# Autonomous Development of Online Assignments Supported by the Large Language Models Related to Mathematics, Computer Science, and Engineering

Andrew Pownuk[1] and M.V. Rama Rao[2]

1 - University of Texas at El Paso, El Paso, Texas, ampownuk@utep.edu
2 - Vasavi College of Engineering, Hyderabad, India, dr.mvrr@gmail.com

30th Joint UTEP/NMSU Workshop on Mathematics, Computer Science, and Computational Sciences
El Paso, Texas, October 28, 2023

# Outline

# Online Learning - ZyBooks - Multiple Choice Questions



**PARTICIPATION ACTIVITY** | 5.4.7: Representing floating-point numbers.

1) Floating-point values are always stored with some inaccuracy.
   - ○ True
   - ○ False

2) If a floating-point variable is assigned with 0.2, and prints as 0.2, the value must have been represented exactly.
   - ○ True
   - ○ False

Figure: ZyBooks - Multiple Choice Questions

# Online Learning - Blackboard - Type of Questions



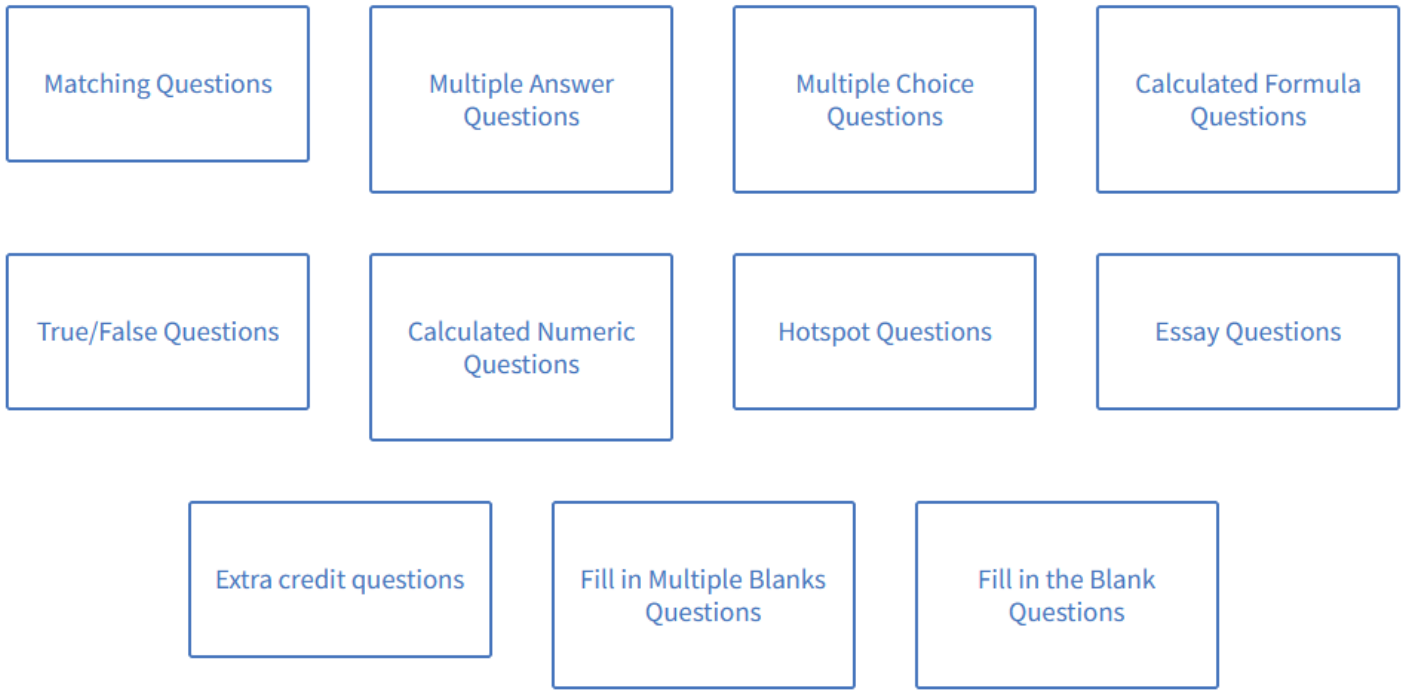



Figure: Blackboard - Type of Questions

# Online Learning - Moodle - PHP Code



```
require_once('../config.php');
require_once($CFG->libdir.'/blocklib.php');
require_once($CFG->dirroot.'/course/lib.php');
require_once('pagelib.php');

require_login();

$mymoodlestr = get_string('mymoodle','my');

if (isguest()) {
$wwwroot = $CFG->wwwroot.'/login/index.php';
if (!empty($CFG->loginhttps)) {
$wwwroot = str_replace('http:','https:', $wwwroot);
}

print_header($mymoodlestr);
notice_yesno(get_string('noguest', 'my').'<br /><br />'.get_string('liketologin'),
$wwwroot, $CFG->wwwroot);
print_footer();
die();
}
```

Figure: Moodle - PHP Code

# Online Learning - WebAssign - Perl Questions



Create Questions > Question Behavior > Perl Variables and Functions >

## Include Perl in Questions

### Example Multiple-Choice Question Using <eqn> and <EQN>

The following table summarizes an actual question.

| | |
|---|---|
| **QID** | |
| **Name** | |
| **Mode** | Multiple-Choice |
| **Question** | <eqn><br>$weight = randnum(40,150,1); # pick a random integer between 40 and 150<br>$venuswt = decform($weight * 0.907, 0); # calculate weight on Venus<br>$marswt = decform($weight * 0.377, 0); # calculate weight on Mars<br>$jupiterwt = decform($weight * 2.364, 0); # calculate weight on Jupiter<br>''<br></eqn><br>If you weighed <eqn $weight> kg on Earth:<br><br><ul><br><li>Your weight on Venus would be <eqn $venuswt> kg</li><br><li>Your weight on Mars would be <eqn $marswt> kg</li><br><li>Your weight on Jupiter would be <eqn $jupiterwt> kg</li><br></ul><br>On which planet would your mass be the least? <_> |
| **Answer** | <EQN $ORDERED=5; 'Earth'><br>Venus<br>Mars<br>Jupiter<br>Your mass would be the same |

