Technion – Israel Institute of Technology

HW7

Numerical Methods

019003

Alon Spinner	305184335	alonspinner@gmail.com
Oren Elmakis	311265516	orenelmakis@gmail.com

January 24, 2022

Question 1: Gauss-Quadrature with 3 sampled points

Given $M = 3$ points to sample from, we will construct a polynomial of degree $M + 2$.

$$f(x) = \sum_{i=0}^{M+2} a_i x^i$$

Given an integral in bracket $[a, b]$, one can introduce a variable change such that after, values in it will range from $[-1,1]$

$$z = \frac{2x - a - b}{b - a}$$

$$x = \frac{b-a}{2} z + \frac{b+a}{2}$$

As such, we will only practice on the integral in bracket $[-1,1]$.

We assume the integral could be written as a sum of weights:

$$I = \int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{M} W_i f(x_i)$$

Integrating over the polynomial (left side):

$$I = \int_{-1}^{1} f(x)dx = 2a_0 + \frac{2}{3}a_2 + \frac{2}{5}a_4$$

Substituting the polynomial values in the sum (right side):

$$I \approx W_1 f(x_1) + W_2 f(x_2) + W_3 f(x_3) =$$

$$= a_0(W_1 + W_2 + W_3) + a_1(W_1x_1 + W_2x_2 + W_3x_3) + a_2(W_1x_1^2 + W_3x_2^2 + W_3x_3^2) \dots =$$

$$= \sum_{i=0}^{5} a_i [x_1^i \quad x_2^i \quad x_3^i] \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$$

Comparing left side evaluation to the right-side evaluation:

$$2a_0 + \frac{2}{3}a_2 + \frac{2}{5}a_4 = \sum_{i=0}^{5} a_i [x_1^i \quad x_2^i \quad x_3^i] \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$$

Comparing Coefficients:

$$2 = W_1 + W_2 + W_3$$

$$0 = W_1x_1 + W_2x_2 + W_3x_3$$

$$\frac{2}{3} = W_1x_1^2 + W_2x_2^2 + W_3x_3^2$$

$$0 = W_1x_1^3 + W_2x_2^3 + W_3x_3^3$$

$$\frac{2}{5} = W_1 x_1^4 + W_2 x_2^4 + W_3 x_3^4$$

$$0 = W_1 x_1^5 + W_2 x_2^5 + W_3 x_3^5$$

We have a system of equations – 6 equations and 6 variables $\{x_i, W_I\}$.

Using MATLAB's symbolic solver to write a general script for M points.
The results for $M = 3$ are as follows:

Table 1 GQ coefficients for M=3

$W_1 = \frac{5}{9}$	$x_1 = -\frac{\sqrt{15}}{5}$
$W_2 = \frac{8}{9}$	$x_2 = 0$
$W_3 = \frac{5}{9}$	$x_3 = +\frac{\sqrt{15}}{5}$

Question 2: Solving $\int_0^2 x^2 \tan^{-1}(x)dx$

In this question we will consider the bracket $[a, b]$, computing the integral over $f(x)$ with N points.

$$[a, b] = [0,2]$$

$$N = 3$$

$$f(x) = x^2 \tan^{-1}(x)$$

a. Analytical Solution:

We used MATLAB to solve the symbolic integral:

$$I = \int_0^2 x^2 \tan^{-1}(x)\, dx = \frac{1}{6}(2x^3 \tan^{-1}(x) - x^2 + \ln(x^2 + 1))|_0^2 \ = 2.5440$$

b. Closed Newton-Cotes with 3 points

Closed Newton-Cotes is the Simpson $\frac{1}{3}$ method

$$h = \frac{b - a}{N - 1} = 1$$

$$x = (a: h: b) = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$W = \frac{h}{3}[1,4,1]$$

$$I = W \cdot f(x) = 2.5234$$

c. Open Newton-Cotes with 3 points

$$h = \frac{b-a}{N+1} = \frac{1}{2}$$

$$x = (a{:}\,h{:}\,b) = \begin{bmatrix} 0.5 \\ 1 \\ 1.5 \end{bmatrix}$$

$$W = \frac{h}{3}[2,-1,2]$$

$$I = W \cdot f(x) = 2.5793$$

d. Gauss-Quadrature with 3 points

From Question 1 we know that:

$$W = \frac{1}{9}[5,8,5]$$

$$z = \left[-\frac{\sqrt{15}}{5}, 0, \frac{\sqrt{15}}{5}\right]^T$$

Computing x from z:

$$x = \frac{b-a}{2} z + \frac{b+a}{2}$$

To avoid losing accuracy to digit precession, we use the long format of double for presentation.

