# Homework 5

Yash Ganatra

April 3, 2016

## Problem 0: Homework checklist

Please identify anyone, whether or not they are in the class, with whom you discussed your homework. This problem is worth 1 point, but on a multiplicative scale.

- Zhehuan Xu, Nate Majid Kesto, Emily Zimovan,Woi Sok Oh
- Exact value of integral is computed using *Wolfram Alpha*

## Problem 1: Gautschi Exercise 3.29 + more

*2-point Gauss-Hermite quadrature rule.*
*Nodes* - Zeros of the Hermite Polynomial Hermite polynomial is found using the recurrence relation in Equation 1

$$\begin{aligned} &\pi_{k+1}(t) = (t - \alpha_k)\,\pi_k(t) - \beta_k\,\pi_{k-1}(t),\; k = 0, 1, 2, \ldots \\ &\pi_0(t) = 1, \pi_{-1}(t) = 0 \\ &\beta_0 = \int_{-1}^{1} w(t)\,dt \end{aligned} \tag{1}$$

The Jacobi Matrix elements are defined by

$$J_k = \begin{cases} \alpha_k & = & 0 \text{ for } k \geq 0 \\ \beta_k & = & \sqrt{\pi} \text{ if } k = 0 \\ & = & k/2 \text{ if } k > 0 \end{cases}$$

The nodes are the ***eigenvalues of the Jacobi Matrix*** as shown in Equation 2

$$J_n\,\mathcal{V} = t_k\,\mathcal{V}, \mathcal{V}_k^T\,\mathcal{V}_k = 1, k = 1, 2, \ldots, n \tag{2}$$

The weights are computed from the eignevectors as shown in Equation 3

$$w_k = \beta_0\,\mathcal{V}_k^2, k = 1, 2, \ldots, n \tag{3}$$

For the *2 point quadrature rule*, the eigenvectors (1 column represents 1 eigenvector) are

$$\mathcal{V} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

The two weights are 0.8862 and the nodes are $-\frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{2}}$

The given function is integrated using change of variables as shown in Equation 4

$$\begin{aligned} t^2 &= \frac{(x-\mu)^2}{2\,\sigma^2} \\ I &= \sqrt{2}\,\sigma \int_{-\infty}^{\infty} f(1+e^{\sqrt{2}\,\sigma\,t})\; e^{-t^2}\,\mathrm{d}t \\ &\approx \sqrt{2}\,\sigma \sum_{k=1}^{k=2} f(1+e^{\sqrt{2}\,\sigma\,t_k})\,w_k \\ f(x) &= \frac{x^{1+\gamma}}{1+\gamma} \end{aligned} \tag{4}$$

Figure 1 shows the error plot for *n-point* Gauss Hermite Quadrature computed numerically (code in Appendix 1,2) As expected the error decreases as the quadrature nodes are increased. Since the polynomial power is not an integer, the given function can't be integrated exactly. After $n = 8$, the error is of the order $1e-16$.

Hence I would recommend *8-pt Quadrature*

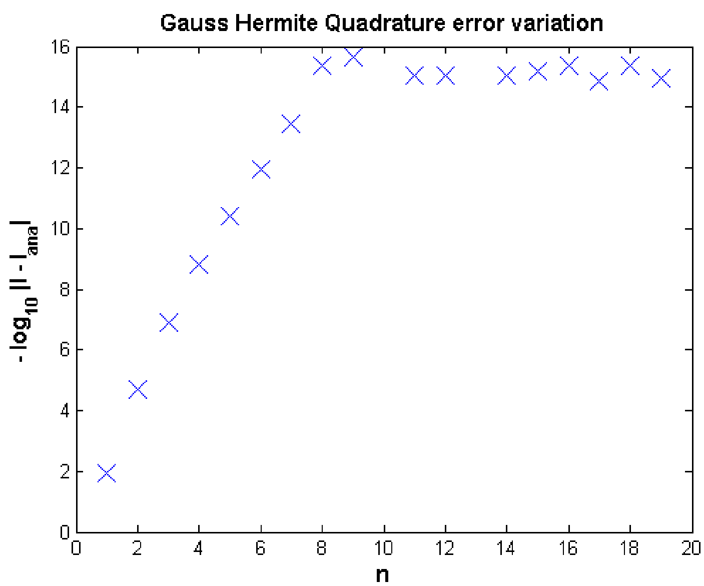


Figure 1: Error plot for *n-point* Gauss Hermite Quadrature

## Problem 2: Multivariate quadrature

Code in Appendix 3,4
Table 1 shows computed integral values

Table 1: 2D Gauss Legendre Quadrature Integration

| Function | Computed value of Integral |
|---|---|
| $x^2 + y^2$ | 2.667 |
| $x^2\,y^2$ | 0.4444 |
| $3e^{x^2+y^2}$ | 8.5574 |
| $4(1-x^2) + 100 * (y-x^2)^2$ | 216.0000 |

