recTiles

Julia Schedler

2023-07-10

Table of contents

Tiles

This is Julia's transcription of what Alex has been showing her so far about his tile stuff.

Julia's First Tiles

Hand-drawn

First Tile

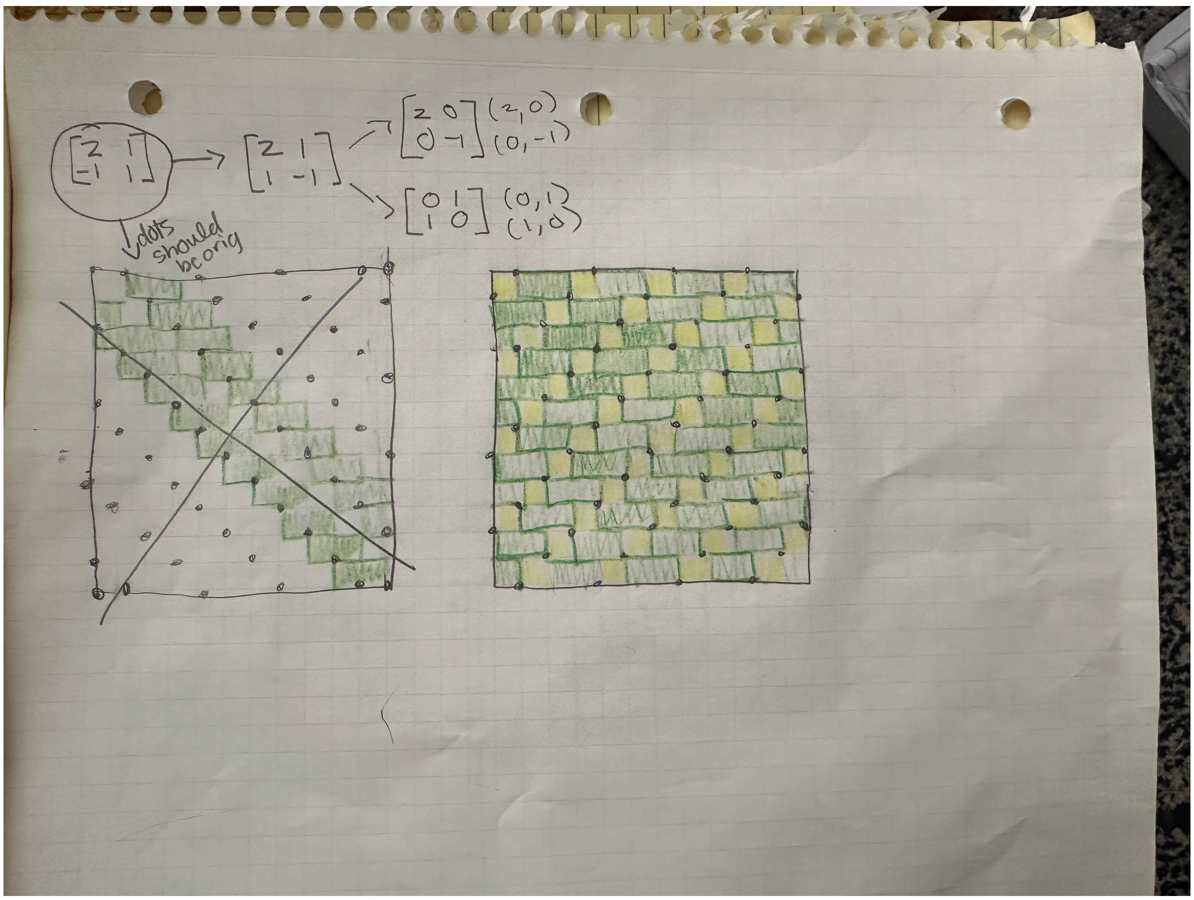
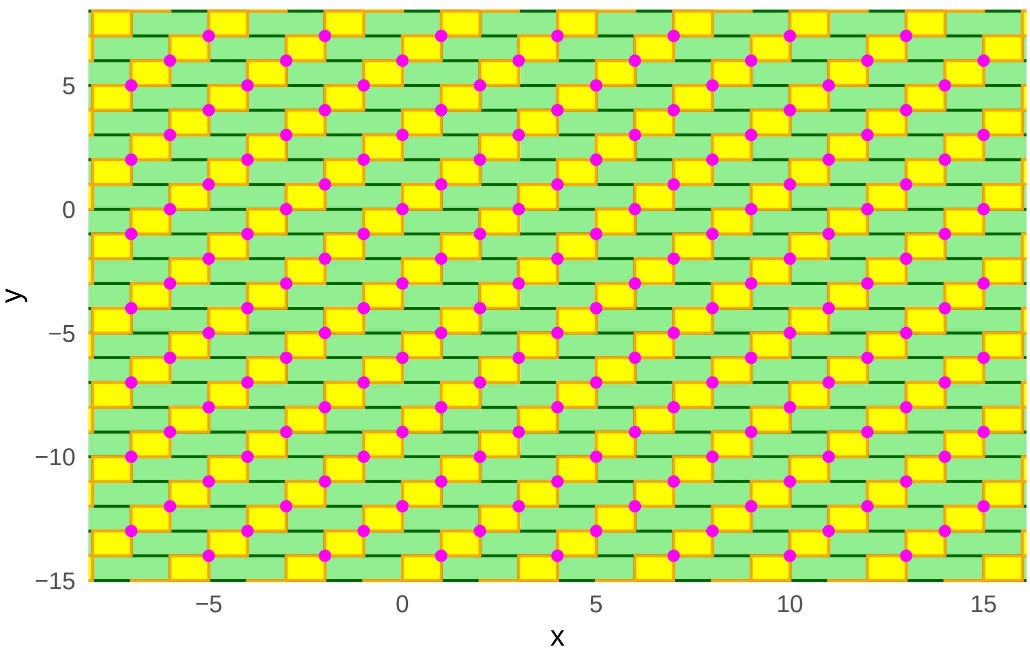


