Chapter 6: Solutions to Exercises

Exercise 1

Part (a)

$$\begin{aligned}
p(x|y) &\underset{x}{\propto} \mathbb{1}(|x-y|<c)\mathbb{1}(0<x<1) \\
&= \mathbb{1}(y-c<x<y+c)\mathbb{1}(0<x<1) \\
&= \mathbb{1}(\max\{0,y-c\}<x<\min\{1,y+c\}) \\
&\underset{x}{\propto} \text{Uniform}\big(x \mid \max\{0,y-c\},\, \min\{1,y+c\}\big).
\end{aligned}$$

By symmetry,

$$p(y|x) = \text{Uniform}\big(y \mid (\max\{0,x-c\},\, \min\{1,x+c\})\big).$$

Hence, the Gibbs sampler (in the (x, y) parametrization) is to initialize x, y and then alternately sample:

$$\begin{aligned}
X|y &\sim \text{Uniform}(\max\{0,y-c\},\, \min\{1,y+c\}), \\
Y|x &\sim \text{Uniform}(\max\{0,x-c\},\, \min\{1,x+c\}).
\end{aligned}$$

Part (b)

See code in Appendix.

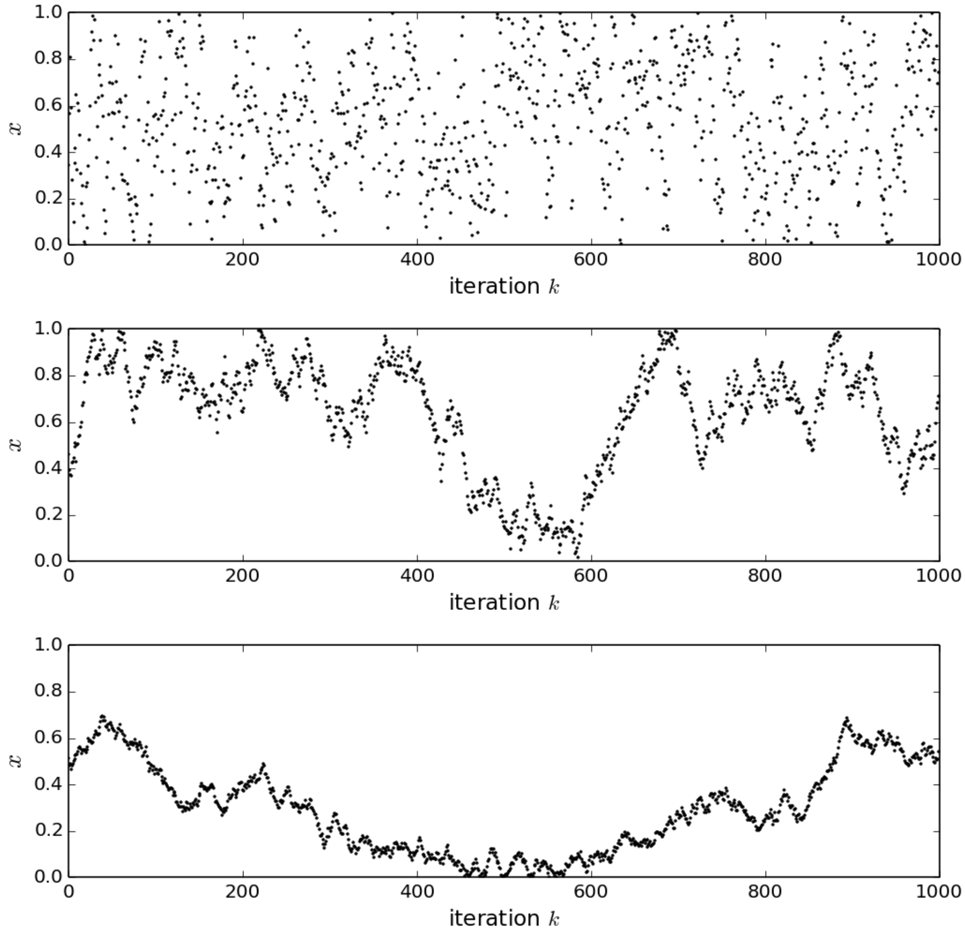
Part (c)

See Figure 1.