$$x = \begin{bmatrix} 0.225403330758517 \\ 1 \\ 1.774596669241483 \end{bmatrix}$$

$$I = W \cdot f(x) = 2.554787510636612$$

e. Comparing the methods

Comparing the results, we produced the following table:

Table 2 Method Comparison

	I	Error	h	x			f(x)		
Analytical	2.554	0	0	0	0	0	0	0	0
NC-closed	2.5234	-0.030574	1	0	1	2	0	0.7854	4.4286
NC-open	2.5793	0.025362	0.5	0.5	1	1.5	0.11591	0.7854	2.2113
GQ	2.5548	0.00081794	NaN	0.2254	1	1.7746	0.011264	0.7854	3.3307

The first thing that's visible right off the bat, is that GQ has the smallest error by a large margin, following NC-open

$$\left\|\frac{Error(NC-open)}{Error(GQ)}\right\| = 31.007$$

$$\left\|\frac{Error(NC-open)}{Error(NC-closed)}\right\| = 1.2055$$

Having NC-open provide better result than NC-closed is not assured, but also not surprising, as both methods have their errors proportional to h^5 (assuming 3 points), but the step size in NC-open is smaller.

Additionally, we have seen in the lectures that the GQ method can integrate a polynomial of degree $2M - 1$ exactly, while the NC methods can do the same for a polynomial of degree $M - 1$. As such, it is of no surprise, that it outperforms them.

We decided to plot the function and the integration points for further insight, presenting the figure below:

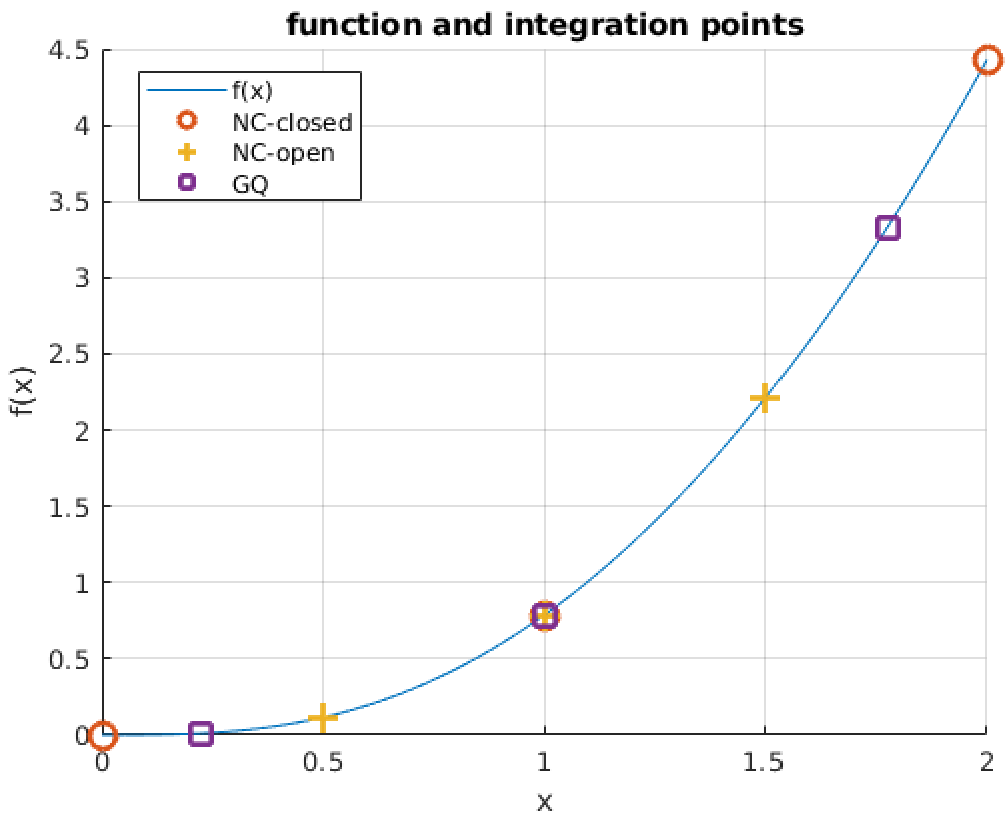

Figure 1 Method Comparison on Axes

The integration bracket, the function's derivative is monotonically increasing. As such, having well placed integration points near the higher end of the bracket will result in a better integration estimation.
Unfortunately, we weren't able to obtain any further intuition from the figure.

One final note:
We decided against evaluating and comparing the computation time for each method, as they are all $O(n)$. In the lecture we were presented with the fact that the GQ method is more efficient.

Question 3: Adaptive Gauss Quadrature

We were asked to implement the function *AdaptQuad(fun,a,b,n,epsilon)* which returns the integral approximation *I*, of function *fun* (sometimes denoted *'f'*), over bracket *[a,b]* given *n*, number of sample points per section, and *epsilon*, the error tolerance for section subdivision.

Use *epsilon* = 10^{-6}

In “Numerical Methods for Engineers” 6th addition, we are given the error term for the GQ method in equation 22.32. It shows as it appears below:

$$E_t = \frac{2^{2n+3}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3} f^{(2n+2)}(\xi)$$

Where $\xi \in [a,b]$

We could think of two methods to implementing an *adaptive gauss quadrature*.

1. Use the error term to develop relationship between the error of $GQ(f,a,b,n)$ and $GQ(f,a,b,n+1)$, increasing n until the division between them will be less than *epsilon*. In this method, one could solve for a sufficient n without integrating once.

   We have let MATLAB do the job and found the following relationship <u>under the assumption that the derivative term in the error remains similar and can be cancelled</u>

$$\frac{E_t(n+1)}{E_t(n)} = \frac{n+2}{2(2n+3)^2(2n+5)}$$

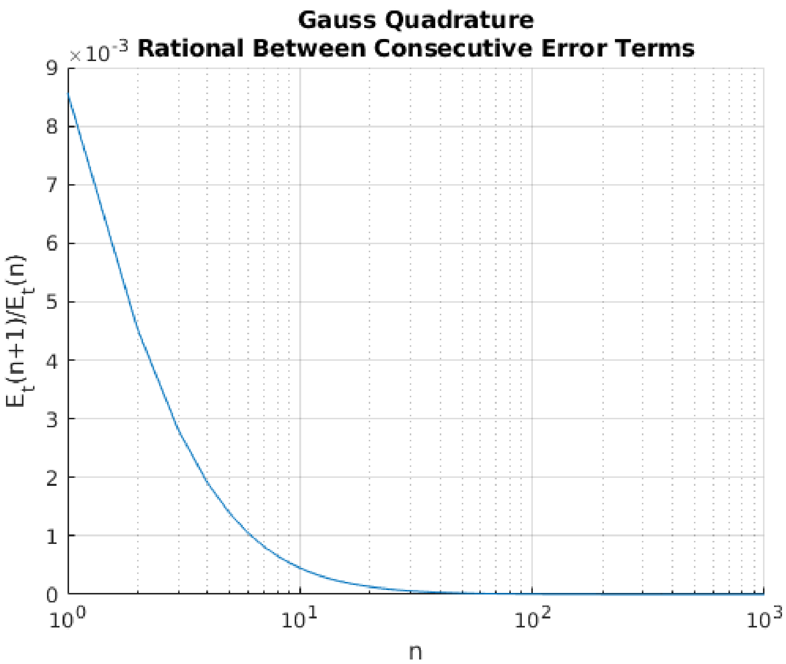

Figure 2 GQ rational between consecutive error terms

   Equating the rational to *epsilon* and solving numerically via *vpasolve* provides the answer of $n = 249$.