Figure 1: Actually I did this one second, but this is the first matrix I chose. Alex suggested I change to the one visualized next.

$$M = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}$$



Second Tile

$$M = \begin{bmatrix} 2 & 5 \\ -1 & 3 \end{bmatrix}$$

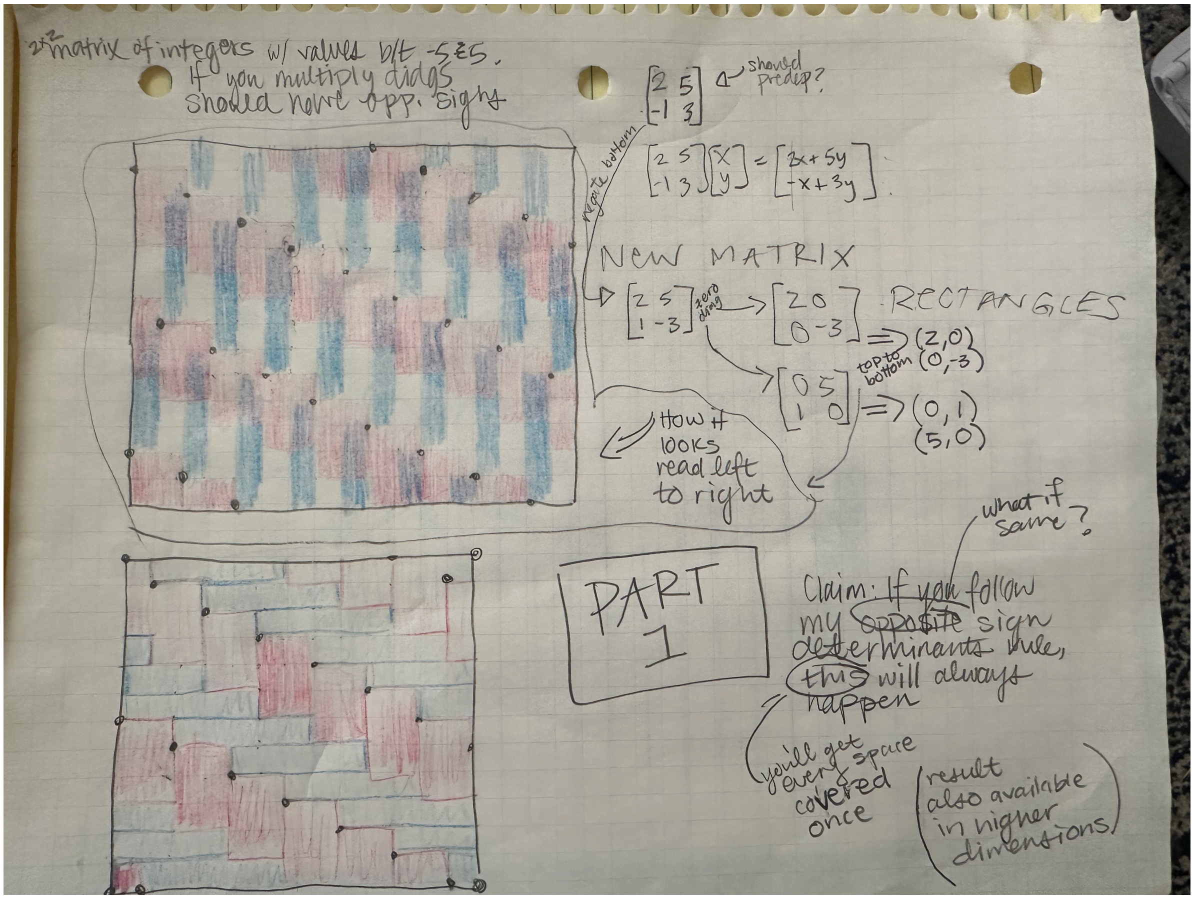
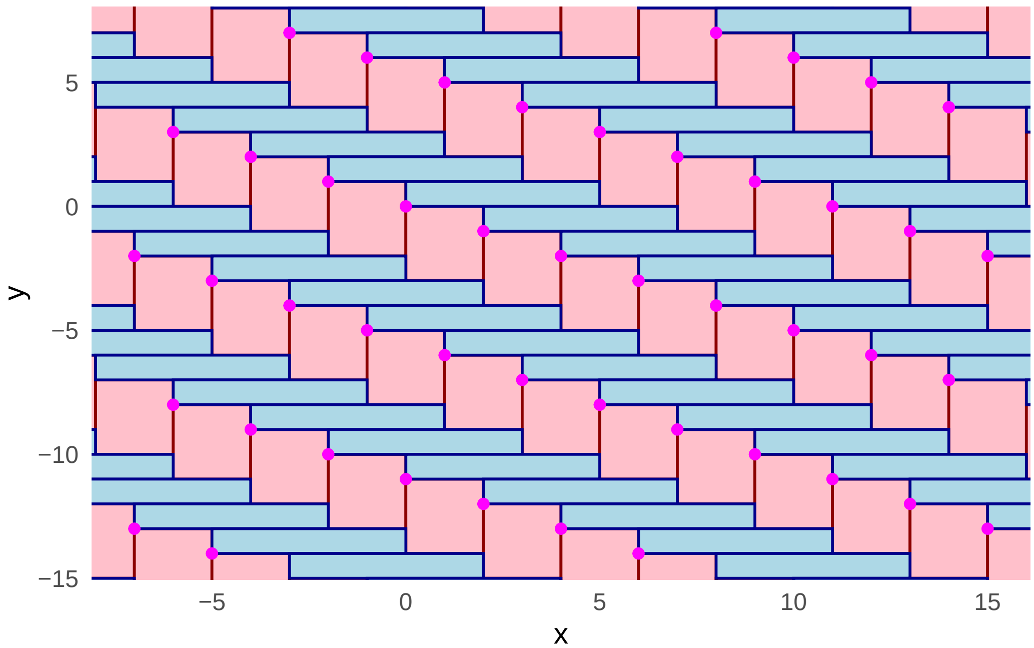
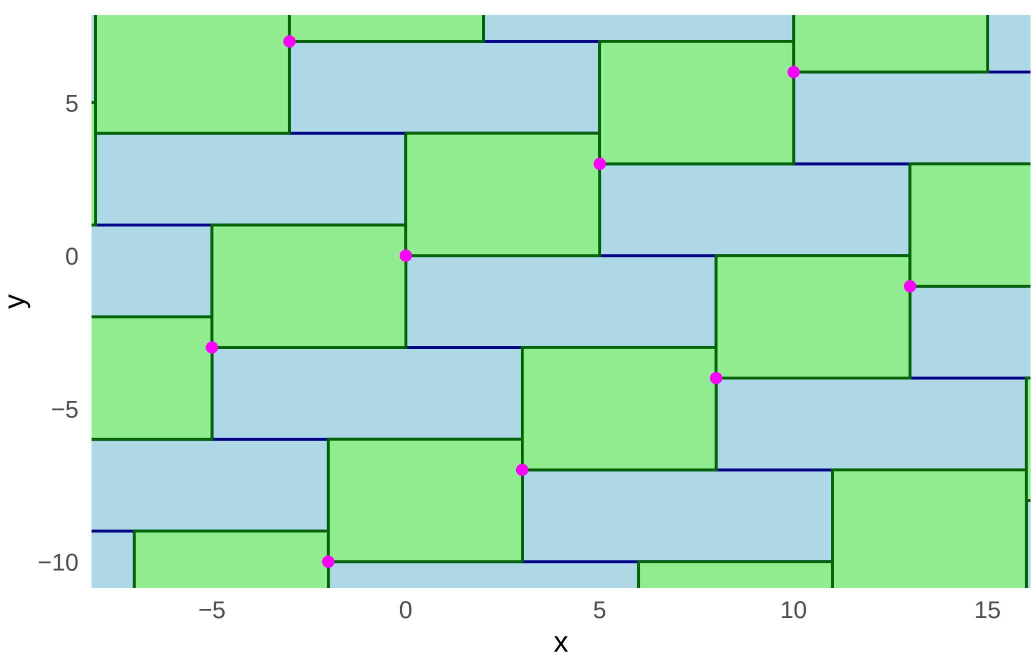