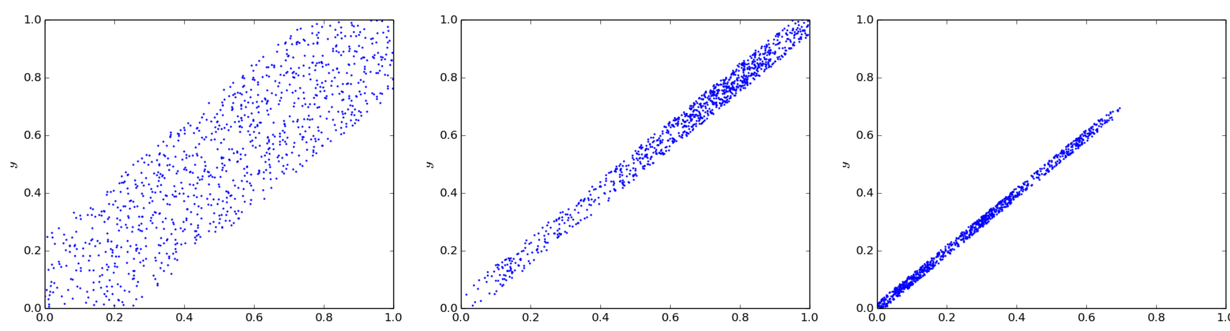
Part (d)

The issue is that when c is small, the range of each variable is highly restricted, given the other variable. As a result, the sampler has to make many "zig–zag" moves in order to travel long distances (see Figure 2). This causes the correlation between successive samples to be high when c is small, as we can see in the traceplots of x—the traceplot indicates that it takes a long time to see significant changes in the samples. This makes the performance of the sampler worse—it takes many more samples to achieve the same level of accuracy when making approximations based on these samples.

Jeffrey W. Miller (2015). *Lecture Notes on Bayesian Statistics*. Duke University, Durham, NC.



(a) Traceplots of x for $c = 0.25$, $c = 0.05$, and $c = 0.02$.



(b) Scatterplots of (x, y) samples for $c = 0.25$, $c = 0.05$, and $c = 0.02$.

Figure 1: Results for Exercise 1.

