

ASSIGNED FORUM PROBLEM 3

Soobin Rho

June 28, 2022

Chapter 8.4: Trigonometric Substitutions

$$24. \int_0^1 \frac{dx}{(4-x^2)^{\frac{3}{2}}} \quad (1)$$

GENERAL GUIDELINES

Please read these instructions carefully. The objective of this template is to enable you in an easy way to style your article attractively in a style similar to that of the typeset journal. It should be emphasized, however, that the final appearance of your paper in print and in electronic media may likely vary to some extent from the presentation achieved in this template.

```
[3]: # I'm from South Korea, but I grew up in Dubai.  
print('Hello, my name is: Soobin Rho')
```

Hello, my name is: Soobin Rho

Level Headings: Subsections

You will usually want to divide your article into (numbered) sections and subsections (perhaps even subsubsections). Use to help organize your document as appropriate. Include hyperlinks using URL <http://www.google.com>.

Hi! This is for my **Math 125** class.

```
[9]: '''Generates a random sentence.'''
from random import randint

def generate_random_sentence() -> str:
    '''
        1. Picks an article and capitalizes the first letter.
        2. Picks a noun.
        3. Picks a verb.
        4. Picks a preposition.
        5. Picks another article.
        6. Picks another noun and ends with a period.
    '''

    # Initializing the return value.
    sentence = ""

    # 1. Picks an article and capitalizes the first letter.
    article = array_articles[randint(0, len(array_articles) - 1)]
    article = article.capitalize()
    sentence += f"{article} "

    # 2. Picks a noun.
    noun = array_nouns[
        randint(0, len(array_nouns) - 1)
    ]
    sentence += f"{noun} "

    # 3. Picks a verb.
    verb = array_verbs[
        randint(0, len(array_verbs) - 1)
    ]
    sentence += f"{verb} "

    # 4. Picks a preposition.
    preposition = array_prepositions[
        randint(0, len(array_prepositions) - 1)
    ]
    sentence += f"{preposition} "

    # 5. Picks another article.
    article = array_articles[
        randint(0, len(array_articles) - 1)
    ]
    sentence += f"{article} "

    # 6. Picks another noun and ends with a period.
    noun = array_nouns[randint(0, len(array_nouns) - 1)]
    sentence += f"{noun}."

    return sentence
```

```
[13]: # Printing the title.
str_title = 'Random Sentence Generator'
print(len(str_title) * '=')
print(str_title)
print((len(str_title) * '=') + '\n')

# Generate 20 random sentences.
for count in range(20):
    print(generate_random_sentence())
```

```
=====
Random Sentence Generator
=====
```

A(n) arrays becomes on a(n) use.
 The true write up the number.
 A(n) arrays upload despite a(n) list.
 The true download down a(n) operators.

A(n) float determine at the players.
The radius download following a(n) value.
The here submitted since a(n) chosen.
A(n) circle download within the order.
A(n) constant picked off a(n) uses.
The asterisks perform around the combining.
The then spelled with a(n) assignment.
A(n) put printed plus the forward.
The while perform like the values.
A(n) one entered of a(n) three.
A(n) reverse spelled out the true.
A(n) rock submit on the homework.
A(n) parameter picked during the area.
A(n) draw adding up the while.
A(n) prints becomes during a(n) numbers.
The form spelled against a(n) cells.

MATHEMATICS

Here is some mathematics. For $A \in M_n$, the factorization $A = LU$, where L is unit lower triangular and U is upper triangular, is called the *LU decomposition*, or *LU factorization*. We can use such a factorization, when it exists, to solve the system $A\mathbf{x} = \mathbf{b}$ by first solving for the vector \mathbf{y} in $L\mathbf{y} = \mathbf{b}$ and then solving $U\mathbf{x} = \mathbf{y}$. However, not every $n \times n$ matrix A has an LU decomposition. The following theorem provides conditions for the existence and uniqueness of an LU decomposition of a $n \times n$ matrix.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2)$$