Figure 2: The second tiling, actually the very first that I drew (well, the first I drew correctly--see my mistake?)



Third Tile

$$M = \begin{bmatrix} 8 & 5 \\ -4 & 3 \end{bmatrix}$$



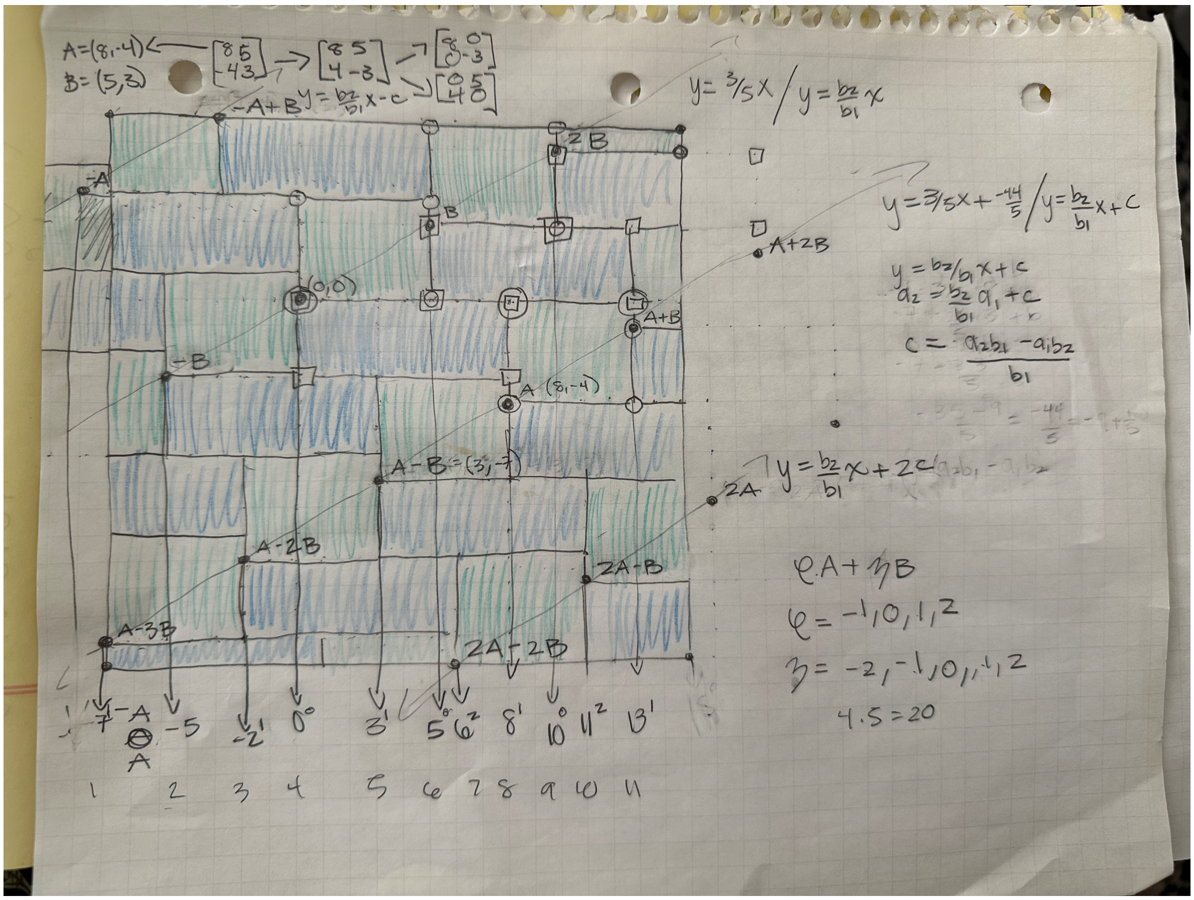


Figure 3: The drawing for the third tiling. I did this one for fun by myself! I chose big numbers so I could have space to write out coordinates and work out some math for the purposes of coding it up.