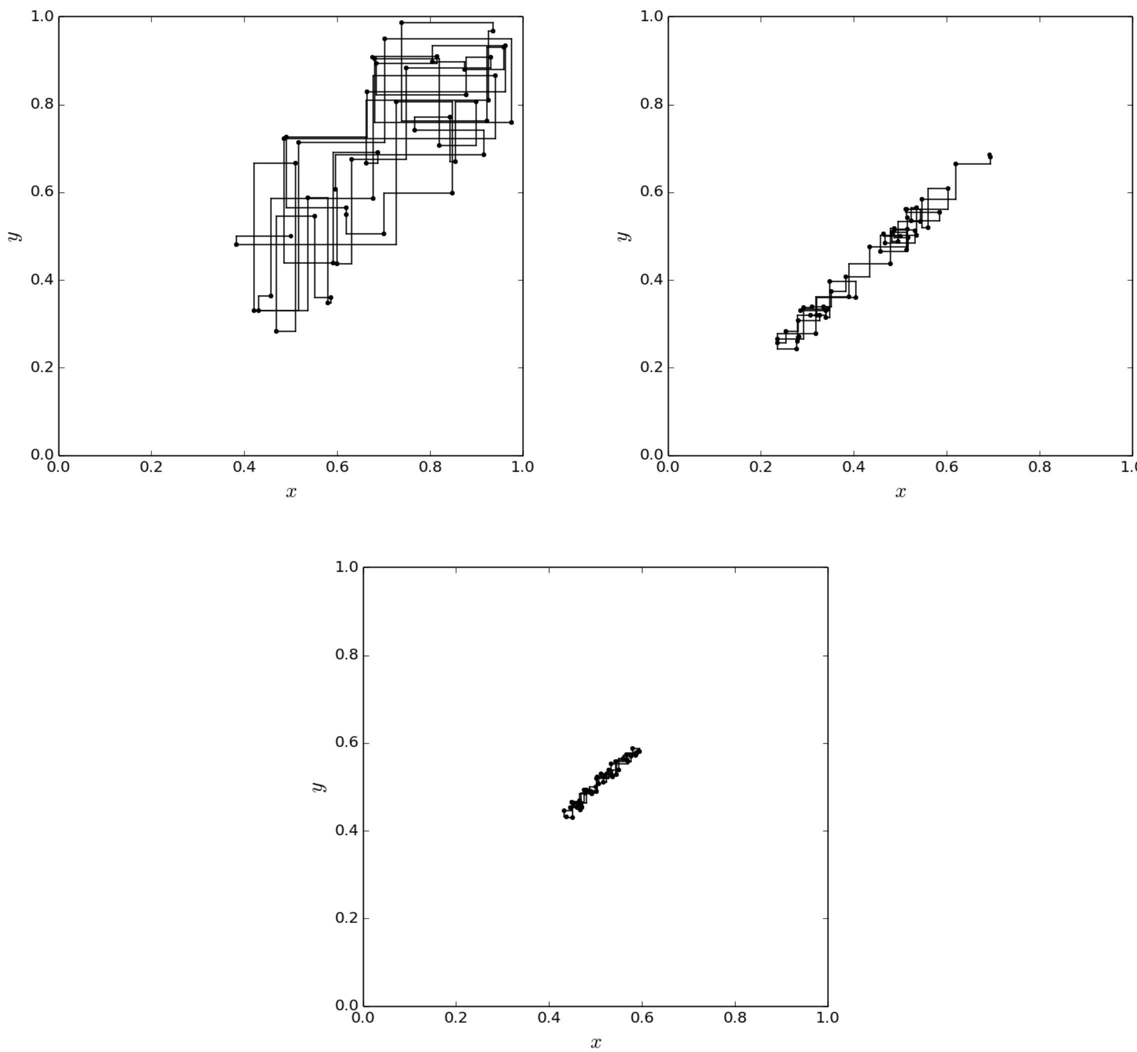


Figure 2: Path of the Gibbs sampler for first few (x, y) samples, for $c = 0.25$, $c = 0.05$, and $c = 0.02$.

Exercise 2

Part (a)

$$
\begin{aligned}
p(u|v) &\underset{u}{\propto} \mathbb{1}(|v| < u < 1 - |v|) \underset{u}{\propto} \text{Uniform}\left(u \,\middle|\, |v|, 1 - |v|\right) \\
p(v|u) &\underset{v}{\propto} \mathbb{1}(|v| < c/2)\mathbb{1}(|v| < u < 1 - |v|) \\
&= \begin{cases} -u < v < u & \text{if } u \le c/2 \\ -c/2 < v < c/2 & \text{if } c/2 < u \le 1 - c/2 \\ u - 1 < v < 1 - u & \text{if } 1 - c/2 < u \end{cases} \\
&\propto \begin{cases} \text{Uniform}(v \mid -u, u) & \text{if } u \le c/2 \\ \text{Uniform}(v \mid -c/2, c/2) & \text{if } c/2 < u \le 1 - c/2 \\ \text{Uniform}(v \mid u - 1, 1 - u) & \text{if } 1 - c/2 < u \end{cases}
\end{aligned}
$$

Part (b)

See code in Appendix.

