



โครงการงาน

Web Numerical Method โดยใช้ React

จัดทำโดย

นาย กรณ์มงคล ฤสิทธิ์พิชัย

รหัสนักศึกษา 6404062610391 ตอนเรียนที่ 1

รายงานนี้เป็นส่วนหนึ่งของวิชา 040613204

Object Oriented Programming

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคการศึกษาที่ 1 ปีการศึกษา 2565

สารบัญ

บทที่ 1 : บทนำ	1
บทที่ 2 : ส่วนการพัฒนา	3
บทที่ 3 : สรุป	9

บทที่ 1 : บทนำ

ที่มาและความสำคัญ

โครงการนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการจะนำเรื่องที่เรียนมาสร้างเป็น web React ผู้จัดทำ ได้สร้าง web นี้ขึ้น เพื่อฝึกฝนการเขียน website และ ฝึกวิธีการคิด วิเคราะห์ ปรับใช้ เนื้อหาที่เรียนมาใน วิชา Object Oriented Programming และ Numerical Method เข้าด้วยกัน โดยวิชา Object Oriented Programming จะใช้ในการทำโครงสร้าง และ ตกแต่ง web และ วิชา Numerical Method ใช้ในการเขียนวิธีการคำนวณ และหาผลลัพธ์ ดังนั้นการทำโครงการนี้จึงได้ประโยชน์มากมายและเป็นการบูรณาการ ทั้ง2วิชาเข้ากันได้อย่างลงตัว

ประโยชน์

1. ฝึกไหวพริบ ฝึกการเขียน web react
2. เพื่อฝึกกระบวนการคิดในสถานการณ์ต่าง ๆ
3. ฝึกสมาธิ
4. ฝึกการเขียนโปรแกรมแบบ Object Oriented Programming
5. ฝึกการบูรณาการ เรื่องที่เรียนเข้าด้วยกัน

ขอบเขตของโครงการ

โดยโครงการนี้ จะเขียน Web Application บน React โดยเนื้อหาที่จะทำ จะมีดังนี้

Root of equations

- Graphical Method
- Bisection Method
- False positions Method
- Newton Rapson Method
- One point iteration Method
- Secant Method

Linear equations

- Cramer Rule
- Gauss Elimination
- Gauss Jordan Elimination
- Inverse Matrix
- LU Decomposition
- Conjugate Gradient

Interpolations

- Newton's divided-differences
- Lagrange Interpolations
- Spline Interpolations

Regression

- Linear Regression
- Multiple Linear Regression
- Polynomial Regression

บทที่ 2 : ส่วนการพัฒนา

Constructor

ตัวอย่าง

- กำหนดค่าตัวแปรและ มีฟังก์ชันที่กำหนดพารามิเตอร์ เป็น class ในรูปแบบหนึ่ง

```
const Calbisection = (x1, xr) => {
  var xm, fXm, fXr, ea, scope;
  var iter = 0;
  var MAX = 50;
  const e = 0.000001;
  var obj={};
  do
  {
    xm = (x1+xr)/2.0;
    scope = {
      x:xr,
    }
    fXr = evaluate(Equation, scope)

    scope = {
      x:xm,
    }
    fXm = evaluate(Equation, scope)

    iter ++;
    if (fXm*fXr > 0)
    {
      ea = error(xr, xm);
      obj = {
        iteration:iter,
        X1:x1,
        Xm:xm,
        Xr:xr
      }
    }
  }
}
```

- ตัวแปร const มีการกำหนดค่าเริ่มต้นให้ตรง useState คล้ายกับ Default Constructor มี Mutator methods (setter)

```
//const [Data,setData] = useState([])
const [html, setHtml] = useState(null);
const [Equation,setEquation] = useState("13-(x^4)")
const [X,setX] = useState(0)
const [show,setShow] = useState(false)
const [XL,setXL] = useState(0)
const [XR,setXR] = useState(0)
```

- เป็นการโยนตัวแปร Object obj ซ้อน object

```
obj = {
  iteration: iter,
  Xl: xl,
  Xm: xm,
  Xr: xr
}
data.push(obj)
```