Figure: WebAssign - Perl Questions

# Online Learning - ZyBooks - Code Evalutation



```
3  public class SensorThreshold {
4     public static void main(String[] args) {
5        Scanner scnr = new Scanner(System.in);
6        double targetValue;
7        double sensorReading;
8
9        targetValue = scnr.nextDouble();
10       sensorReading = scnr.nextDouble();
11
12       if (/* Your solution goes here  */) {
13          System.out.println("Equal");
14       }
15       else {
16          System.out.println("Not equal");
17       }
18    }
19 }
```

Figure: ZyBooks - Code Evalutation

# Online Homework - Numerical Analysis



```
public static double SimpsonsRule(
FunctionFDelegate f, double a, double b, int n)
{
if (n % 2 != 0){
throw new ArgumentException("n must be even"); }
    double h = (b - a) / n;
    double sum = f(a) + f(b);
    for (int i = 1; i < n; i += 2){
        double x = a + i * h;
        sum += 4 * f(x);}
    for (int i = 2; i < n - 1; i += 2){
        double x = a + i * h;
        sum += 2 * f(x);}
    return h / 3 * sum;
}
```

# Online Homework - HTML Code



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFil

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Homework</title>
  <script src="https://polyfill.io/v3/polyfill.min.js
  <script id="MathJax-script" async
          src="https://cdn.jsdelivr.net/npm/mathjax@3
  </script>
</head>
<body>
```

# Online Homework - C# Code



```
using System;
using System.Collections.Generic;
using System.Xml;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Homework_1 : System.Web.UI.Page
{
    public string HomeworkIDString = "Homework06";
    public bool debugMode = false;

    protected void Page_Load(object sender, EventArgs
    {
```

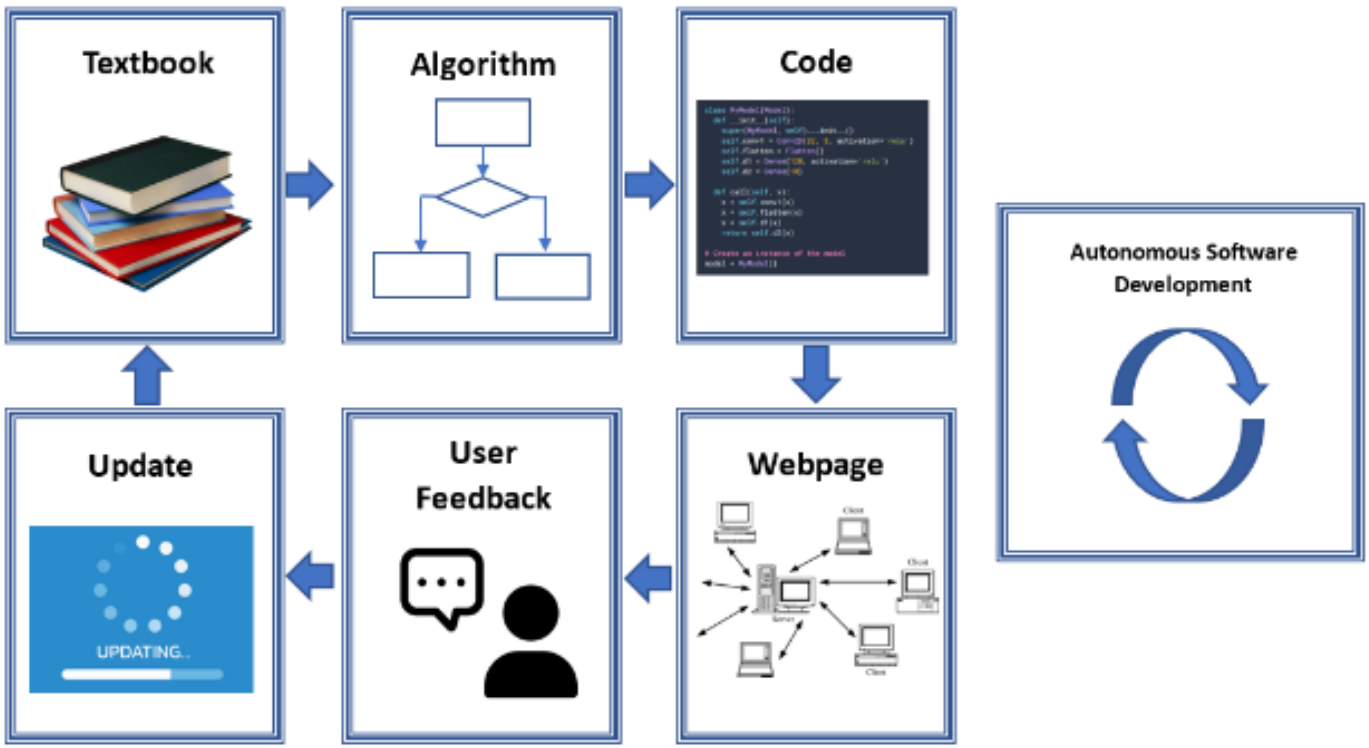
# Online Homework - PHP