Part (c)

See Figure 3. The reason why this works so much better is that u can move long distances, given v — the strong coupling that was present between x and y is not exhibited by u and v. This significantly reduces the correlation between samples, and will result in far better approximation accuracy for the same number of samples.

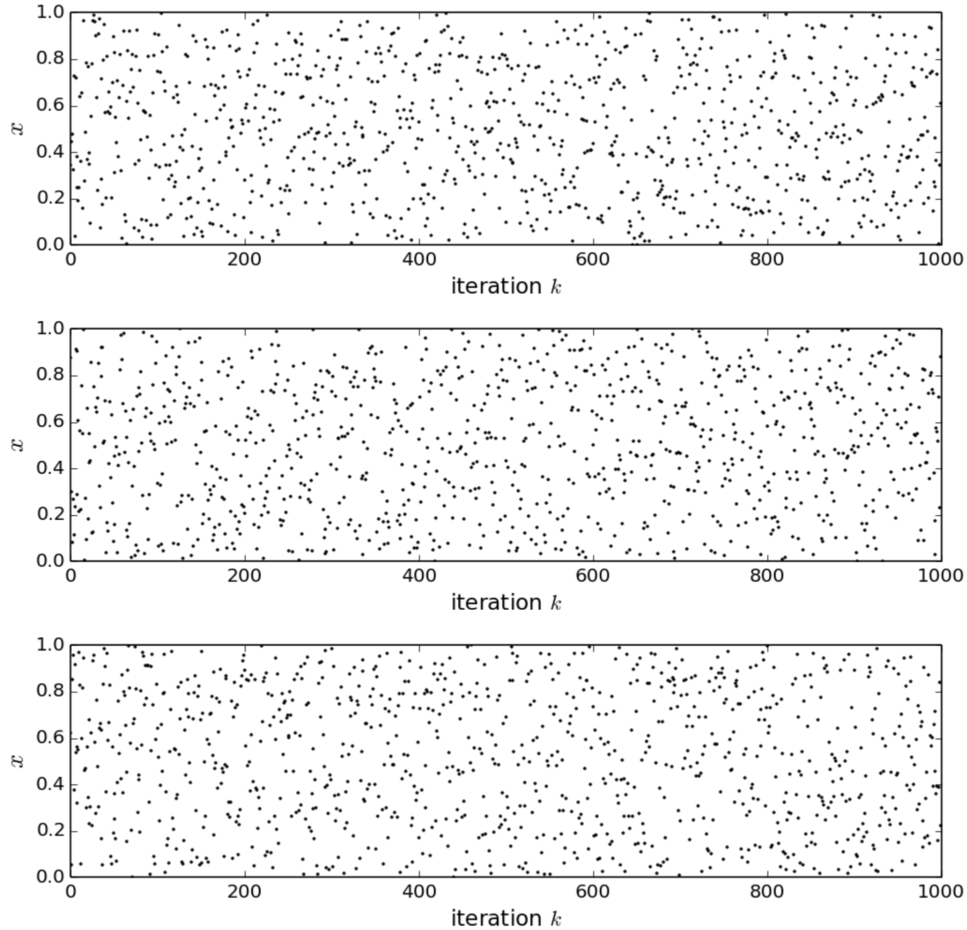
A Source code in Julia language

Exercises 1 and 2

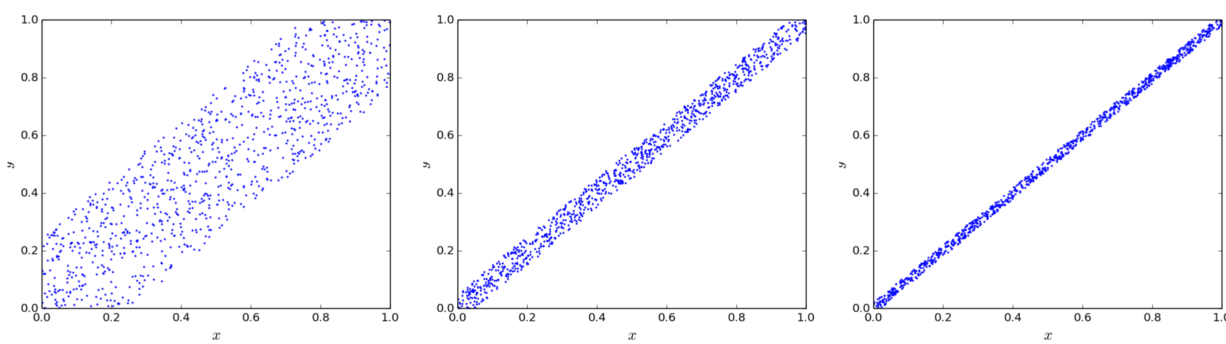
```
module Box

using PyPlot, Distributions
draw_now() = (pause(0.001); get_current_fig_manager()[:window][:raise_]())

rand_uniform(a,b) = rand()*(b-a)+a
rand_box(a,c) = rand_uniform(max(0,a-c),min(1,a+c))
rand_u(v,c) = rand_uniform(abs(v),1-abs(v))
function rand_v(u,c)
    @assert(c<=1)
    if u<=c/2
        return rand_uniform(-u,u)
    elseif u<=1-c/2
        return rand_uniform(-c/2,c/2)
    else
        return rand_uniform(u-1,1-u)
    end
```



(a) Traceplots of x for $c = 0.25$, $c = 0.05$, and $c = 0.02$.



(b) Scatterplots of (x, y) samples for $c = 0.25$, $c = 0.05$, and $c = 0.02$.

Figure 3: Results for Exercise 2, using transformed Gibbs sampler.

```
end

for c in [0.25,0.05,0.02]
    # ========== Untransformed ==========
    x,y = 0.5,0.5
    N = 10^3
    xr = zeros(N)
    yr = zeros(N)
    for i=1:N
        x = rand_box(y,c)
        y = rand_box(x,c)
        xr[i] = x
