October 22, 2018

Empirical Methods HA 4

Konstantin Guryev

Pennsylvania State University

2018

Problem 1 See the code. I use *qnwequi* function from the CEtools. $\pi = 3.1414$ (100000 points).

Problem 2 See the code. I use *newton_coates* function. $\pi = 3.1416$ (10000 points).

Problem 3 See the code. $\pi = 3.1416$ (100000 points). The approximation is more precise with a single dimensional integral given the same number of points.

Problem 4 See the code. $\pi = 3.1416$ (10000 points).

Problem 5

$$Results = \begin{pmatrix} 0.0216 & 0.0016 & 0.0008 \\ 0.0104 & 0.0009 & 0.0000 \\ 0.0113 & 0.0003 & 0.0000 \\ 0.0061 & 0.0002 & 0.0000 \end{pmatrix}.$$

First column refers to 100 draws, second to 1000 and third to 10000 draws. The rows from 1 to 4 refer to Newton-Coates for double integral, Newton-Coates for single integral, quasi-Monte Carlo for double integral and quasi-Monte Carlo for single integral respectively.

Matlab Code

```
%function [x,w] = qnwequi(n,a,b,type)
function [x,w] = qnwequi(n,a,b,type)
global equidist_pp

if isempty(equidist_pp)
  equidist_pp=sqrt(primes(7920));   % good for d<=1000
end

d  = max(length(n),max(length(a),length(b)));
n=prod(n);
if nargin<4, type='N'; end

i=(1:n)';
switch upper(type(1))
  case 'N'                  % Neiderreiter
    j=2.^((1:d)/(d+1));
    x=i*j;
    x=x-fix(x);
  case 'W'                  % Weyl
    j=equidist_pp(1:d);
    x=i*j;
    x=x-fix(x);
  case 'H'                  % Haber
    j=equidist_pp(1:d);
    x=(i.*(i+1)./2)*j;
    x=x-fix(x);
  case 'R'                  % pseudo-random
    x=rand(n,d);
  otherwise
    error('Unknown sequence requested')
end
u=ones(n,1);
r = b-a;
x = a(u,:) + x.*r(u,:);
```

```
w = (prod(r)/n)*u;
```

Matlab Code

```matlab
function [x,w]=newton_coates(n,a,b)
w=zeros(n,1); x=zeros(n,1);
for i=1:n
    if or(i==1,i==n)
        w(i)=1/(3*n);
    elseif mod(i,2)==0
            w(i)=4/(3*n);
    elseif mod(i,2)~=0
        w(i)=2/(3*n);
    end
end
for j=1:n
    if j==1
        x(j)=a;
    else
        x(j)=x(j-1)+((b-a)/n);
    end
end
end
```

Matlab Code

```matlab

%% Problem 1
[x1,w1] = qnwequi(100000,[0,0],[1,1],'N');
p1=x1(:,1).^2+x1(:,2).^2 <= 1;
pimc1 = 4*w1'*p1; %quasi-Monte Carlo

%% Problem 2
f1=@(x,y)(double(x.^2+y.^2<=1));
[x2,wx2] = newton_coates(10000,0,1);
f_val2 = zeros(10000,10000);
for i = 1:length(x2)
      f_val2(i,:) = f1(repmat(x2(i),1,length(x2)),x2');
end
pinc1 = 4*wx2'*f_val2*wx2; %Newton-Coates

%% Problem 3
[x3,w3] = qnwequi(100000,0,1,'N');
pimc2 = 4*w3'*(1-x3.^2).^0.5; %quasi-Monte Carlo

%% Problem 4
[x4,wx4] = newton_coates(10000,0,1);
f_val4 = (1-x4.^2).^0.5;
pinc2 = 4*wx4'*f_val4;%Newton-Coates

%% Problem 5
simul = zeros(4,2);
f1=@(x,y)(double(x.^2+y.^2<=1));
draws=[100,1000,10000];
for k=1:length(draws)
      [a,b]=newton_coates(draws(k),0,1);
      f_val1=zeros(length(a),length(a));
      for i=1:length(a)
      f_val1(i,:)=f1(repmat(a(i),1,length(a)),a');
      end
```

```
    simul(3,k)=4*b'*f_val1*b;
end
clear a b f_val1 k
for l=1:length(draws)
    [c,d]=newton_coates(draws(l),0,1);
    f_val3=(1-c.^2).^0.5;
    simul(4,l)=4*d'*f_val3;
end
clear c d f_val3 l
for m=1:length(draws)
    [e,f]=qnwequi(draws(m),[0,0],[1,1],'N');
    p1=e(:,1).^2+e(:,2).^2<=1;
    simul(1,m)=4*f'*p1;

end
clear e f test1 m

for n=1:length(draws)
    [g,h]=qnwequi(draws(n),0,1,'N');
    simul(2,n)=4*h'*(1-g.^2).^0.5;
end
clear g h n
results = abs(simul-pi);
```