```
<?php
session_start();
if (isset($_POST['username']) && isset($_POST['passwo
    $xml = simplexml_load_file('users.xml');
    $users = $xml->xpath("/users/user");
    foreach ($users as $user) {
        if ($user->username == $_POST['username'] &&
            $_SESSION['username'] = (string)$user->us
            header('Location: welcome.php');
            exit;
        }
    }
    $error = "Invalid username or password.";
}
?>
```

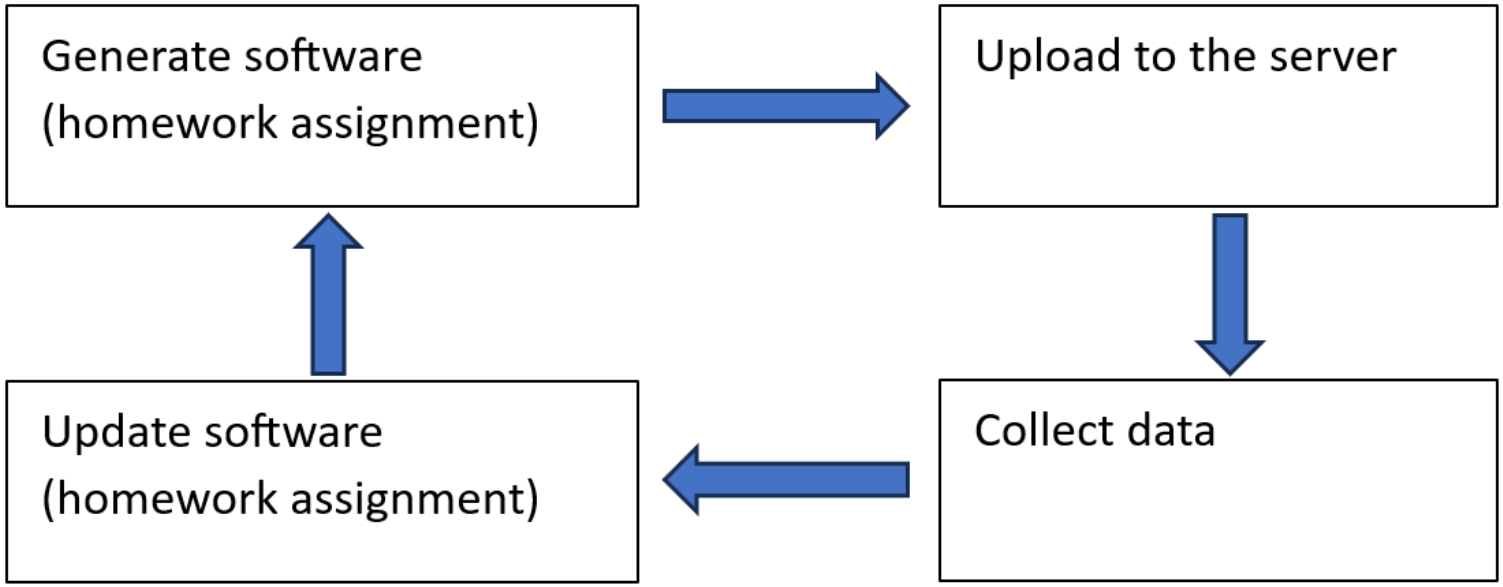
# Autonomous Algorithms





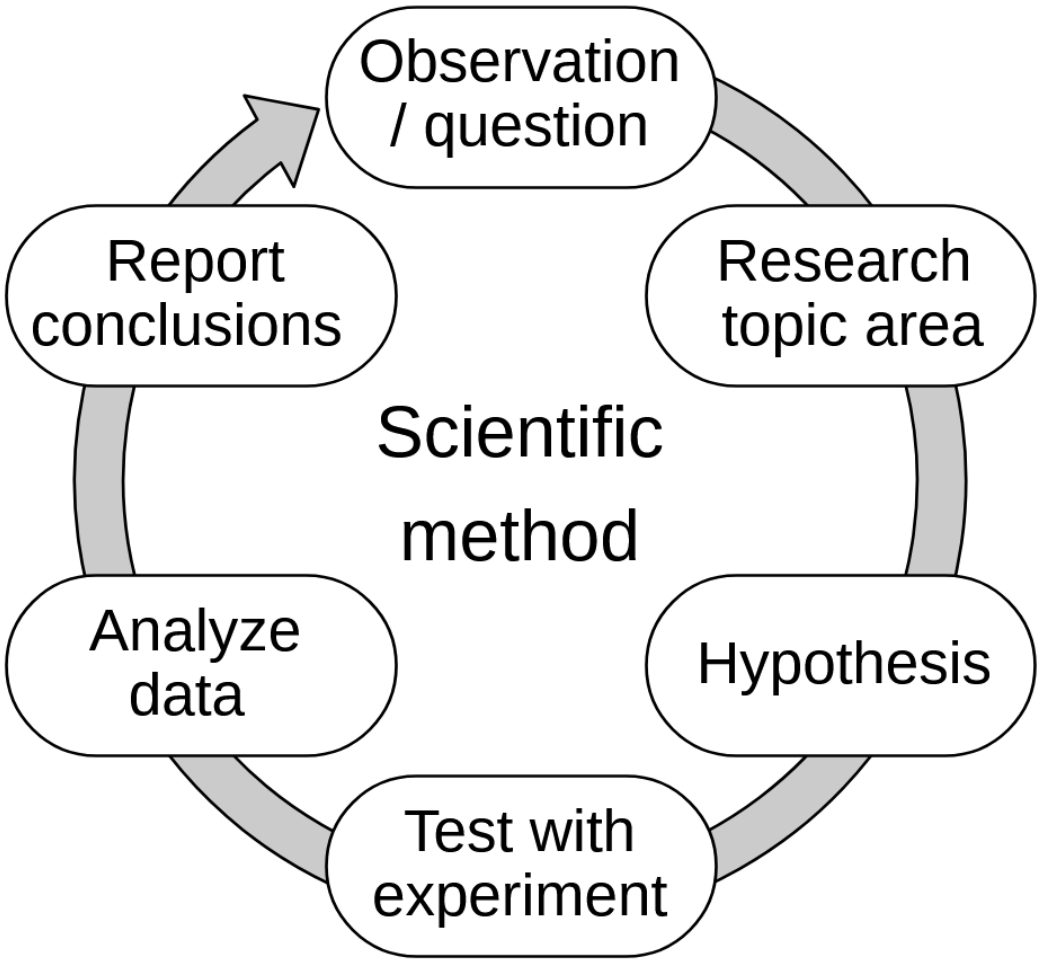
# Autonomous Algorithms





# Scientific Method (Aristotle 384–322 BCE)





# Write Speed




- Write speed HDD: 7833 pages per second.
- Write speed memory: 232098 pages per second.

# Autonomous Code Generation



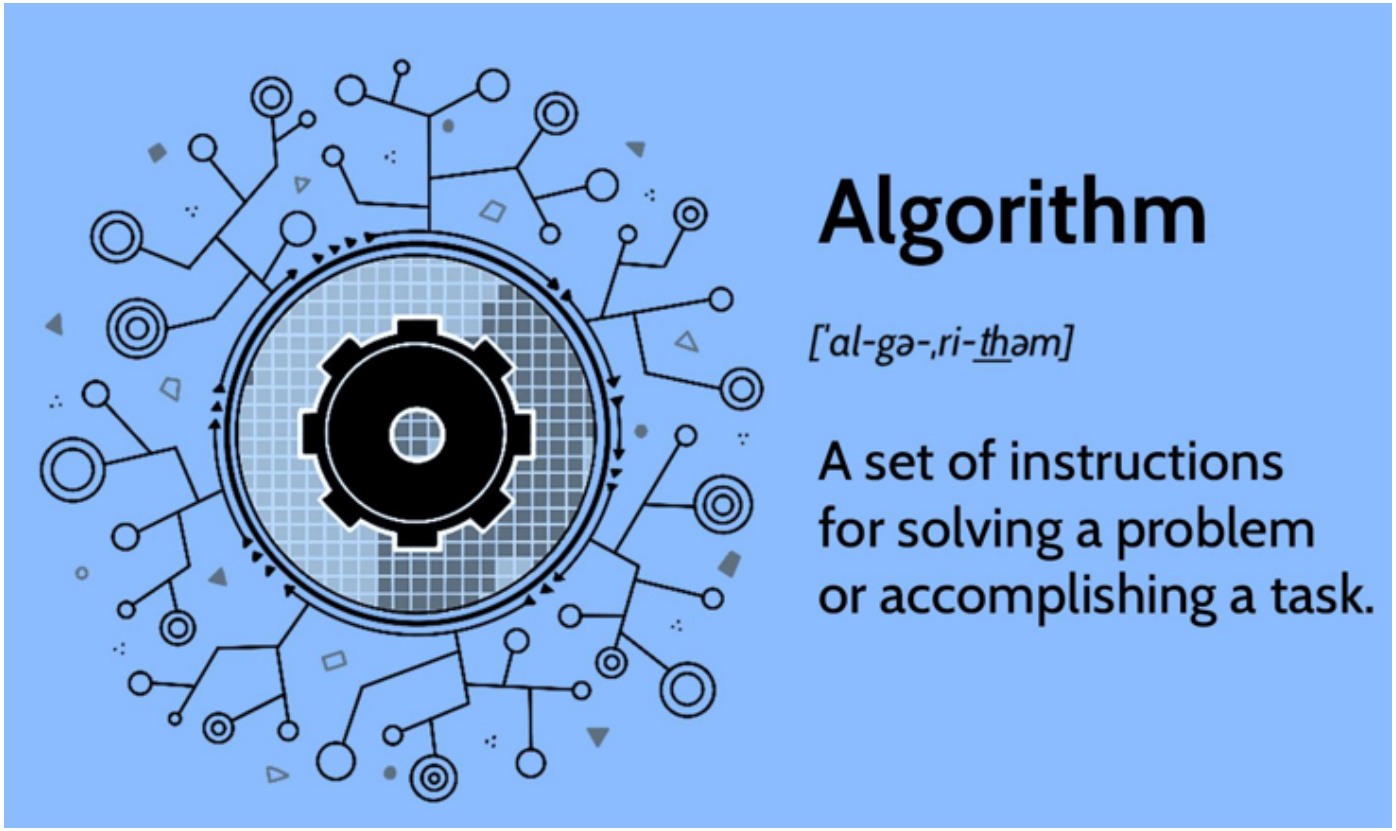
# Assignments



Academic education may be related to all scientific subjects known today and new research topics. In many cases for a given scientific topic it is possible to create related computer code which simulates some aspects of given scientific phenomena. Online assignments can be viewed as distributed programs which run in parallel on many electronic devices (computers, phones, tablets etc.) and communicate with the server. Students can interact with online web-applications and study related scientific topics. Instructors can constantly improve the quality of assignments and examples.

# Assignments and Related Algorithms





# Poster Presentation



## Applications of Autonomous Machine Learning Algorithms for Automated Development of Online Assignments

Evelyn Fang[1] and Andrew Pownuk[2]

[1] Student, El Paso Community College

[2] Mentor, University of Texas at El Paso

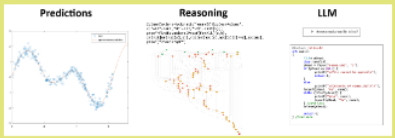
### Abstract

By using Turing complete computer languages (Java, C/C++, Python, etc.) it is possible to express a wide class of algorithms that cover much of the known scientific knowledge. In education it is necessary to study fundamental scientific principles and create online assignments that are related to various topics (computer programming, mathematics, engineering, etc.). With machine learning, new features of existing software can be predicted and a better version can be developed.

### Introduction

To create online assignment in automated/autonomous way it is possible to use approximation properties of various machine learning methods and data generated by appropriate software. In some cases instead of predictions it is more convenient to use the Monte Carlo tree search. Large language models can produce code which can be used in the development of online assignments.

