MATH 416/516 ASSIGNMENT 6 SOLUTIONS

- Problem 10.7: the simplest method could use the uniform distribution on [0,1],
  so that $E[X] \approx \frac{\sum_{i=1}^{N} X_i e^{X_i + X_i^2}}{\sum_{i=1}^{N} e^{X_i + X_i^2}}$, with $X_i \sim Uni(0,1)$.
  You could also sample from e^x, on [0,1], with cdf $(e^x - 1)/(e - 1)$,
  so that $E[X] \approx \frac{\sum_{i=1}^{N} X_i e^{X_i^2}}{\sum_{i=1}^{N} e^{X_i^2}}$, with $X_i = \ln(1 + (e-1)U_i)$ with $U_i \sim Uni(0,1)$.

- Problem 10.9: a, c) Matlab for raw MC

```
R = 100; U = rand(10,R); T = exp(sum(U));
disp([mean(T) var(T) 2*std(T)/sqrt(R)])
       250.83        42032       41.003
R = 100; tic; U = rand(10,R); T = exp(sum(U));
disp([mean(T) var(T) 2*std(T)/sqrt(R) toc])
       222.08        44694       42.282  0.003697
```

  This is not very accurate, with estimated error ≈ 40

  b, c) Matlab for simple LHS

```
R = 100; U = rand(10,R);
for i = 1 : 10, US(i,:) = ( randperm(R)-1 + U(i,:) )/R; end
L = exp(sum(US)); disp([mean(L) var(L) 2*std(L)/sqrt(R)])
       224.84        51174       45.243
R = 100; tic; U = rand(10,R);
for i = 1 : 10, US(i,:) = ( randperm(R)-1 + U(i,:) )/R; end
L = exp(sum(US)); disp([mean(L) var(L) 2*std(L)/sqrt(R) toc])
       215.99        65087       51.024  .007684
```

  d) Simple LHS does not seem to be better, maybe a little worse, and it takes more time.

- Problem 11.1a, just use book algorithm Chi-Square test, with Matlab

```
n = 564; k = 3; p = [.25 .5 .25]; N = [141 291 132]; t = 0;
for i = 1 : k
  t = t + ( N(i) - n*p(i) )^2/(n*p(i));
end, tu = t; disp( [ tu 1-chi2cdf(t,k-1) ] )
        0.8617      0.64996
```

  The null hypothesis is accepted because of the large p value.

- Problem 11.2a, just use book algorithm Chi-Square test, with Matlab

```
n = 1000; k = 6; p = ones(1,6)/6;
N = [158 172 164 181 160 165]; t = 0;
for i = 1 : k
  t = t + ( N(i) - n*p(i) )^2/(n*p(i));
end, tu = t; disp( [ tu 1-chi2cdf(t,k-1) ] )
         2.18      0.82372
```

  The null hypothesis is accepted because of the large p value.

- Problem 11.4, just use book algorithm for Kolmogorov-Smirnov, but you need the cdf $F(x) = (x - 50)/150$; with Matlab

```
cdf = @(x)(x-50)/150;
Y = [164 142 110 153 103 52 174 88 178 184 58 62 132 128];
n = 14; YS = sort(Y); X = [1:n]/n; Xm = [0:n-1]/n;
d = max(max([ X-cdf(YS); cdf(YS)-Xm ]))
      0.13429
for i = 1:500, US = sort(rand(1,n));
  I(i) = max(max([X-US;US-Xm])) > d;
end, p = mean(I)
    0.934
```

  The null hypothesis is accepted because of the large p value.

- Problem 11.5, just use book algorithm for Kolmogorov-Smirnov, with Matlab

```
Y = [86 133 75 22 11 144 78 122 8 146 33 41 99];
n = 13; YS = sort(Y); X = [1:n]/n; Xm = [0:n-1]/n;
cdf = @(x)1-exp(-x/50);
d = max(max([ X-cdf(YS); cdf(YS)-Xm ]))
    .39225
for i = 1:500, US = sort(rand(1,n));
  I(i) = max(max([X-US;US-Xm])) > d;
end, p = mean(I)
   0.024
```

  The null hypothesis is rejected because of the small p value.

