Table of Contents

hw 3 Problem 1

PART (a) Integrate with qromo

```
profile on

% (i)
func1 = @(x) sqrt(x)/sin(x);
mid1 = NumericalRecipes.Midpnt(func1, 0, pi/2);
i1 = NumericalRecipes.qromo(mid1,1.e-7);   % info in Midpnt class
fprintf('\nqromo : %.10g\n',i1);
% the integral for (i) is 2.7515  (qromo : 2.751522289)

% (ii)
func2 = @(x) sin(x)/x^2;
mid2 = NumericalRecipes.Midinf(func2, pi/2, 10^99);
i2 = NumericalRecipes.qromo(mid2, 1.e-7);
fprintf('\nqromo : %.10g\n',i2);
% the integral for (ii) is 0.1646 (qromo : 0.1646165504)

% (iii)
func3 = @(x) exp(-x)/sqrt(x);
low = NumericalRecipes.Midsql(func3, 0, 2);
mid = NumericalRecipes.Midexp(func3, 2, 10^99);
i3 = NumericalRecipes.qromo(low, 1.e-7);
i3 = i3 + NumericalRecipes.qromo(mid, 1.e-7);

% alternative for func3
% less accurate with only Midexp (comparing w Wolfram Alpha 1.77245)
% mid = NumericalRecipes.Midexp(func3, 0, 10^99);
% i3 = NumericalRecipes.qromo(mid, 1.e-7);

fprintf('\nqromo : %.10g\n',i3);
% the integral for (iii) is 1.7724 (qromo : 1.772461296)

% profile viewer
```

PART (b) improve the convergence

```
% Comment:
% Looking at the profile viewer in part a,
% with all 3 integrals being evaluated with the same error accuracy (1.e-7)
```

```
%  func1 took 177147 calls and 0.29s, func2 took 531441 calls and 0.656s,
%  and func3 took 162 calls and 0.0001s.

% integral (ii) converges the slowest and much more slowly than the
% others. This is because we are using Midinf and the transformation
% (substitution x = 1/u) we conducted ended up up turning the integral
% into sin(1/u) and u goes from 0 to 2/pi,
% When I plot out sin(1/x) within the bounds from 0 to 2/pi, the integral
% is consists of dense oscillations. Therefore, it would take a long time
% calculating through a large amount of oscillations.

% To improve the convergence, I used a simple analytic transformation
% (integration by parts) to transform the orginal integral.
% I took (x^-2) as u and (sinx) as my dv, so v is (-cosx).
% After the transformation, the integral becomes -2*(cosx)/(x^3) and the
% calculation speed was significantly faster; even much faster than the
% other integrals.
% This demonstrates the importance and brilliance of ways to transform
% integrals.

func4 = @(x) -cos(x)*2/x^3;
mid4 = NumericalRecipes.Midinf(func4, pi/2, 10^99);
i4 = NumericalRecipes.qromo(mid4, 1.e-7);
fprintf('\nqromo : %.10g\n',i4);
% this transformation improves the convergence and reduced it to 729 calls
% and 0.006s.

profile viewer
```

```
qromo : 2.751522289

qromo : 0.1646165504

qromo : 1.772461296

qromo : 0.1646636
```

Published with MATLAB® R2022b