### Predictions and Reasoning



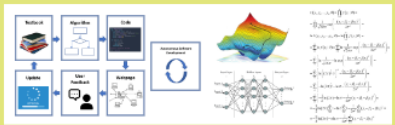

### Future Work

Some parts of the code and related web applications can be created in fully automated and autonomous way using approximation/prediction/classification, inference, and LLMs. It is necessary to constantly optimize the quality of the final code and integrate with new software, hardware, and scientific knowledge to find optimal tools for development of software used in science, engineering, and education.

### Background

Online assignments can be related to wide range of topics which are present in academic textbooks. By using Turing complete programming languages (C/C++, Java, Python etc.) it is possible to describe wide class of algorithms related to computer science, mathematics, engineering, etc.. Online assignments are computer programs which are related to various scientific and engineering problems.

### Autonomous Software



### Publication

A. Pownuk, I. Skalna, and E. Fang, Self-organizing Scientific Databases and Related Applications in Automated Development of Online Assignments, Abstracts of the 29th Joint NMSU/UTEP Workshop on Mathematics, Computer Science, and Computational Sciences, New Mexico State University, Las Cruces, April 1, 2023.

### Methodology

- Applications of machine learning to predict future form of scientific theories and related code from data.
- Applications of automated reasoning for autonomous development of online web-applications.
- Applications of large language models to generate code and build library of common assignments.

### Conclusions

Based theoretical research, literature review, and computational experiments it is possible to efficiently approximate large class of assignments (scientific problems) based on available data and use model prediction to create online assignment in autonomous way (in part). In some experiments autonomous methods for code development were much faster than traditional methods of software development.

### Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2034030. Any opinions, findings and conclusions or recommendations expressed in this material are these of the authors and do not necessarily reflect views of the National Science Foundation.