- Problem 11.7, first use the data to estimate the mean for the exponential $\lambda = 125$, then use the Kolmogorov-Smirnov test, with cdf $F(x) = 1 - e^{-x/125}$; with Matlab

```
cdf = @(x)1-exp(-x/125);; Y = [122 133 106 128 135 126];;
n = 6; YS = sort(Y); X = [1:n]/n; Xm = [0:n-1]/n;
d = max(max([ X-cdf(YS); cdf(YS)-Xm ])), disp(d)
     0.57173
for i = 1:500, US = sort(rand(1,n));
  I(i) = max(max([X-US;US-Xm])) > d;
end, p = mean(I)
    0.016
```

  The null hypothesis is rejected because of the small p value.

- Problem 11.12*: too tedious to compute by hand; Matlab "ranksum" function computes $p = .682$.

- Problem 11.13: Matlab

```
X = [19 31 39 45 47 66 75]; n = 7; Y = [28 36 44 49 52 72 72]; m = 7;
XY = sort([ X Y ]); nm = n + m; r = 0;
for i = 1:n, for j = 1:nm, if X(i) == XY(j), r = r+j; break, end, end
end, disp(r)
    49
rs = (r-n*(nm+1)/2)/sqrt(n*m*(nm+1)/12), disp([rs 2*normcdf(rs)]) % 13a
  -0.44721        0.65472
% Simulation to Approximate p      13b
for k = 1:1000, S = [1:nm]; rr = 0;
  % Determine rr for random subset
  for i = nm : -1 : nm-n+1, j = ceil(i*rand);
    rr = rr + S(j); S(j) = S(i);
  end, M(k) = rr <= r; N(k) = rr >= r;
end, disp( 2*min( mean(M), mean(N) ) )
      0.7116
```

  The null hypothesis is accepted because of large p values from approximation and simulation.

- Problem 12.4: For Gibb's sampling, start the chain with a vector $\mathbf{X}_0 = (X_{1,0}, X_{2,0}, \ldots, X_{10,0})$ satisfying the condition $\prod_{i=1}^{10} X_{i,0} > 20$, say with all $X_{i,0} = 1.35$.
  Basic Algorithm Step: given $\mathbf{X}_k$, define $P_{-j} = \prod_{i \neq j} X_{i,k}$,
  a) set $\mathbf{X}_{k+1} = \mathbf{X}_k$
  b) set $j = \lceil 10U \rceil$, with $U \sim Uni(0,1)$;
  c) generate the new $X_{j,k+1} \sim Exp(1)$, conditional on $X_{j,k+1} > 20/P_{-j}$;
  details: get a new $U \sim Uni(0,1)$, and let $c = 20/P_{-j}$; then invert
  $U = \int_c^{X_{j,k+1}} e^{-x}dx / \int_c^{\infty} e^{-x}dx = (e^{-c} - e^{-X_{j,k+1}})/e^{-c}$, to get $X_{j,k+1} = c - \ln(1-U)$.
  End.

- Problem 12.5: For Gibb's sampling, start the chain with a vector $\mathbf{X}_0 = (X_{1,0}, X_{2,0}, \ldots, X_{n,0})$ satisfying the condition $a_1 X_{1,0} < a_2 X_{2,0} < \ldots < a_n X_{n,0}$, say by picking $X_{1,0} \sim Uni(0,1), X_{2,0} \sim Uni(a_1 X_{1,0}/a_2, 1), \ldots, X_{n,0} \sim Uni(a_{n-1} X_{n-1,0}/a_n, 1)$.
  Basic Algorithm Step: given $\mathbf{X}_k$,
  a) set $\mathbf{X}_{k+1} = \mathbf{X}_k$
  b) set $j = \lceil nU \rceil$, with $U \sim Uni(0,1)$;
  c) generate the new $X_{j,k+1} \sim \begin{cases} Uni(0, a_2 X_{2,k}/a_1) & \text{if } j = 1 \\ Uni(a_{n-1} X_{n-1,k}/a_n, 1) & \text{if } j = n \\ Uni(a_{j-1} X_{j-1,k}/a_j, a_{j+1} X_{j+1,k}/a_j) & \text{otherwise} \end{cases}$;
  End.
- Problem 12.6: for Gibbs sampling, given y, the pdf for x is Ce^{-xy}, so an RV $X \sim Exp(y)$ conditioned on $x < 1$. To find X, given U, solve $U = K \int_0^X e^{-xy} dx$ with K determined by $K \int_0^1 e^{-xy} dx = 1$. So $K = y/(1 - e^{-y})$, and $X = -\ln(1 - (1 - e^{-y})U)/y$. The analysis for Y is similar. The Gibbs sampling algorithm alternates randomly between X and Y. Matlab results

```
N = 10000; X = ones(1,2)/2;
for n = 1:N, i = ceil(2*rand);
  X(i) = -log( 1 - rand*(1 - exp(-X(3-i)) ) )/X(3-i);
  V(n) = prod(X); W(n) = X(1);
end
disp([mean(V) mean(W); 2*std(V)/sqrt(N) 2*std(W)/sqrt(N)])
     0.20905      0.46046
     0.003978    0.0057347
```

