 2012. ISBN: 978-1-107-29559-9 978-1-107-60463-6. URL: http://dx.doi.org/10.1017/CBO9781107295599 (visited on 08/24/2020).

[12] Roozbeh Hazrat. *Mathematicaő: A Problem-Centered Approach*. 2nd ed. 2015. Springer Undergraduate Mathematics Series. Cham: Springer International Publishing : Imprint: Springer, 2015. 1 p. ISBN: 978-3-319-27585-7. DOI: 10.1007/978-3-319-27585-7 (cit. on p. 1).

[13] *Iteration vs. Recursion - CS 61A Wiki*. Dec. 19, 2016. URL: https://www.ocf.berkeley.edu/~shidi/cs61a/wiki/Iteration_vs._recursion (visited on 08/24/2020).

[14] *Julia Set*. In: *Wikipedia*. Page Version ID: 967264809. July 12, 2020. URL: https://en.wikipedia.org/w/index.php?title=Julia_set&oldid=967264809 (visited on 08/25/2020).

[15] Sophia Kivelson and Steven A. Kivelson. "Defining Emergence in Physics". In: *npj Quantum Materials* 1.1 (1 Nov. 25, 2016), pp. 1–2. ISSN: 2397-4648. DOI: 10.1038/npjquantmats.2016.24. URL: https://www.nature.com/articles/npjquantmats201624 (visited on 08/25/2020).

[16] Eric Lehman, Tom Leighton, and Albert Meyer. *Readings | Mathematics for Computer Science | Electrical Engineering and Computer Science | MIT OpenCourseWare*. Sept. 8, 2010. URL: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2010/readings/ (visited on 08/10/2020) (cit. on p. 6).

[17] Oscar Levin. *Solving Recurrence Relations*. Jan. 29, 2018. URL: http://discrete.openmathbooks.org/dmoi2/sec_recurrence.html (visited on 08/11/2020) (cit. on p. 7).

[18] *Mandelbrot Set - Rosetta Code*. URL: https://rosettacode.org/wiki/Mandelbrot_set#Python (visited on 08/25/2020).

[19] *Multi-Dimensional Arrays û The Julia Language*. URL: https://docs.julialang.org/en/v1/manual/arrays/#man-comprehensions-1 (visited on 08/24/2020).

[20] Olympia Nicodemi, Melissa A. Sutherland, and Gary W. Towsley. *An Introduction to Abstract Algebra with Notes to the Future Teacher*. OCLC: 253915717. Upper Saddle River, NJ: Pearson Prentice Hall, 2007. 436 pp. ISBN: 978-0-13-101963-8 (cit. on p. 7).

[21] Heinz-Otto Peitgen, H. Jürgens, and Dietmar Saupe. *Chaos and Fractals: New Frontiers of Science*. 2nd ed. New York: Springer, 2004. 864 pp. ISBN: 978-0-387-20229-7.

[22] A.P. Sinha and I. Vessey. "Cognitive Fit: An Empirical Study of Recursion and Iteration". In: *IEEE Transactions on Software Engineering* 18.5 (May 1992), pp. 368–379. ISSN: 00985589. DOI: 10.1109/32.135770. URL: http://ieeexplore.ieee.org/document/135770/ (visited on 08/24/2020).

[23] S Smolarski. *Math 60 – Notes A3: Recursion vs. Iteration*. Feb. 9, 2000. URL: http://math.scu.edu/~dsmolars/ma60/notesa3.html (visited on 08/24/2020).

[24] V. Anton Spraul. *How Software Works: The Magic behind Encryption, CGI, Search Engines, and Other Everyday Technologies*. San Francisco: No Starch Press, 2015. 198 pp. ISBN: 978-1-59327-666-9. URL: https://learning.oreilly.com/library/view/how-software-works/9781457189968/ (visited on 08/24/2020).

[25] Alan Turing. "The Chemical Basis of Morphogenesis". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641 (Aug. 14, 1952), pp. 37–72. ISSN: 2054-0280. DOI: 10.1098/rstb.1952.0012. URL: https://royalsocietypublishing.org/doi/10.1098/rstb.1952.0012 (visited on 08/25/2020).

[26] Dennis G Zill and Michael R Cullen. *Differential Equations*. 7th ed. Brooks/Cole, 2009 (cit. on pp. 7, 8).

[27] Dennis G. Zill and Michael R. Cullen. "8.4 Matrix Exponential". In: *Differential Equations with Boundary-Value Problems*. 7th ed. Belmont, CA: Brooks/Cole, Cengage Learning, 2009. ISBN: 978-0-495-10836-8 (cit. on pp. 7, 10).