### References

[1] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[2] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018. eBook (Creative Commons Licensed).

[3] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

# Reimann Zeta Function



AN Define Riemann Zeta function and discuss their properties

The Riemann Zeta function, denoted by ζ(s), is a complex function that plays a fundamental role in number theory, complex analysis, and various areas of mathematics. It was introduced by the 19th-century German mathematician Bernhard Riemann. The function is defined as follows:

For a complex number s with a real part greater than 1, ζ(s) is given by the following series:

ζ(s) = 1^(-s) + 2^(-s) + 3^(-s) + 4^(-s) + ...

This series continues indefinitely, and ζ(s) is essentially the sum of the reciprocals of the powers of natural numbers raised to the complex power s.

Here are some important properties of the Riemann Zeta function:

1. Analytic Continuation: The original definition of ζ(s) is limited to values of s with a real part greater than 1. However, ζ(s) can be analytically continued to a larger domain in the complex plane, except for a simple pole at s = 1. This means that ζ(s) can be defined for a broader range of complex numbers, including those with real parts less than or equal to 1.
2. Euler's Identity: For s = 2, the Riemann Zeta function has a special value, and Euler's famous identity arises:

ζ(2) = 1^(-2) + 2^(-2) + 3^(-2) + 4^(-2) + ... = π^2 / 6

Regenerate

# De Morgan Law



**Proof:**

Suppose $x$ is an arbitrary element of the set on the left side, $(A \cup B)'$. This means that $x$ is not an element of $A \cup B$. In other words, $x$ does not belong to the union of sets A and B.

By the definition of set union, $x$ not belonging to $A \cup B$ implies that $x$ does not belong to A and $x$ does not belong to B. This can be written as $x \notin A$ and $x \notin B$.

Now, consider the set on the right side, $A' \cap B'$. An element $x$ is in this set if and only if it is both in $A'$ and in $B'$. In other words, $x$ does not belong to A and $x$ does not belong to B.

Since we have shown that $x$ does not belong to A and $x$ does not belong to B for both sets, it follows that $x$ belongs to both sets or, in other words, $x$ belongs to the intersection of their complements. Therefore, the sets on both sides are equal.

So, we have shown that $(A \cup B)' = A' \cap B'$, which proves the first De Morgan's Law.

