Computation of integrals using Simpson's and Newton's rules

Szymon Górski

1 Assignment

1.1 Content

The goal of the project was to solve the following task:

> Write a computer program to implement Simpson's and Newton's ($\frac{3}{8}$) rules for computing the integral $\int_a^b f(x)dx$, where
>
> $$f(x) = b_0 + \sum_{k=1}^{n} (a_k \sin(kx) + b_k \cos(kx)).$$
>
> Compare the results.
>
> Use the Goertzel method.

I denote the numerical value of this integral as S(f), therefore $S_{\text{approximation method}} \approx \int_a^b f(x)dx$.

1.2 Newton–Cotes formulas

Newton–Cotes formulas are formulas used in numerical integration. They base on calculations of values of a given function within equally distanced points, called nodes or knots.

1.2.1 The Simpson's Rule

The Simson's Rule bases on 3 nodes at $a, \frac{a+b}{2}, b$. The approximation of the integral $S_S(f)$ and its error $E_S(f)$ for function $f(x)$ are given by the following formulas:

$$S_S(f) = \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right], \tag{1}$$

$$E_S(f) = \frac{1}{90} * (\frac{b-a}{2})^5 * f^{(4)}(\xi) = \frac{1}{2880} * (b-a)^5 f^{(4)}(\xi), \tag{2}$$

where $\xi \in (a, b)$.

1.2.2 The Newton's Rule

The Newton's Rule is also known as the Simpson's $\frac{3}{8}$ Rule. With 4 nodes at $a, \frac{2a+b}{3}, \frac{a+2b}{3}, b$, the approximation of the integral $S_N(f)$ and its error $E_N(f)$ for function $f(x)$ are given by the following formulas:

$$S_N(f) = \frac{b-a}{8}\left[f(a) + 3f(\frac{2a+b}{3}) + 3f(\frac{a+2b}{3}) + f(b)\right], \tag{3}$$

$$E_N(f) = \frac{3}{80} * (\frac{b-a}{3})^5 * f^{(4)}(\xi) = \frac{1}{6480} * (b-a)^5 f^{(4)}(\xi), \tag{4}$$

where $\xi \in (a, b)$.

1.3 The Goertzel's algorithm

1.3.1 Usage

The Goertzel's algorithm is used to evaluate polynomials with complex arguments. For a polynomial of a form

$$f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n, \tag{5}$$

where $a_k \in R$. We divide it by $(x - z)(x - \bar{z})$, which results in

$$f(x) = c_0 + c_1 x + (x - z)(x - \bar{z}) \sum_{k=2}^{n} c_k x^{k-2}, \tag{6}$$

where $c_k \in R$. $(x - z)(x - \bar{z}) = x^2 - px - q$, where $p = 2 * \Re(z), q = -|z|^2$ and $p, q \in R$. For $f(z)$ we get

$$f(z) = c_0 + c_1 z + 0 * \sum_{k=2}^{n} b_k z^{k-2} = c_0 + c_1 z. \tag{7}$$

It is enough to calculate c_0 and c_1 to obtain $f(z)$, using the following algorithm.

1.3.2 Content

- $p = 2 * \Re(z)$
  $q = -(\Re(z)^2 + \Im(z)^2)$
- $c_{n+1} = 0$
  $c_n = a_n$

for $k = n - 1, ..., 1$

- $c_k = a_k + pc_{k+1} + qc_{k+2}$
- $\Re(f(z)) = a_0 + \Re(z) * c_1 + qb_2$
  $\Im(f(z)) = \Im(z) * c_1$

2 Solutions

I prepared the solutions in MATLAB.

2.1 Calculations

I used the Goertzel's algorithm to calculate each value of function f. It is possible, because one can expand a summation of form $\sum_{k=0}^{n} a_k * \cos(kt)$ to a polynomial, basing on an observation that $z^k = |z|^k\left(\cos(kt) + i\ \sin(\mathrm{kt})\right)$, where $z \in C$.

$$
\begin{gathered}
f(x)dx = b_0 + \Im\left(\sum_{k=1}^{n} b_k\ \sin(kx)\right) + \Re\left(\sum_{k=1}^{n} b_k\ \cos(kx)\right) = \\
= b_0 + \Im\left(\sum_{k=1}^{n} a_k(\cos(kx) + i\ \sin(kx))\right) + \Re\left(\sum_{k=1}^{n} b_k(\cos(kx) + i\ \sin(kx))\right) = \\
= b_0 + \Im\left(a_0 + a_1 z_a + ... + a_n z_a^n\right) + \Re\left(b_0 + b_1 z_b + ... + b_n z_b^n\right) \rightarrow \\
\xrightarrow[\text{Goertzel's algorithm}]{\text{applying the}} b_0 + \Im(u_1 + iv_1) + \Re(u_2 + iv_2) = b_0 + v_1 + u_2
\end{gathered} \tag{8}
$$

2.2 Implementation of the Goertzel's algorithm

```
function [u, v] = GoertzelsAlgorithm(x, A0, A)
    A_size = size(A,2);
    N = A_size;
    z_x = cos(x);
    z_y = sin(x);
    p = 2*z_x;
    q = -(z_x^2 + z_y^2);
    C = zeros(1, N+1);
    C(N) = A(N);
    for i = N-1:-1:1
        C(i) = A(i) + p*C(i+1) + q*C(i+2);
    end
    u = A0 + z_x*C(1) + q*C(2);
    v = z_y*C(1);
end
```