- Problem 12.7: For Gibb's sampling, start the chain $\mathbf{X}_0 = (X_{1,0}, X_{2,0}, \ldots, X_{9,0})$, satisfying the condition that no two points are closer than .1, say with $X_{i,0} = (2i - 1)/18$.
  Basic Algorithm Step: given $\mathbf{X}_k$:
  a) set $\mathbf{X}_{k+1} = \mathbf{X}_k$
  b) set $j = \lceil 9U \rceil$, with $U \sim Uni(0,1)$;
  c) generate the new $X_{j,k+1} \sim \begin{cases} Uni(0,\ X_{2,k} - .1) & \text{if } j = 1 \\ Uni(X_{8,k} + .1,\ 1) & \text{if } j = 9 \\ Uni(X_{j-1,k} + .1, X_{j+1,k} - .1) & \text{otherwise} \end{cases}$;
  End.

- Problem 12.9
  a) with Gibbs sampling, this is similar to lecture problem, Matlab

```
N = 10000; n = 3; c = 15; X = 6*ones(n,1);
for m = 1 : N; i = ceil(n*rand);
  S = sum([1 2 3]*X) - i*X(i);
  X(i) = max((c - S)/i,0) - log(rand);
  W(m) = S + i*X(i);
end, disp([mean(W) 2*std(W)/sqrt(N)])
  18.104     0.061228
```

  b) the conditional RV generation for x_i solves $U = K\int_0^{X_i} e^{-x_i}$, with $K = 1/(1-e^{-(1-S)/i})$, so $X_i = -\ln(1-U(1-e^{-(1-S)/i}))$, with $S = \sum_{j,j\neq i} jx_j$. Matlab simulation results

```
N = 100000; n = 3; X = ones(n,1)/10;
for m = 1 : N; i = ceil(n*rand); S = sum([1 2 3]*X) - i*X(i);
  X(i) = -log(1-rand*(1-exp(-(1 - S)/i))); W(m) = S + i*X(i);
end, disp([mean(W) 2*std(W)/sqrt(N)])
   0.72646    0.0012835
```

- Problem 12.11: for Gibbs sampling, given y, z, the pdf for x is $c(y,z)e^{-x(1+y+z)}$, so the RV $X \sim Exp(1+y+z)$, so $X = -\ln(U)/(1+Y+Z)$, with $U \sim Uni(0,1)$, can be used.
  The analysis for Y and Z is similar, so Gibb's sampling uppdates X and Y and Z randomly, with the same formula. Matlab simulation results are

```
N = 100000; X = zeros(3,1);
for n = 1:N, i = ceil(3*rand);
  X(i) = -log(rand)/(1+sum(X)-X(i)); V(n) = prod(X);
end, disp([mean(V) 2*std(V)/sqrt(N)])
         0.085305    0.0032149
```

  So $E[XYZ] \approx .085$.

- Problem 12.12*: the mixed pmf/pdf is proportional to $\binom{n}{i} y^{i+1}(1-y)^{n-i+2}\frac{4^n}{n!}$.
  For Gibbs sampling, given y and n, $i \sim Binomial(n,y)$;
  given i and n, $y \sim Beta(i+2, n-i+3)$; and
  given i and y, the pmf for n is proportional to $\frac{4^{n-i}(1-y)^{n-i}}{(n-i)!}$, so $n \sim i + Poisson(4(1-y))$.

```
K = 100000; I = 1; Y = .5; N = 2;
for k = 1 : K,
  I = binornd(N,Y); Y = betarnd(I+2,N-I+3);
  N = I + poissrnd(4*(1-Y)); X(k,:) = [I Y N];
end, disp([mean(X);2*std(X)/sqrt(K)])
      1.6068    0.40038     4.0080
      0.0095    0.00127     0.0127
```

  Results are $E[I] \approx 8/5$, $E[Y] \approx 2/5$, $E[N] \approx 4$, consistent with results from the respective means (ny, $(i+2)/(n+5)$, and $i + 4(1-y)$) for the three distributions.