Lab 11: MATLAB Integration Routines & Gauss Quadrature

Melissa Butler

November 08, 2021

1 Result

1.1 Plot

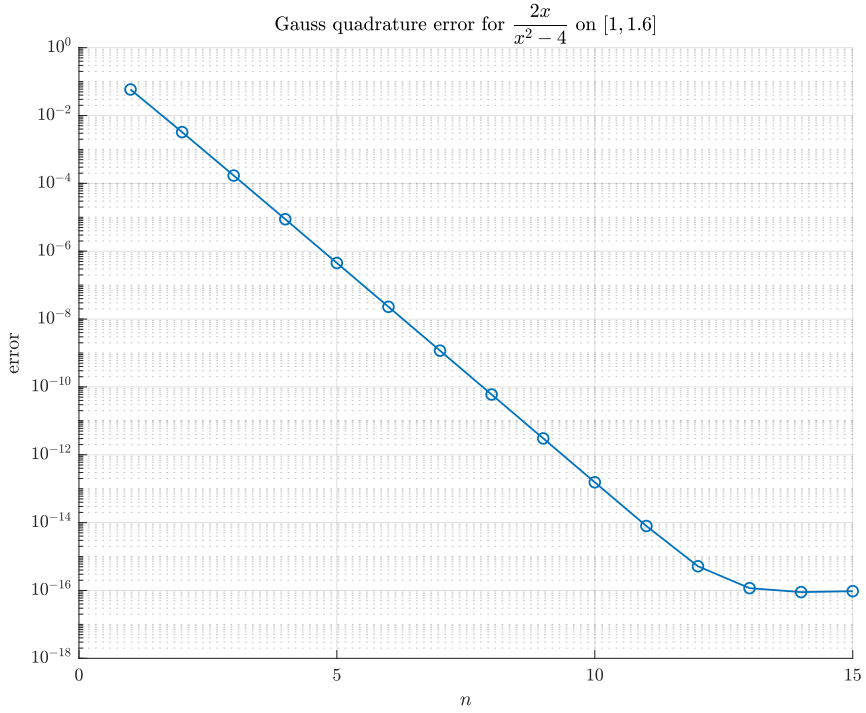

Figure 1: Gauss Qudrature Error

1.2 Output file: lab_11_output.txt

```
lab_11_script

=============== Part 1 ===============

------------------ 1(a) -----------------
Integrating x.^2 - 2.*x + 1 on [-1,3]
 Using polyint I = 5.333333
Using integral I = 5.333333
           Error = 8.881784e-16

------------------ 1(b) -----------------
Integrating x.^2 - 2.*x + 1 on [-1,3]
Using trapz and cumtrapz

   Using trapz I = 6.000000
           Error = 6.666667e-01
Using cumtrapz:
     0.000000
     2.500000
     3.000000
     3.500000
     6.000000

------------------ 1(c) -----------------
Integrating f(x,y) = y*sin(x) + x*cos(y)
 Using integral2 I = -19.739209

------------------ 1(d) -----------------
Integrating f(x,y,z) = y

 Using integral3 I = 0.071429

=============== Part 2 ==============
Integrating 2*x./(x.^2-4) on [1.0 1.6]
Exact Value I = -0.7339691750802005

Using gaussQuad.m :
 N    Gauss Quadrature                Error
 1 -0.6753246753246753  0.0586444997555252
 2 -0.7307230362719193  0.0032461388082813
 3 -0.7337990222564578  0.0001701528237428
 4 -0.7339603934418905  0.0000087816383101
 5 -0.7339687248604583  0.0000004502197422
 6 -0.7339691520791420  0.0000000230010585
 7 -0.7339691739075659  0.0000000011726346
 8 -0.7339691750204980  0.0000000000597025
```

```
 9 -0.7339691750771636  0.0000000000030369
10 -0.7339691750800461  0.0000000000001545
11 -0.7339691750801925  0.0000000000000080
12 -0.7339691750802000  0.0000000000000005
13 -0.7339691750802004  0.0000000000000001
14 -0.7339691750802004  0.0000000000000001
15 -0.7339691750802004  0.0000000000000001
diary off
```