2.3 Implementation of the Newton-Cotes formulas

Implementation of the Simpson's Rule:

```
function [S] = SimpsonsRule(a, b, A, B0, B)
    [U1,~] = GoertzelsAlgorithm(a, B0, B);
    [~,V1] = GoertzelsAlgorithm(a, 0, A);
    [U2,~] = GoertzelsAlgorithm((a+b)/2, B0, B);
    [~,V2] = GoertzelsAlgorithm((a+b)/2, 0, A);
    [U3,~] = GoertzelsAlgorithm(b, B0, B);
    [~,V3] = GoertzelsAlgorithm(b, 0, A);
S = (b-a)*(U1+V1 + 4*(U2+V2) + U3+V3)/6;
end
```

Implementation of the Newton's Rule:

```
function [S] = NewtonsRule(a, b, A, B0, B)
    [U1,~] = GoertzelsAlgorithm(a, B0, B);
    [~,V1] = GoertzelsAlgorithm(a, 0, A);
    [U2,~] = GoertzelsAlgorithm(((2*a+b)/3), B0, B);
    [~,V2] = GoertzelsAlgorithm(((2*a+b)/3), 0, A);
    [U3,~] = GoertzelsAlgorithm(((a+2*b)/3), B0, B);
    [~,V3] = GoertzelsAlgorithm(((a+2*b)/3), 0, A);
    [U4,~] = GoertzelsAlgorithm(b, B0, B);
    [~,V4] = GoertzelsAlgorithm(b, 0, A);
    S = (b-a)*(U1+V1 + 3*(U2+V2) + 3*(U3+V3) + U4+V4)/8;
end
```

3 Accuracy

All the comparisons within the code are preformed by the `ExactSolution` function.

3.1 Inaccuracies by the Newton-Cotes formulas

Each Newton-Cotes formula provides an error term, which contains a coefficient of form $f^{(2n)}(\xi)$, where $\xi \in (a, b)$. However, there is no common rule how to obtain ξ. However, the maximum error which can occur is bounded by:

$$E_S(f) = \frac{1}{2880} * (b-a)^5 \max_{\xi \in [a,b]} |f^{(4)}(\xi)|, \tag{9}$$

$$E_N(f) = \frac{1}{6480} * (b-a)^5 \max_{\xi \in [a,b]} |f^{(4)}(\xi)|. \tag{10}$$

The terms are used to show if the order of the error satisfies its user. Dividing $E_S(f)$ and $E_N(f)$ one by the other, we can see that statistically the Newton's Rule is $\frac{E_S(f)}{E_N(f)} = 2.25$ times as accurate as the Simpson's Rule.

3.2 Exact calculations of the integral

One can also calculate the integral explicitly and check the difference. There are two main arguments for using this method:

1. they might be considered easier as it does not demand to find ξ,

2. it includes the errors that are caused by the hardware and software computational inaccuracy.

3.2.1 Calculations

$$\begin{aligned}\int_a^b f(x)dx = \int_a^b [b_0 + a_1\ \sin(x) + b_1\ \cos(x) + ... + a_n\ \sin(nx) + b_n\ \sin(nx)] = \\ = \int_a^b b_0 dx + \int_a^b a_1\ \sin(x)dx + \int_a^b b_1\ \cos(x)dx + ... + \\ + \int_a^b a_n\ \sin(nx)dx + \int_a^b b_n\ \sin(nx)dx\end{aligned} \tag{11}$$

$$\int_a^b \sin(kx)dx = \frac{\cos(ak) - \cos(bk)}{k} \tag{12}$$

$$\int_a^b \cos(kx)dx = \frac{\sin(bk) - \sin(ak)}{k} \tag{13}$$

$$\int_a^b f(x)dx = (b-a)b_0 + \sum_{k=1}^{n} \frac{1}{k}(a_k\ \sin(bk) + b_k\ \cos(ak) - a_k\ \sin(ak) - b_k\ \cos(bk)) \tag{14}$$

3.2.2 Implementation

```
function [Exact] = ExactSolution(a, b, A, B0, B)
    A_size = size(A,2);
    N = A_size;
    Exact = B0*(b-a);
    Function = @(i,x) B(i)*cos(i*x)+A(i)*sin(i*x);
    for i=1:N
        Exact = Exact + integral(@(x) Function(i,x), a, b);
    end
end
```

4 Exceptions

4.1 Different number of arguments

The $f(x)$ formula given in Sect. 1.1 uses one summation sign only, which iterates through all a_k and b_k arguments. Therefore, number of a and b arguments must equal each other. By the number of these arguments the program reads n, used in iterations inside the Goertzel's algorithm. Providing A and B arrays with different number of arguments results with error "`Number of a_(k) and b_(k) arguments differ from each other!`".

4.2 Case $a = b$

In such a case, the Newton-Cotes formulas correctly approximate values of the integral to 0. This is because one of coefficients in each Newton-Cotes formula is $b - a$, which in this case always equals 0. Therefore, an approximation of integrals with this property can be done without calculating values of the function. However, in order not to change the computation method given in the task, this observation was not implemented in `SimpsonsRule` nor `NewtonsRule`. It was included in `CompareResults` only, to emphasise this observation.