1. The Second De Morgan's Law (Complement of Intersection):
   For any two sets A and B, the complement of their intersection is equal to the union of their complements:

$$(A \cap B)' = A' \cup B'$$

The proof of the second De Morgan's Law follows a similar logic to the first one, using set membership and element manipulation.

By understanding and applying De Morgan's Laws, you can simplify expressions involving set operations and complements in set theory, which is useful for various mathemat logical applications.

Regenerate

# Differential Equations



The given differential equation is $y' = \frac{x}{y}$. This is a first-order separable ordinary differential equation. To solve it, we can separate the variables and integrate both sides.

Separating the variables:

$$\frac{dy}{dx} = \frac{x}{y}$$
$$y\,dy = x\,dx$$

Now, integrate both sides:

$$\int y\,dy = \int x\,dx$$
$$\frac{1}{2}y^2 = \frac{1}{2}x^2 + C$$

Here, $C$ is the constant of integration.

Now, apply the initial condition $y(1) = 1$ to solve for $C$:

$$\frac{1}{2}(1)^2 = \frac{1}{2}(1)^2 + C$$
$$\frac{1}{2} = \frac{1}{2} + C$$
$$C = 0$$

Regenerate

# Geometry



To find the equation of the plane that passes through the points (0, 0, 0) and (1, 2, 2) and is parallel to the vector $\mathbf{v} = [5, 6, 7]$, we can use the point-normal form of the equation of a plane.

The point-normal form of the equation of a plane is given by:

$$A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$$

Where $(x_0, y_0, z_0)$ is a point on the plane, and $(A, B, C)$ is a normal vector to the plane.

We are given two points on the plane: $(0, 0, 0)$ and $(1, 2, 2)$. We can choose either point as our $(x_0, y_0, z_0)$. Let's choose $(0, 0, 0)$ for convenience.

The normal vector $(A, B, C)$ is parallel to $\mathbf{v} = [5, 6, 7]$, which means they are scalar multiples of each other.

So, we can write:

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = k \begin{pmatrix} 5 \\ 6 \\ 7 \end{pmatrix}$$

for some scalar $k$.

Regenerate

# Sample Problem in Mathematics



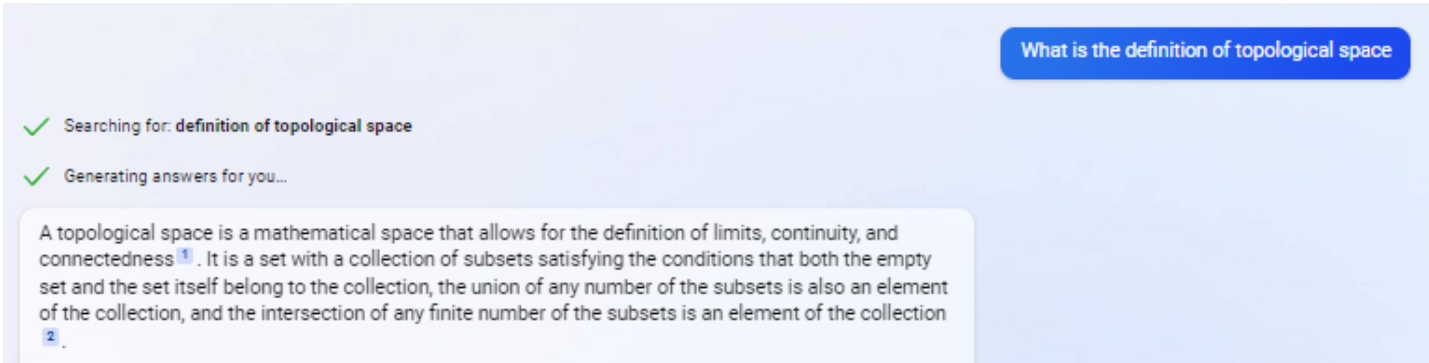



Figure: Definition of the topological space

# Sample Problem in Statistics



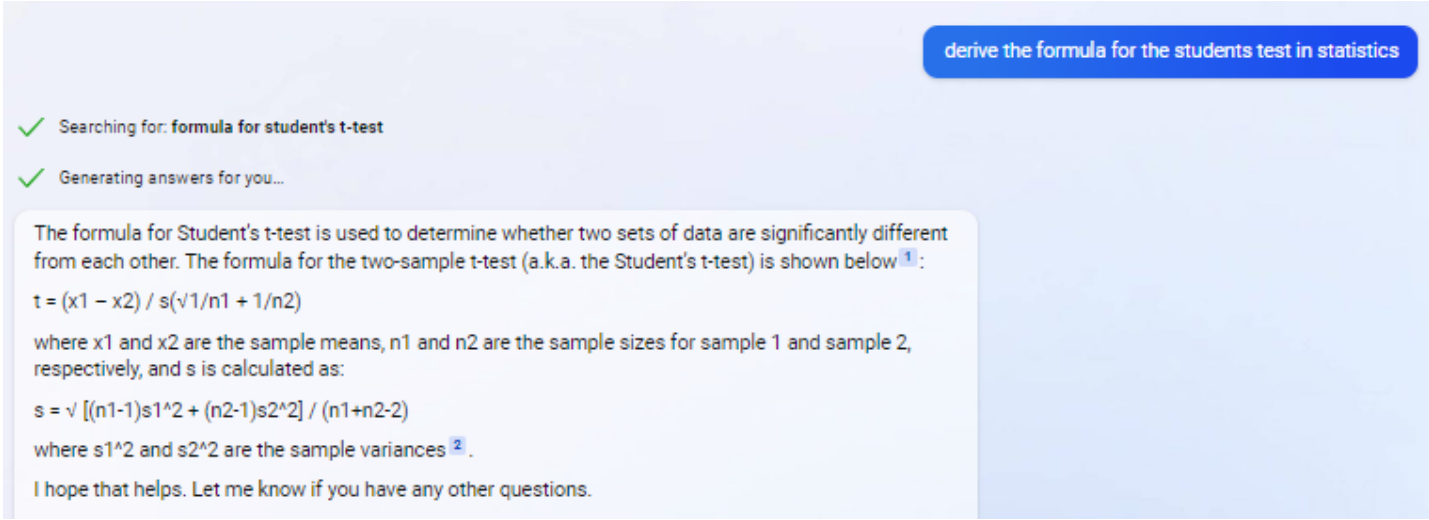
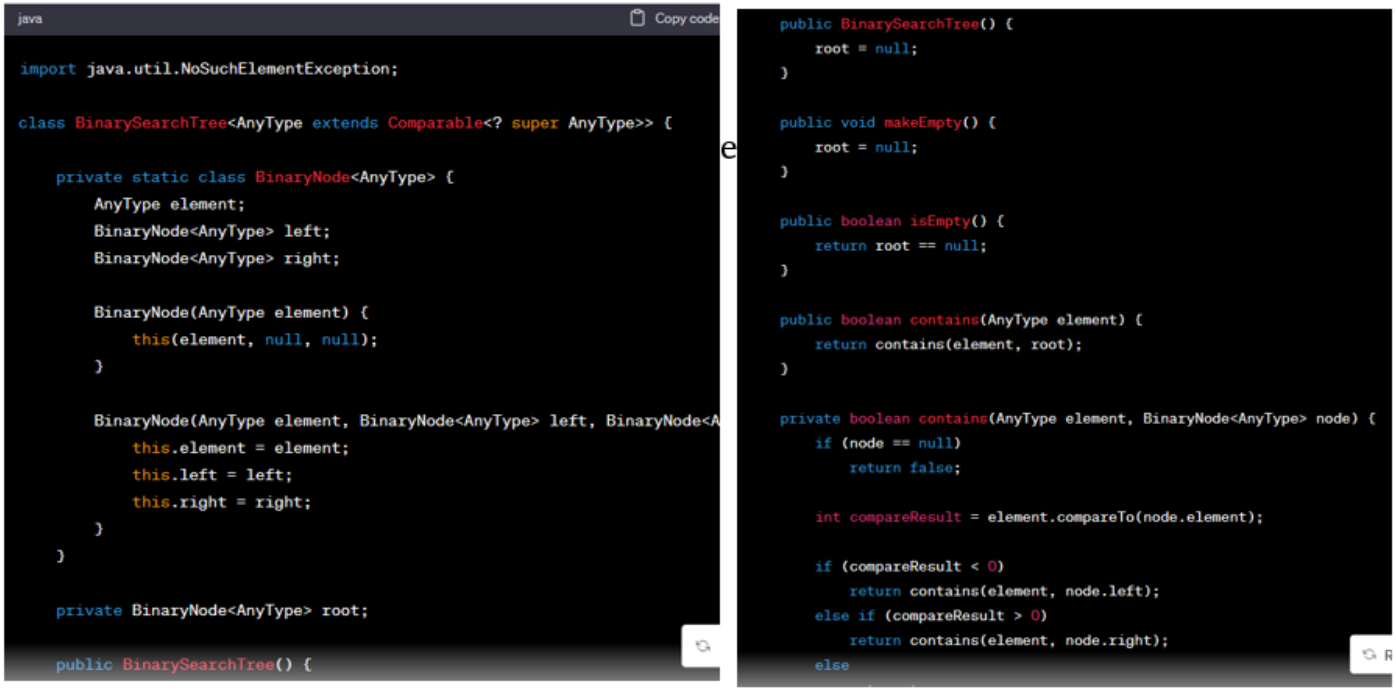



Figure: Student's t-test

# Java Code Generation





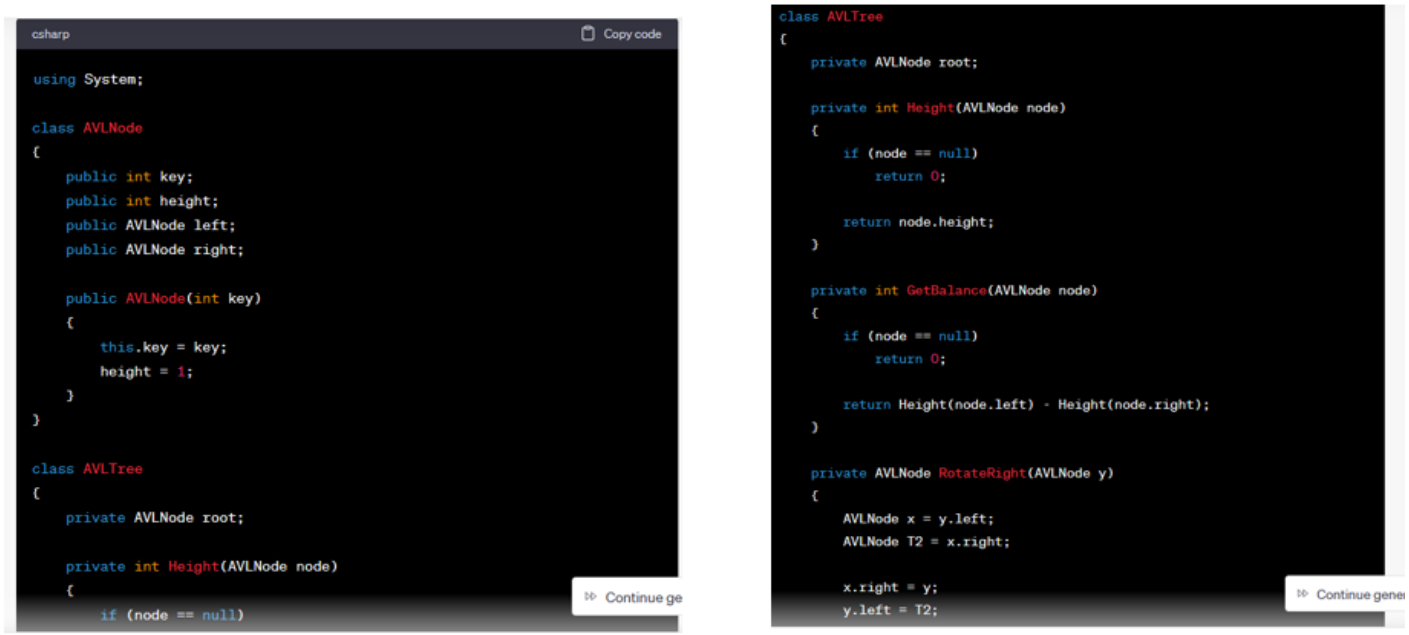
```java
import java.util.NoSuchElementException;