- เป็นการเรียก Function component ของแต่ละไฟล์มาแสดงที่หน้าเว็บ คล้ายๆการเรียก class

```
function App() {
  return (
    <div className="App">
      <BrowserRouter>
        <NavBar/>
        <Routes>
          <Route path="/Home" element={<Home/>} />
          <Route path="/Bisection" element={<Bisection/>} />
          <Route path="/False-Position" element={<FalsePosition/>} />
          <Route path="/One-Point" element={<OnePoint/>} />
          <Route path="/Newton-Raphson" element={<NewtonRap/>} />
          <Route path="/Secant" element={<Secant/>} />
          <Route path = "/Matrix" element={<Matrix/>}/>
        </Routes>
      </BrowserRouter>
    </div>
  );
}
```

อธิบายส่วนของโปรแกรม

ตัวอย่างเป็น code ของ Secant Methods

```
import { useState } from "react"
import { Button, Container, Form, Table } from "react-bootstrap";
import { evaluate } from 'mathjs'
import 'chart.js/auto'
import { Line } from "react-chartjs-2";

const Secant = () => {
  const print = () =>{
    console.log(data)
    setValueIter(data.map((x)=>x.iteration));
    setValueX0(data.map((x)=>x.x0));
    setValueX1(data.map((x)=>x.x1));
    return [
      <Container>
        <Table striped bordered hover variant="dark">
          <thead>
            <tr>
              <th width="10%">Iteration</th>
              <th width="30%">X0</th>
              <th width="30%">X1</th>
            </tr>
          </thead>
          <tbody>
            {data.map((element, index)=>{
              return (
                <tr key={index}>
                  <td>{element.iteration}</td>
                  <td>{element.x0}</td>
                  <td>{element.x1}</td>
                </tr>
              )
            })}
          </tbody>
        </Table>
      </Container>
    ]
  }
}
```

- Import Component ที่ต้องใช้ สร้าง function component ขึ้นมา และเขียน export default function component ในส่วนของหน้านี้จะเป็นการ สร้าง arrow function print() เพื่อนำข้อมูลที่ได้จากการคำนวณมาแสดงผลในรูปแบบตาราง โดยใช้ Table


```

const error = (xold, xnew) => Math.abs((xnew-xold)/xnew)*100;

const CalSecant = (X0,X1) => {
  var X2,fx0,fx1,ea,score;
  var iter = 0;
  var MAX = 50;
  const e = 0.00001;
  var obj={};

  do
  {
    scope = {
      x:X0
    }
    fx0 = evaluate(Equation,score)
    scope = {
      x:X1
    }
    fx1 = evaluate(Equation,score)
    X2 = (X1-fx1*(X0-X1)/(fx0-fx1))
    X0 = X1;
    X1 = X2;
    iter++;
    ea = error(X0,X1)
    obj = {
      iteration:iter,
      x0:X0,
      x1:X1
    }
    data.push(obj)
  }while(ea>e && iter<MAX)
  setX(X2)
}

```

- วิธีการทางคณิตศาสตร์ Secant Methods คำตอบ ในแต่ละ iterations จะถูกเก็บใน array data เมื่อหาคำตอบได้แล้ว ก็จะเรียกใช้ Method setX และค่าใน useState ของ X จะถูกเปลี่ยนค่า และนำค่าที่ได้ไปแสดงผลหน้าเว็บ

```

const data = [];
const [valueIter, setValueIter] = useState([]);
const [valueX0, setValueX0] = useState([]);
const [valueX1, setValueX1] = useState([]);
const state = {
  labels: valueIter,
  datasets: [
    {
      label: 'x0',
      fill: false,
      lineTension: 0.5,
      backgroundColor: 'white',
      borderColor: 'blue',
      borderWidth: 2,
      data: valueX0
    },
    {
      label: 'x1',
      fill: false,
      lineTension: 0.5,
      backgroundColor: 'white',
      borderColor: 'red',
      borderWidth: 2,
      data: valueX1
    }
  ]
}

//const [Data, setData] = useState([])
const [html, setHtml] = useState(null);
const [Equation, setEquation] = useState("(x^2)-7")
const [X, setX] = useState(0);
const [x0, setX0] = useState(0);
const [x1, setX1] = useState(0);
const [show, setShow] = useState(false);


const inputEquation = (event) =>{
  console.log(event.target.value)
  setEquation(event.target.value)
}

const inputX0 = (event) =>{
  console.log(event.target.value)
  setX0(event.target.value)
}