2. Use the same halving technique used in the common NC adaptive quadrature, which is developed using *Boole’s rule*, applying GQ integration instead of NC integration within it. As we have no Boole’s rule for GQ, our best integral estimate will be

$$\sum_{i=0}^{k} GQ(f,a(k),b(k),n)$$

k is the number of brackets after all subdivisions have been performed.

We stop subdividing in bracket $[a, b]$ when the relative error is smaller than *epsilon*

$$Stop\ Subdividing\colon \frac{GQ(f, a, b, n) - (GQ(f, a, m, n) + GQ(f, m, b, n))}{GQ(f, a, b, n)} < \epsilon$$

We decided to implement method 2, and provide the pseudo-code used as a reference from "Numerical Methods for Engineers" 6th addition figure 22.5 below.

We also changed the header of the function adding an optional argument $I_1 = GQ(f, a_1, b_1, n)$ which makes our recursive function more efficient when it must test whether to subdivide or not.

Table 3 Adaptive Newton Cotes from 'Numerical Methods for Engineers'

```
FUNCTION quadapt(a, b)            (main calling function)
tol = 0.000001
c = (a + b)/2                            (initialization)
fa = f(a)
fc = f(c)
fb = f(b)
quadapt = qstep(a, b, tol, fa, fc, fb)
END quadapt

FUNCTION qstep(a, b, tol, fa, fc, fb) (recursive function)
h1 = b − a
h2 = h1/2
c = (a + b)/2
fd = f((a + c)/2)
fe = f((c + b)/2)
I1 = h1/6 * (fa + 4 * fc + fb)        (Simpson's 1/3 rule)
I2 = h2/6 * (fa + 4 * fd + 2 * fc + 4 * fe + fb)
IF |I2 − I1| ≤ tol THEN     (terminate after Boole's rule)
  I = I2 + (I2 − I1)/15
ELSE                            (recursive calls if needed)
  Ia = qstep(a, c, tol, fa, fd, fc)
  Ib = qstep(c, b, tol, fc, fe, fb)
  I = Ia + Ib
END IF
qstep = I
END qstep
```

a. Solving $\int_0^{\frac{\pi}{4}} e^{2x} \sin(2x)\, dx$

The figures below plot computation time and absolute error terms for different choices of n. For each n we computed the integral estimate with our adaptive GQ 10 times for a better estimate of computation time via averaging. The integral estimate itself is deterministic.

We would note that the MATLAB's integral is not a function of n, and was also computed 10 times for an average estimate, with the same relative error *epsilon*.

The absolute error from I_{true} was computed using MATLAB's symbolic solver.

We started from $n = 2$ as using only one integration point is inadequate for any method. After-all, Integration solves for area given length.

