20211046
Hyeonjang An

Homework 1

EC6301
Numerical Opimization
September 11, 2021

Problem 1

Implement the method of bisection , Newtons's, secant, regular falsi.

1. Optimizing Method Class

```
class Method {
public:
  Method(std::function<float(const float&)> f):function(f){};

  // optimization methods
  float bisection(float start, float end);
  float newtons(float x);
  float secant(float x1, float x0);
  float regular_falsi(float start, float end);

protected:
  // target function as member
  std::function<float(const float&)> function;
};
```

2. Bisection method

method 1: bisection

```
float Method::bisection(float start, float end) {
  assert( function(start)*function(end)<0 );

  auto midpoint = (start + end)/2.f;

  if(function(midpoint)==0 || end-start<MIN)
    return midpoint;

  if(function(midpoint)*function(start)<0)
    midpoint = bisection(start, midpoint);
  else
    midpoint = bisection(midpoint, end);

  return midpoint;
}
```

Homework 1

3. Newtons's method

method 2: Newton's

```
float Method::newtons(float x0) {
  // approximattion of derivative lambda function
  auto d =
  [](std::function<float(const float&)> func, float x, float eps=1e-6)
  {
      return (func(x+eps) - func(x))/eps;
  };

  float x1 = x0;
  while(function(x1)>0.f) {
      float t = x1;
      x1 = t - function(t)/d(function, t);
  }
  return x1;
}
```

4. Secant method

method 3: secant

```
// Two point approximation method
float Method::secant(float x1, float x0) {
  // no matter which is bigger
  x1 = std::min(x1, x0);
  x0 = std::max(x1, x0);

  float x2 = MAX; // initial next point
  while(function(x2)>0.f) {
    x2 = x1 - ((x1-x0)/(function(x1)-function(x0))) * function(x1);

    x0 = x1;
    x1 = x2;
  }
  return x2;
}
```

Homework 1

5. Regular-falsi method

method 4: regular false

```
float Method::regular_falsi(float start, float end) {
  // secant method lambda
  auto sec = [](std::function<float(const float&)> func, float x1,
     float x0)
  {
      return x1 - ((x1-x0)/(func(x1)-func(x0))) * func(x1);
  };

  assert( function(start)*function(end)<0 );

  // new x-axis intersection point
  float x = sec(function, start, end);

  if(function(x)==0 || end-start<MIN)
      return x;

  // do recursivly until the end
  if(function(start)*function(x) < 0)
      x = regular_falsi(start, x);
  else
      x = regular_falsi(x, end);

  return x;
}
```

Homework 1

Problem 2

Discuss thier comparative performance for at least four different problems you generate.

1. Target functions

$$X(m,n) = \left\{ \begin{array}{ll} x(n), & \text{for } 0 \leq n \leq 1 \\ \frac{x(n-1)}{2}, & \text{for } 0 \leq n \leq 1 \\ \log_2 \lceil n \rceil & \text{for } 0 \leq n \leq 1 \end{array} \right\} = xy$$

2. The derivative of target functions

3. Peformance comparision