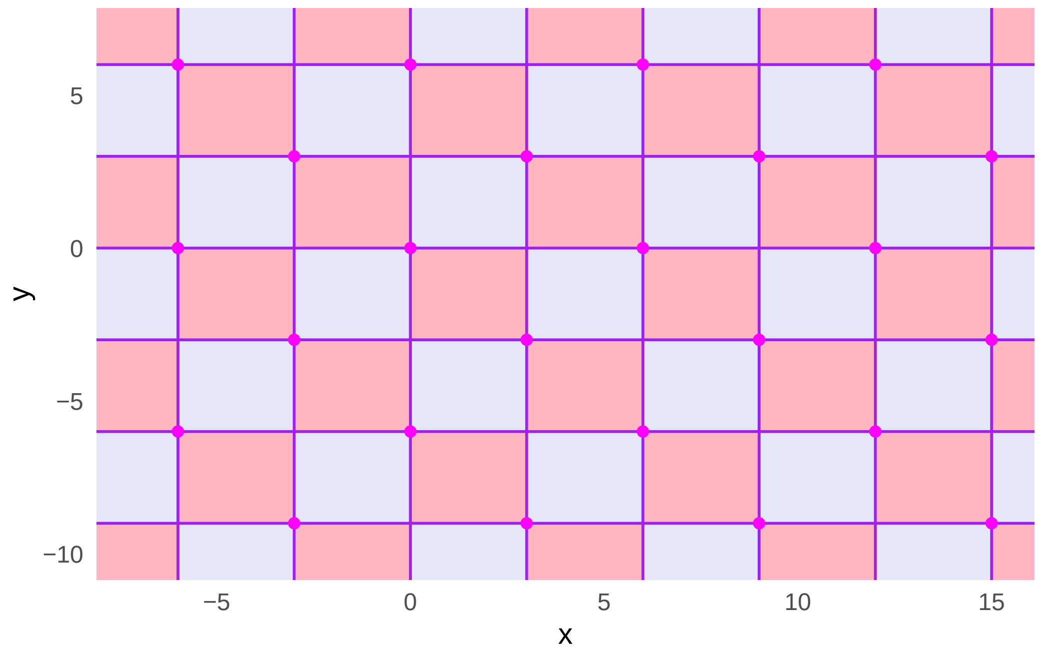
Computer-generated

Fourth Tile

$$M = \begin{bmatrix} 3 & 3 \\ -3 & 3 \end{bmatrix}$$

```
library(ggplot2)
## original coordinates
A <- matrix(c(3, -3), ncol = 1)
B <- matrix(c(3, 3), ncol = 1)

## coordinates for 4 copies/combos
copies <- -10:10
coefs <- expand.grid(copies, copies)

x.coords <- coefs$Var1*A[1] + coefs$Var2*B[1]
y.coords <- coefs$Var1*A[2] + coefs$Var2*B[2]

plot.dat <- data.frame(x = x.coords, y = y.coords)
p <- ggplot(data = plot.dat, aes(x = x, y = y)) +
    geom_point() +
    ylim(c(-14, 7)) +
    xlim(c(-7, 15)) +
    geom_rect(xmin = -7, xmax = 15, ymin = -14, ymax = 7,
                        fill = "#FFFFFF00", col = "black") +
    theme_minimal()


## could fix the grid to make these for printing!

## create on-diag rectangles
p1 <- p + geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                           ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                           fill = "lightpink", col = "pink") +
