

# Oblig 1 Matte 3

Adam Aske

24. januar 2022

## Innhold

<b>1</b>	<b>Del 1</b>	<b>2</b>
<b>2</b>	<b>Lese og skrive til fil</b>	<b>2</b>

## 1 Del 1

Jeg har valgt funksjonen;  $f(x, y) = \sin(\text{PI} \cdot x) \cdot \sin(\text{PI} \cdot y)$ . Omerådet  $0 < x < 1$ ,  $0 < y < 3$  og steg = 0.2. Funksjonen tar inn en array og en størrelse. Først blir arrayen fylt med tilfeldige tall.

Listing 1: trianglesurface.cpp

```
//Create triangle
float xmin=0.0f, xmax=1.0f, ymin=0.0f, ymax=3.0f, h=0.2f;
for (auto x=xmin; x<xmax; x+=h)
{
    for (auto y=ymin; y<ymax; y+=h)
    {
        float z = func(x, y);
        mVertices.push_back(Vertex{x,y,z,x,y,z});
        z = func(x+h, y);
        mVertices.push_back(Vertex{x+h,y,z,x,y,z});
        z = func(x, y+h);
        mVertices.push_back(Vertex{x,y+h,z,x,y,z});
        mVertices.push_back(Vertex{x,y+h,z,x,y,z});
        z = func(x+h, y);
        mVertices.push_back(Vertex{x+h,y,z,x,y,z});
        z = func(x+h, y+h);
        mVertices.push_back(Vertex{x+h,y+h,z,x,y,z});
    }
}
```

Listing 2: trianglesurface.hh

```
static float func(float x, float y){
    //Matte oblig funksjon
    return sin(PI*x)*sin(PI*y);
}
```

## 2 Lese og skrive til fil

Listing 3: trianglesurface.cpp

```
void TriangleSurface::readFile(std::string fileName) {
    std::ifstream inn;
    inn.open(fileName.c_str());
    if (inn.is_open())
    {
        int n;
        Vertex vertex;
        inn >> n;
        mVertices.reserve(n);
        for (int i=0; i<n; i++) {
            inn >> vertex;
            mVertices.push_back(vertex);
        }
        inn.close();
    }
}

void TriangleSurface::writeFile(std::string fileName){
    std::ofstream wF;
    wF.open(fileName.c_str())
    if(wF.is_open())
```

```

    {
        wF << mVertices.size() << "\n";
        for (int i = 0; i < mVertices.size(); i++)
        {
            wF << mVertices[i] << "\n";
        }
    }
    else
    {
        std::cout << "Failed to write to file.\n";
    }
    wF.close();
}

```

### 3 Del 2

#### 4 B)

For å regne integralet numerisk lagde jeg en funksjon i trianglesurface.cpp og skriver resultatene til en fil. Funksjoner gjør det 4 ganger og halverer steg lengden for hver iterasjon. Resultatene blir lagret i Numerisk.txt.

Listing 4: trianglesurface.cpp

```

void TriangleSurface::CalculateNumerical(){
    std::ofstream file;
    file.open("Numerisk.txt");
    if(file.is_open())
    {
        float xmin= -1.0f, xmax = 1.0f, ymin = 0.0f, ymax = 1.0f, h = 0.25f, result = 0.0f;
        for(int i = 0; i < 4; i++)
        {
            for(auto x = xmin; x < xmax; x+=h)
            {
                for(auto y = ymin; y < ymax; y+= h)
                {
                    float z = func(x, y);
                    result += z;
                }
            }
            h = h / 2;
            file << result << "\n";
        }
    }
    else
    {
        std::cout << "Failed to write to file.\n";
    }
    file.close();
}

```

Resultatene ble: h1 = 1.78814e-07 h2 = -4.76837e-07 h3 = -1.02818e-06 h4 = 4.12203e-06