        yr[i] = y
    end

    figure(1,figsize=(6.5,6)); clf(); hold(true)
    subplots_adjust(bottom = 0.2)
    plot(xr,yr, "b.", markersize=3)
    xlim(0,1)
    ylim(0,1)
    xlabel("\$x\$",fontsize=16)
    ylabel("\$y\$",fontsize=16)
    draw_now()
    savefig("box-scatter-c=$c.png",dpi=120)

    figure(2,figsize=(10,3)); clf(); hold(true)
    subplots_adjust(bottom = 0.2)
    plot(xr, "k.", markersize=3)
    ylim(0,1)
    xlabel("iteration \$k\$",fontsize=14)
    ylabel("\$x\$",fontsize=16)
    draw_now()
    savefig("box-x_trace-c=$c.png",dpi=120)


    # ========== Transformed ==========
    u,v = 0.5,0.0
    N = 10^3
    xr = zeros(N)
    yr = zeros(N)
    for i = 1:N
        u = rand_u(v,c)
        v = rand_v(u,c)
        xr[i] = u + v
        yr[i] = u - v
    end

    figure(3,figsize=(6.5,6)); clf(); hold(true)
    subplots_adjust(bottom = 0.2)
    plot(xr,yr, "b.", markersize=3)
```

```
    xlim(0,1)
    ylim(0,1)
    xlabel("\$x\$",fontsize=16)
    ylabel("\$y\$",fontsize=16)
    draw_now()
    savefig("box-scatter-c=$c-transformed.png",dpi=120)

    figure(4,figsize=(10,3)); clf(); hold(true)
    subplots_adjust(bottom = 0.2)
    plot(xr, "k.", markersize=3)
    ylim(0,1)
    xlabel("iteration \$k\$",fontsize=14)
    ylabel("\$x\$",fontsize=16)
    draw_now()
    savefig("box-x_trace-c=$c-transformed.png",dpi=120)
end

end
```