*Degree of Exactness of 2D Gauss Quadrature*
Table 2 for compares the computed and actual value of integral for different polynomials and combinations of grid points to find the degree of exactness of multi-variate functions. For univariate functions the degree of exactness is $d = 2n - 1$ where $n$ is the number of quadrature points. For two variable functions the degree of exactness depends on the **highest power of x and y**. The values shaded green show the (optimal / minimum) combination of points for which the points in x and y exactly integrate the function. For the last function , the highest power of $xis4$ and $yis2$, hence there should be a minimum of *3 point in x* and *2 points in y*

Table 2: 2D Gauss-Legendre integration of sample functions to find degree of exactness

| Function | nx | ny | Integral | True value of integral |
|---|---|---|---|---|
| $x^2 + y$ | 1 | 1 | 0 | 1.3333 |
| | 2 | 1 | 1.3333 | |
| | 1 | 2 | 0 | |
| | 3 | 2 | 1.3333 | |
| $x^2\,y^2$ | 2 | 2 | 0.4444 | 0.4444 |
| | 3 | 3 | 0.4444 | |
| $x^2\,y^3$ | 2 | 2 | 0 | 0 |
| | 2 | 3 | -2.7756e-17 | |
| | 3 | 3 | -2.7756e-17 | |
| $x^2\,y^4$ | 2 | 2 | 0.1481 | 0.2667 |
| | 2 | 3 | 0.2667 | |
| $(1-x^2) + 100\,(y-x^2)^2$ | 3 | 2 | 216.0000 | 216.0000 |
| | 2 | 2 | 180.4444 | |

References:

- Abebe Geletu, Integration Methods for Multidimensional Probability Integrals,Technische Universität Ilmenau,IFAC2011 Pre-Conference Tutorial
- Prof. Suvranu De, MANE 4240 & CIVL 4240, Introduction to Finite Elements
- Prof Paulino, Finite Element methods, GeorgiaTech,2014

## Appendix

Listing 1: Gauss Hermite Quadrature function - Nodes and Weights

```
function [nodes,w] = p1_gh_nw(n)
beta_0 = sqrt(pi);
r2 = sqrt((1:(n-1))./2);
J = diag(r2,1); %SuperDiag
i=1:(n-1);
J((i-1).*n+(i+1))= J(i.*(n+1)); %Sub Diag
[V,D] = eig(J);
%% Weights
w = beta_0.*(V(1,:).^2);
%% Nodes
nodes = diag(D);
end
```

Listing 2: Gauss Hermite Quadrature main program - Compute integal

```
clc;close all;clear all;
%% n-pt quadrature rule
my_n = 1:6;
I = zeros(1,length(my_n));
error_abs = zeros(1,length(my_n));
count =0;
%% Compute nodes and weights
for n = my_n
count = count+1;
[t,w] = p1_gh_nw(n);
gamma = -0.5;
sig = 0.25;
mu =0.15;
%% Function Eval
f = @(x) ((x.^(1+gamma))./(1+gamma));
k=1:n;
f_eval = f(1+exp((sqrt(2).*sig.*t)+mu));
%% Evaluate integral
I(count) = sqrt(2)*sig*w*f_eval;
syms x ;
int_ana1 = (1+exp(x))^(1+gamma)*exp(-(x-mu)^2/(2*sig^2))
    /(1+gamma);
g = matlabFunction(int_ana1);
I_ana = integral(g,-inf,inf);
error_abs(count) = abs(I_ana - I(count));
end
figure(1)
```

```
plot(my_n,-log10(error_abs),'kx','MarkerEdgeColor','b'
    ,...
'MarkerFaceColor','g','MarkerSize',14)
xlabel('n','FontSize',12,'FontWeight','bold');
ylabel('- log_{10} |I - I_{ana}|','FontSize',12,'
   FontWeight','bold');
title('Gauss Hermite Quadrature error variation',...
    'FontSize',12,'FontWeight','bold');
```

Listing 3: Gauss Legendre Quadrature function - Nodes and Weights

```
function [nodes,w] = p2_gl_nw(n)
beta_0 = 2;
beta = 1./sqrt(4-power(1:(n-1),-2));
J = diag(beta,1)+diag(beta,-1);
[V,D] = eig(J);
%% Weights
w = beta_0.*(V(1,:).^2);
%% Nodes
nodes = diag(D);
%[x,i] = sort(x) % nodes (= Legendre points)
end
```

Listing 4: 2D Gauss Legendre Quadrature main program - Compute integal

```
clc;close all;clear all;
%% 2D Gauss Legendre Quadrature
nx = 2;
ny = 2;
%% Compute nodes and weights for x and y
[t_x,w_x] = p2_gl_nw(nx);
[t_y,w_y] = p2_gl_nw(ny);
%% Function
f = @(x,y)(x.^2.*y.^2);
%%
%Nodes grid
[xx,yy] = meshgrid(t_x,t_y);
f_eval = f(xx,yy);
%Weights grid
[w_xx,w_yy] = meshgrid(w_x,w_y);
%%
prod = f_eval.*w_xx.*w_yy;
% Integral
sum_c = sum(prod,2);
I = sum(sum_c,1)
I_ana = integral2(f,-1,1,-1,1)
```