Example 1. The 3×3 matrix $A = \begin{bmatrix} 1 & 5 & 1 \\ 1 & 4 & 2 \\ 4 & 10 & 2 \end{bmatrix}$ has all non-zero principle minors, A_1, A_2 and A_3 . Therefore, there is a unique LU factorization with both L and U nonsingular given by

$$\begin{bmatrix} 1 & 5 & 1 \\ 1 & 4 & 2 \\ 4 & 10 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 10 & 1 \end{bmatrix} \begin{bmatrix} 1 & 5 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 12 \end{bmatrix}.$$

DEFINITION, THEOREM, COROLLARY

Definition 1. *Definitions are if and only if statements.*

`\begin{definition}`

Definitions are if and only if statements.

`\end{definition}`

Theorem 2 (Matrices with Infinitely Many LU Factorizations). *For $A \in M_n$, if two or more of any first $(n - 1)$ columns are linearly dependent or any of the first $(n - 1)$ columns are 0, then A has infinitely many LU factorizations.*

Proof. We will prove only for the the case when $A \in M_3$.

$$dm + r = e \Rightarrow r = e - dm \quad (3)$$

$$dn + rp = f \Rightarrow p = \frac{f - dn}{r} \quad (4)$$

$$gm + s = h \Rightarrow s = h - gm \quad (5)$$

$$gn + sp + t = i \Rightarrow t = i - sp - gn \quad (6)$$

□

Corollary 3. *If x , then y .*

EXAMPLES

Here is an example of an example.

Example 2. *Let $\{1, 2, 3\}$ and $\{2, 1, 3\}$ be two lists of integers. Then, to check if the two lists are equal we would have,*

$$\{1, 2, 3\} == \{2, 1, 3\} .$$

```
\begin{example}
Let  $\{1, 2, 3\}$  and  $\{2, 1, 3\}$  be two lists of integers. Then, to check
\begin{center}
\texttt{\{1, 2, 3\} == \{2, 1, 3\}}.
\end{center}\label{ex:equallists}
\end{example}
```

LISTS

NAGJ uses the **outline** package.

Enumerated List

The following code produces an enumerated list. The enumerated list numbers each list item with Arabic numerals.

```
\begin{outline}[enumerate]
  \1 First Level
    \2 Second level
      \3 Third level
\end{outline}
```

1. First Level

(a). Second level

i. Third level

Itemized List

The following code produces an itemized list.

```
\begin{outline}
  \1 First item
    \2 Second level item
      \3 Third level sub-item
\end{outline}
```

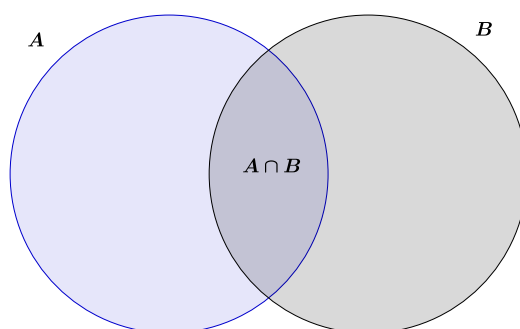
- First Level
 - Second level
 - * Third level

TABLES AND FIGURES

A **caption** that briefly describes the material is required for figures and tables. Any required information, such as photo credits or data source, may be included in captions. Furthermore, as part of the authorization to use that content, providers of this material may need a specific credit line, which might be placed in the caption (or wherever the provider has requested).

Figures

Figures should be high quality (1200 dpi for line art, 600 dpi for grayscale and 300 dpi for color, at the correct size). The preferred method of including graphics in the *North American GeoGebra Journal* is to export to TiKz. Other acceptable file formats include: EPS, PS, PNG, JPEG, or TIFF.



Tables

We use the **booktabs** package. Tables should present new information rather than duplicating what is in the text. Readers should be able to interpret the table without reference to the text. Please supply editable files.

Date	Time	Average	Standard Deviation
Jan 1	1100	4.7	0.6
Jan 2	2300	16.7	2.9
Jan 3	1400	11.4	3.5
Jan 4	1130	8.4	2.1
Jan 5	500	5.2	1.9
Jan 6	1700	7.9	2.2