class BinarySearchTree<AnyType extends Comparable<? super AnyType>> {

    private static class BinaryNode<AnyType> {
        AnyType element;
        BinaryNode<AnyType> left;
        BinaryNode<AnyType> right;

        BinaryNode(AnyType element) {
            this(element, null, null);
        }

        BinaryNode(AnyType element, BinaryNode<AnyType> left, BinaryNode<A
            this.element = element;
            this.left = left;
            this.right = right;
        }
    }

    private BinaryNode<AnyType> root;

    public BinarySearchTree() {
```

```java
    public BinarySearchTree() {
        root = null;
    }

    public void makeEmpty() {
        root = null;
    }

    public boolean isEmpty() {
        return root == null;
    }

    public boolean contains(AnyType element) {
        return contains(element, root);
    }

    private boolean contains(AnyType element, BinaryNode<AnyType> node) {
        if (node == null)
            return false;

        int compareResult = element.compareTo(node.element);

        if (compareResult < 0)
            return contains(element, node.left);
        else if (compareResult > 0)
            return contains(element, node.right);
        else
```

Figure: Binary Search Tree in Java

# C# Code Generation





```csharp
using System;

class AVLNode
{
    public int key;
    public int height;
    public AVLNode left;
    public AVLNode right;

    public AVLNode(int key)
    {
        this.key = key;
        height = 1;
    }
}

class AVLTree
{
    private AVLNode root;

    private int Height(AVLNode node)
    {
        if (node == null)
            return 0;

        return node.height;
    }

    private int GetBalance(AVLNode node)
    {
        if (node == null)
            return 0;

        return Height(node.left) - Height(node.right);
    }

    private AVLNode RotateRight(AVLNode y)
    {
        AVLNode x = y.left;
        AVLNode T2 = x.right;

        x.right = y;
        y.left = T2;
```

Figure: AVL Tree in C#

# Implementation of Homework assignments by Using Computer Code