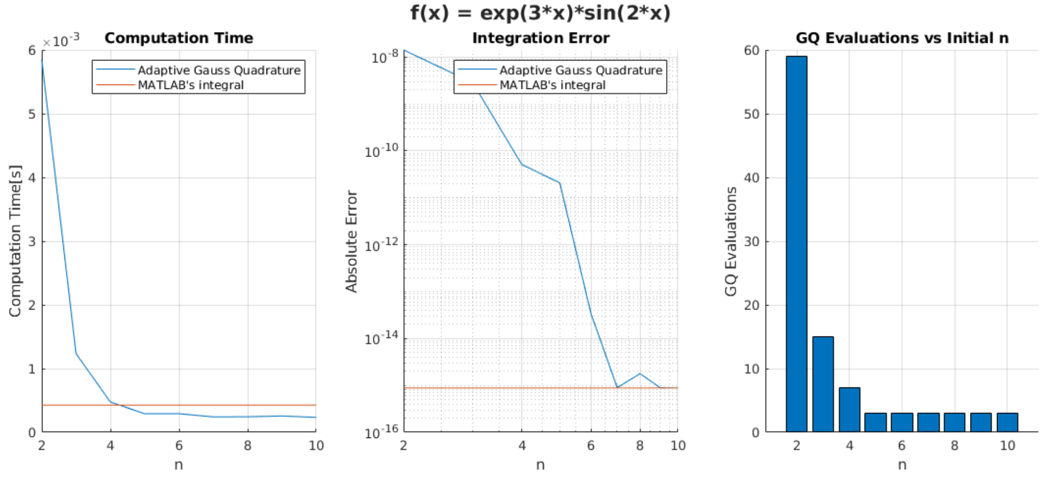

Figure 3 Computation Time and Error analysis for f(x)= exp(3x)sin(2x)

First thing to notice is that the time computation decreases as we increase n, which works well with our recursive method, as a lower initial n will require our algorithm to create more subdivisions.

Second thing is, that after $n = 5$, the GQ evaluations keep at the value 3, which means that rational error term was small enough that we exited after the first function call.
Additionally, we saw that for a reasonable n number we reach the same computation error as MATLAB's method, for a cheap computational cost.

It is worth mentioning that if we increase n higher and higher, then our algorithm will produce the same integral approximation, but for a much higher computational price, stemming from the nonlinear optimization problem that is solved to find the GQ coefficients, as for $n > 5$ there are always only 3 GQ evaluations.
This can be seen in the figure below:

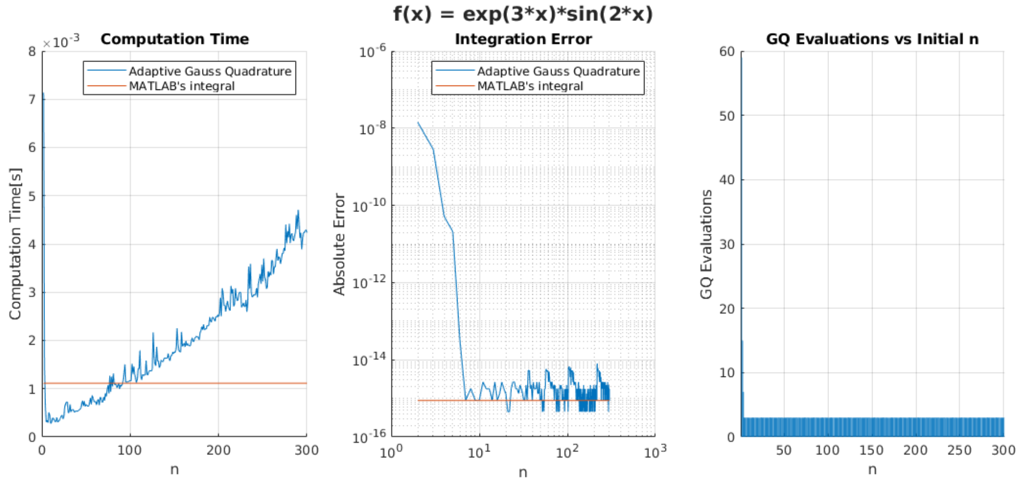

Figure 4 f(x)= exp(3x)sin(2x) analysis for high n numbers

b. Solving $\int_0^{\pi} x\sin(x^2)dx$

We used the exact same computation and averaging technique as in segment (a) of this question.