2 Script and Function files

2.1 Script file: lab_11_script_01.m

```
% Math 3341, Fall 2021
% Lab 11: MATLAB Integration Routines & Gauss Quadrature
% Author: Melissa Butler
% Date: 11/08/2021

clear; close all; clc;
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

%% 1 Built-in Integration Functions
fprintf('\n=============== Part 1 ===============\n')
%--------------------------------------------
% Single integrals - integral & polyint
%--------------------------------------------
f = @(x) x.^2 - 2.*x + 1;               % define function p(x)
p = [1, -2, 1];                    % stores coeff values of p(x)
a = -1;                                 % lower bound of integral
b = 3;                                  % upper bound of integral

% 1(a)
% using polyint
pI = polyint(p);                    % calculate the integral with polyint
pI_value = diff(polyval(pI, [a, b]));         % evaluates the integral on [a,b]
% using integral
I = integral(f, a, b);

fprintf('\n------------------ 1(a) -----------------\n')
fprintf('Integrating x.^2 - 2.*x + 1 on [%d,%d]\n', a, b);
fprintf(' Using polyint I = %f\n', pI_value)
fprintf('Using integral I = %f\n', I)
fprintf('           Error = %e\n', diff([pI_value,I]))

%--------------------------------------------
% Single integrals - trapz & cumtrapz
%--------------------------------------------
xdata = -1:3;
ydata = f(xdata);

% 1(b)
% evaluate integral via ydata (discrete data) using trapz
trapI = trapz(xdata, ydata);
cumtrapI = cumtrapz(xdata, ydata);

fprintf('\n------------------ 1(b) -----------------\n')
fprintf('Integrating x.^2 - 2.*x + 1 on [%d,%d]\n', a, b);
fprintf('Using trapz and cumtrapz\n\n')
fprintf('   Using trapz I = %f\n', trapI)
fprintf('           Error = %e\n', diff([I,trapI]))
```

```
fprintf('Using cumtrapz: \n')
fprintf('     %f\n',cumtrapI)

%-------------------------------------------
% Double integrals
%-------------------------------------------

% 1(c)
f3 =  @(x,y) y.*sin(x) + x.*cos(y);
I3 = integral2(f3, -pi, -3 * pi / 2, 0, 2 * pi);

fprintf('\n------------------ 1(c) -----------------\n')
fprintf('Integrating f(x,y) = y*sin(x) + x*cos(y)\n')
fprintf(' Using integral2 I = %f\n', I3)

%-------------------------------------------
% Triple integrals
%-------------------------------------------

% 1(d)
f4 = @(x,y,z) y;
zmin = @(x,y) x-y;
zmax = @(x,y) x+y;
ymin = @(x) x.^2;
ymax = @(x) x;
xmin = 0;
xmax = 1;

I4 = integral3(f4, xmin, xmax, ymin, ymax, zmin, zmax);

% prints results of calculations
fprintf('\n------------------ 1(d) -----------------\n')
fprintf('Integrating f(x,y,z) = y \n\n')
fprintf(' Using integral3 I = %f\n', I4)

%% 2 Gauss Quadrature

% define the function for part II
f = @(x) 2*x./(x.^2-4);
a = 1;
b = 1.6;
% define g here
g = @(x) f((b - a) * x / 2 + (a + b) / 2) * (b - a) / 2;
% find exact value of integral
f_exact= integral(f, a, b);
% print results of calculations
fprintf('\n=============== Part 2 ==============\n')
fprintf('Integrating 2*x./(x.^2-4) on [%.1f %.1f] \n', a, b)
fprintf('Exact Value I = %.16f\n\n', f_exact)

fprintf('Using gaussQuad.m :\n')
% place function calls for gaussQuad below
errors = [];
fprintf('%2s %19s %19s\n', 'N', 'Gauss Quadrature', 'Error');
```

```
for N = 1:15
    f_gauss = gauss_quad(g, N);
    errors(N) = abs(f_gauss - f_exact);
    fprintf('%2d %19.16f %19.16f\n', N, f_gauss, errors(N));
end

% plot error
fig = figure(1); hold on;
plot(errors, 'o-', 'LineWidth', 1);
grid on;
set(gca, 'YScale', 'log');
xlabel('$n$');
ylabel('error');
title('Gauss quadrature error for $\displaystyle \frac{2x}{x^2 - 4}$ on $[1, 1.6]$');
fig.PaperPositionMode = 'auto';
f1_pos = fig.PaperPosition;
fig.PaperSize = [f1_pos(3) f1_pos(4)];
print(fig, '-dpdf', 'lab_11_figure.pdf')
```

2.2 Function file: gauss_quad.m

```
function [val] = gauss_quad(f, N)
%GAUSSQUAD Perform Gauss quadrature on [-1, 1]
% INPUT:
%   f = f(x) to integrate,
%   N = number of legendre points to use
% OUTPUT:
%   val = integral of f(x)

[x, w] = legendre_pair(N);

% Calculate the integral on interval [-1, 1]
val = dot(w, f(x));
% val = 0;
% for j = 1:N
%     val = val + w(j) * feval(f, x(j));
% end

end
```