    ylim(c(-10, 7)) +
    xlim(c(-7, 15))
## create off-diag rectangles
p2 <- p1 + geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + B[1],
                           ymin = plot.dat$y, ymax = plot.dat$y - A[2],
                           fill = "lavender", col = "purple")
```

```r
p2 + geom_point(col = "magenta")
```

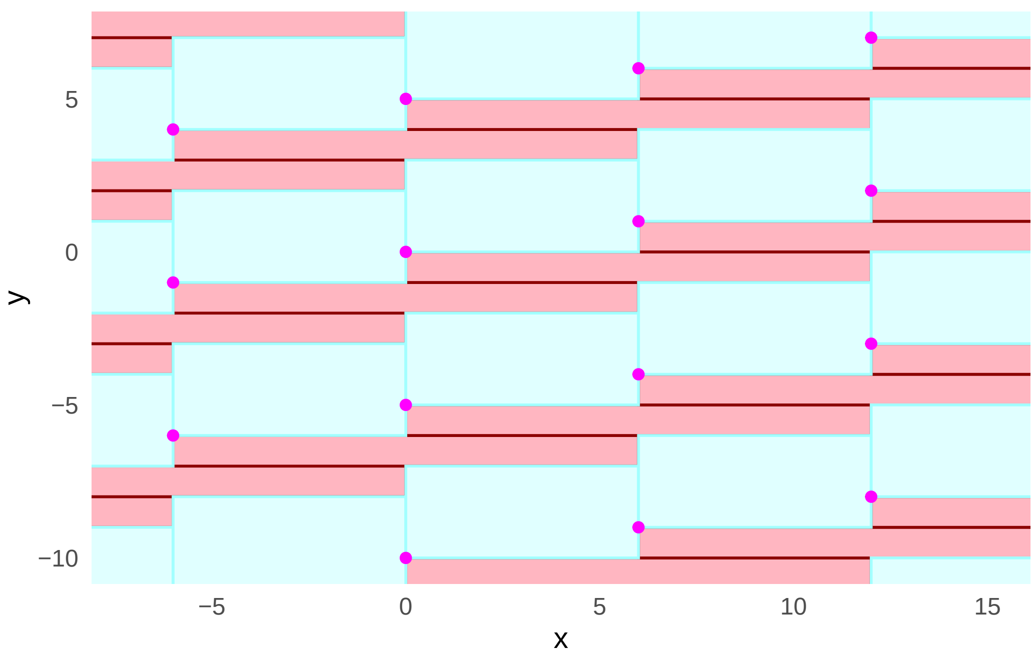


Fifth Tile

$$M = \begin{bmatrix} 12 & 6 \\ -3 & 1 \end{bmatrix}$$

```r
library(ggplot2)
## original coordinates
A <- matrix(c(12, -3), ncol = 1)
B <- matrix(c(6, 1), ncol = 1)

## coordinates for 4 copies/combos
copies <- -10:10
coefs <- expand.grid(copies, copies)

x.coords <- coefs$Var1*A[1] + coefs$Var2*B[1]
y.coords <- coefs$Var1*A[2] + coefs$Var2*B[2]

plot.dat <- data.frame(x = x.coords, y = y.coords)
p <- ggplot(data = plot.dat, aes(x = x, y = y)) +
```