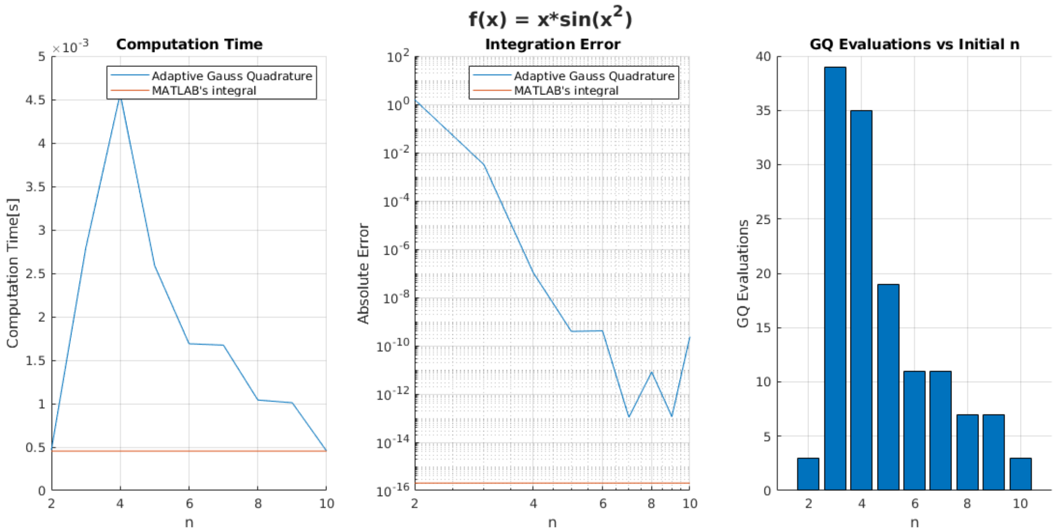

Figure 5 Computation Time and Error analysis for f = xsin(x^2)

Unlike the function in section (a), we have an increase in the number of *GQ* evaluations between $n = 2$ and $n = 3$.
This spike is caused by 'luck'. As it happens, the *GQ* evaluation points land just right when $n = 2$ to balance the negative and positive errors. Intuition for that can be provided by looking on the figure below.

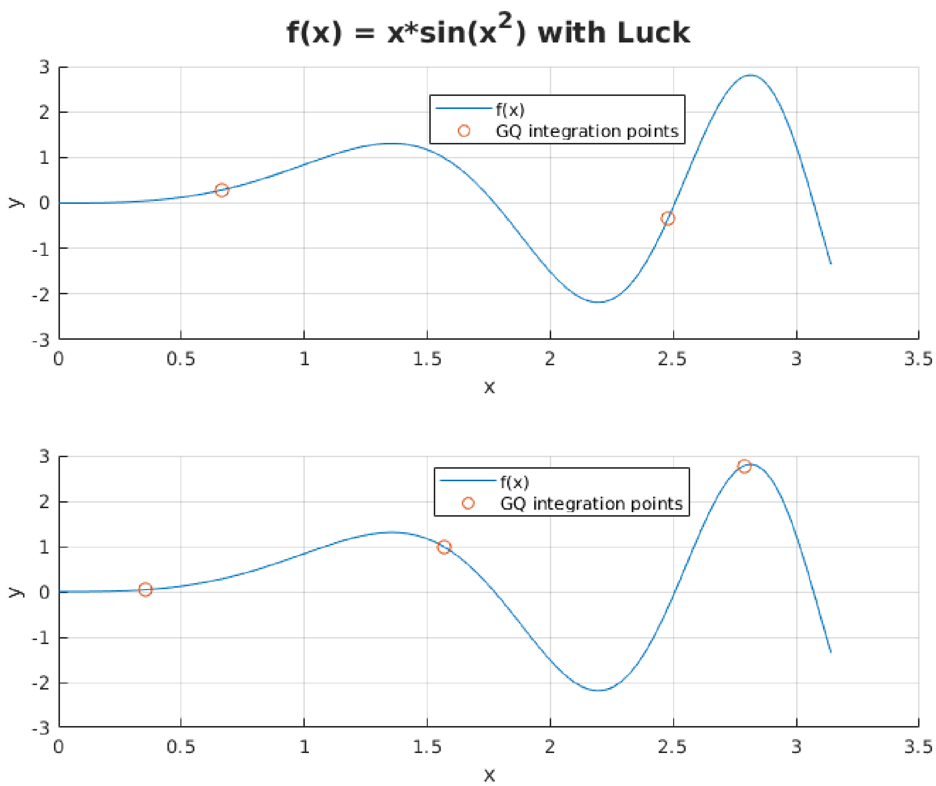

*Figure 6 n=2 vs n=3 for f(x)=x*sin(x^2)*

Other than this fact, the results don’t add or deduct from our previous analysis in section (a), and for that reason we will refer you there.