```

- ในหน้านี้เป็นการสร้างตัวแปรที่ใช้ useState และ setState อีกทั้งยังสร้างตัวแปรชนิด const ไว้เก็บข้อมูลของกราฟ ในส่วนสุดท้ายเป็นการดักจับ Event ตรง input type เวลา user กรอกข้อมูล แล้วนำไป setState

ตัวอย่างหน้าเว็บ

 Kornmongkhon Ruesingpisch Project
 Home Root of Equation Linear Algebraic Equation Interpolation And Extrapolation

Secant Method


Input $f(x)$

Input X_0

Input X_1

Calculate

-กรอกค่า และกดปุ่ม Calculate

 Kornmongkhon Ruesingpisch Project
 Home Root of Equation Linear Algebraic Equation Interpolation And Extrapolation

Secant Method

Input $f(x)$

Input X_0

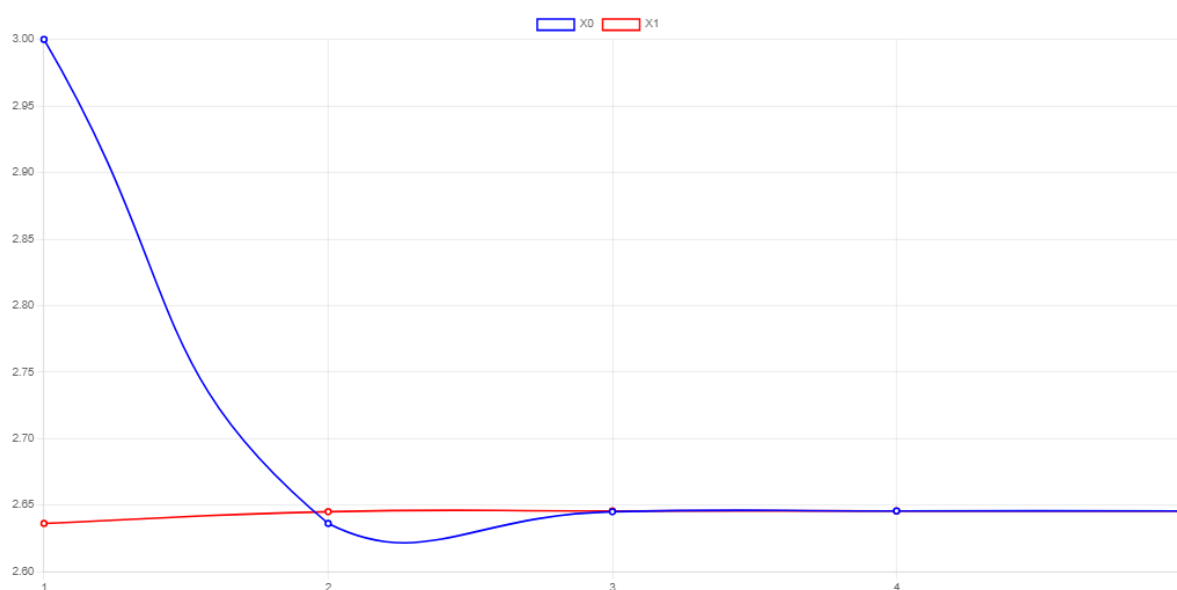
Input X_1

Calculate

Answer = 2.645751311064590716

Iteration	X_0	X_1
1	3	2.6363636363636362
2	2.6363636363636362	2.6451612903225805
3	2.6451612903225805	2.645752359800111
4	2.645752359800111	2.64575131094764
5	2.64575131094764	2.6457513110645907

-จะแสดงกราฟที่แสดงคำตอบที่ได้จากในตาราง และการ plot เส้นจะเทียบคำตอบกับ Iteration



บทที่ 3 : สรุป

ปัญหาที่พบบ่อยระหว่างการพัฒนา

- ปัญหาในการใช้ useState กับ setState
- ปัญหาในการเชื่อมต่อกราฟเวลา plot จุด ของ scatter plot
- ปัญหาเวลาตอนแสดงผลกราฟ กดปุ่ม 1 ที่แล้วกราฟไม่แสดงเนื่องจาก React ประมวลผลไวเกินไป