```
public class TopologicalSort
{
    private int V; // Number of vertices
    private Dictionary<int, List<int>> graph;

    public TopologicalSort(int V)
    {
        this.V = V;
        graph = new Dictionary<int, List<int>>();
        for (int i = 0; i < V; i++)
        {
            graph[i] = new List<int>();
        }
    }

    public void AddEdge(int u, int v)
    {
        graph[u].Add(v);
    }

    public List<int> TopologicalSortMethod()
    {
        List<int> result = new List<int>();
        bool[] visited = new bool[V];
        Stack<int> stack = new Stack<int>();

        for (int i = 0; i < V; i++)
        {
            if (!visited[i])
            {
                TopologicalSortUtil(i, visited, stack);
            }
```

```
public class PrimMST
{
    public static int[][] PrimAlgorithm(int[][] graph)
    {
        int numVertices = graph.Length;
        int[] parent = new int[numVertices];
        int[] key = new int[numVertices];
        bool[] mstSet = new bool[numVertices];

        for (int i = 0; i < numVertices; i++)
        {
            key[i] = int.MaxValue;
            parent[i] = -1;
            mstSet[i] = false;
        }

        key[0] = 0; // Start with vertex 0 as the source

        for (int count = 0; count < numVertices - 1; count++)
        {
            int minKeyVertex = GetMinKeyVertex(key, mstSet);
            mstSet[minKeyVertex] = true;

            for (int vertex = 0; vertex < numVertices; vertex++)
            {
                if (graph[minKeyVertex][vertex] != 0 && !mstSet[vertex]
                {
                    parent[vertex] = minKeyVertex;
                    key[vertex] = graph[minKeyVertex][vertex];
```

Figure: Homework assignment implemented in c#

# ISO/TC 98/SC 2




Figure: ISO standardization committee

# Sample Engineering Problem



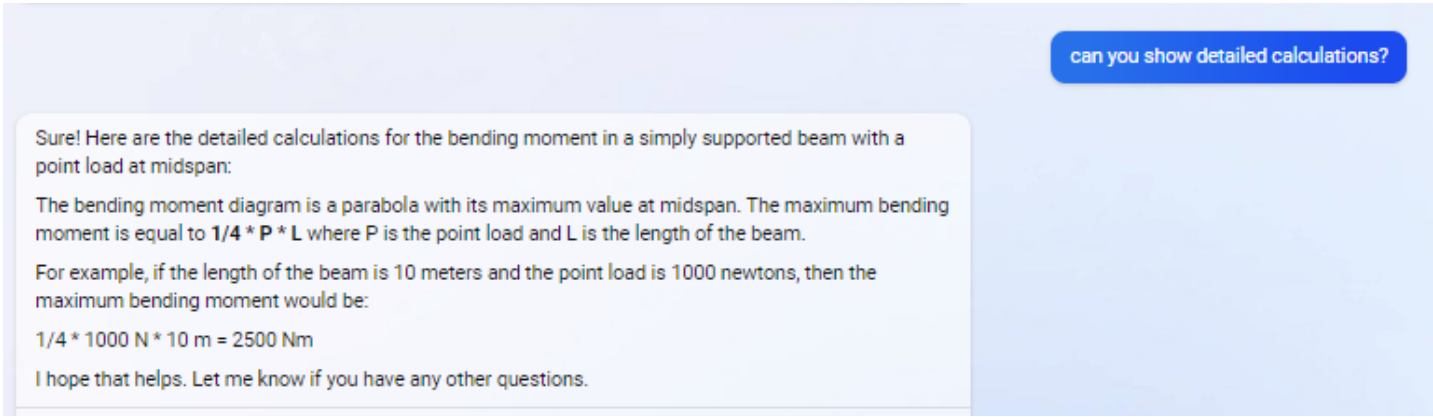



Figure: Bending moment in the beam

# Engineering Problems and Education



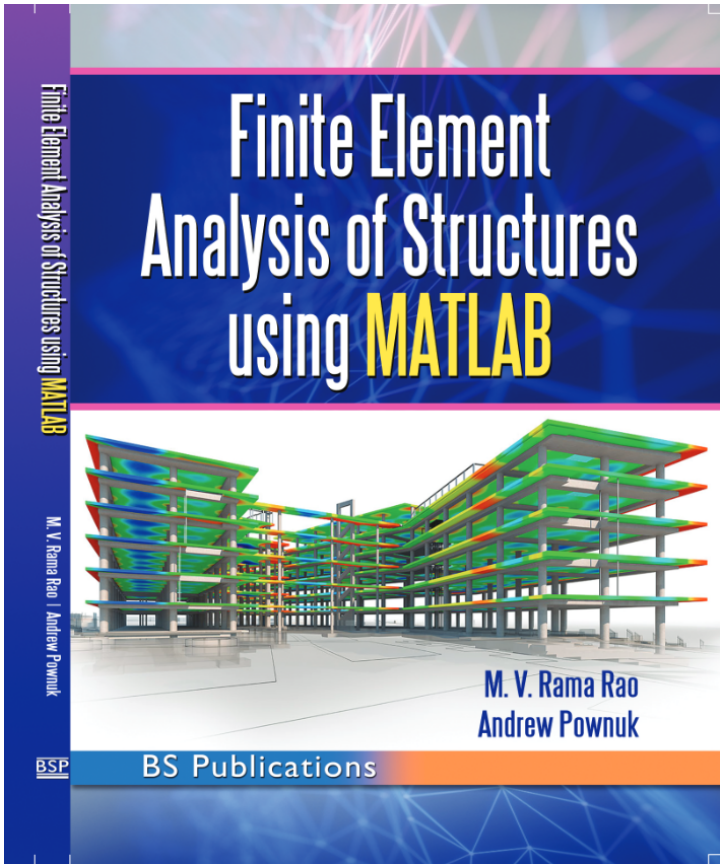



Figure: FEM book

# Computer Graphics - Stable Diffusion Algorithm



# Conclusions



- Online assignments are widely used in education.
- Online assignments are distributed programs which communicate with students and the server.
- By using Turing completed programming languages it is possible to describe wide range of scientific problems as online assignments.
- Large language models can be used as source of the solution of typical assignments.
- It is possible to fully automate some aspects of development of online assignments.

# Thank You