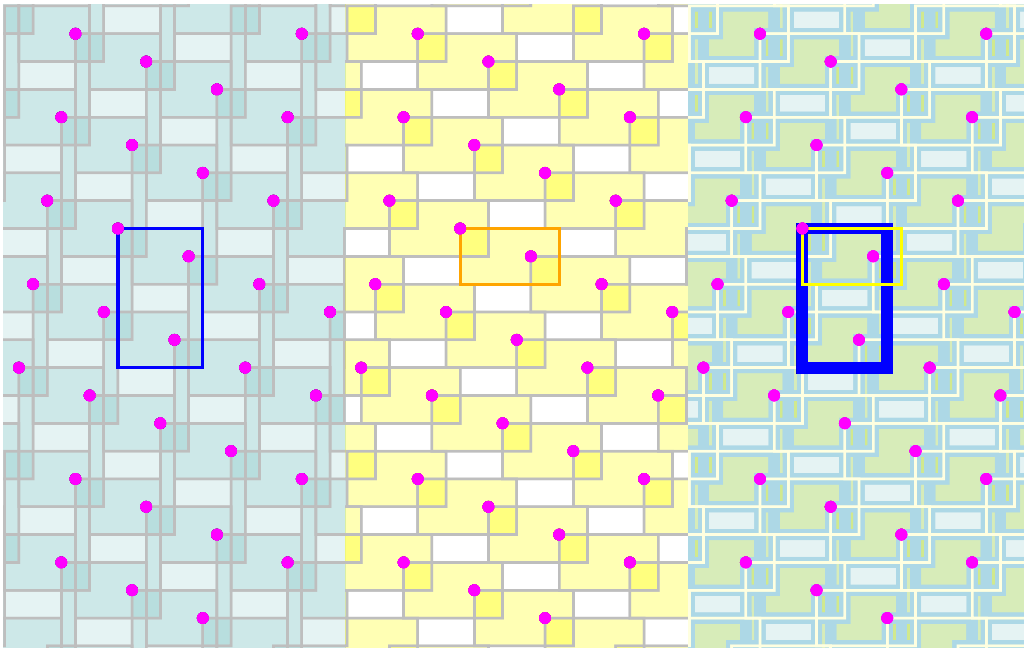
```r
    geom_point() +
    ylim(c(-14, 7)) +
    xlim(c(-7, 15)) +
    geom_rect(xmin = -7, xmax = 15, ymin = -14, ymax = 7,
                        fill = "#FFFFFF00", col = "black") +
    theme_minimal()


## could fix the grid to make these for printing!

## create on-diag rectangles
p1 <- p + geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                              ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                              fill = "lightpink", col = "darkred") +
    ylim(c(-10, 7)) +
    xlim(c(-7, 15))
## create off-diag rectangles
p2 <- p1 + geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + B[1],
                              ymin = plot.dat$y, ymax = plot.dat$y - A[2],
                              fill = "#E0FFFF", col = "#A2FFFF")
p2 + geom_point(col = "magenta")
```



Sixth tile

```
library(ggplot2)
## original coordinates
A <- matrix(c(7, 5), ncol = 1)
B <- matrix(c(6, 2), ncol = 1)

## coordinates for 4 copies/combos
copies <- -15:15
coefs <- expand.grid(copies, copies)

x.coords <- coefs$Var1*A[1] + coefs$Var2*B[1]
y.coords <- coefs$Var1*A[2] + coefs$Var2*B[2]

plot.dat <- data.frame(x = x.coords, y = y.coords)
p <- ggplot(data = plot.dat, aes(x = x, y = y)) +
    geom_point() +
    ylim(c(-14, 7)) +
    xlim(c(-7, 15)) +
    # geom_rect(xmin = -7, xmax = 15, ymin = -14, ymax = 7,
    #                   fill = "#FFFFFF00", col = "black") +
    theme_void()


## could fix the grid to make these for printing!

## create on-diag rectangles
p1 <- p +geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + B[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - A[2],
                            fill = "#008b8b", alpha = .1) +
                            geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + B[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - A[2],
                            fill = "#FFFFFF00", col = "grey") +
                            geom_rect(xmin = plot.dat$x[481], xmax = plot.dat$x[481] + B[1
                            ymin = plot.dat$y[481], ymax = plot.dat$y[481] - A[2],
                            fill = "#FFFFFF00", col = "blue") +
                            geom_point(col = "magenta") +
                            ylim(c(-14, 7)) +
                            xlim(c(-7, 15))
```

```
library(patchwork)
p2 <- p + geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                            fill = "yellow", alpha = .3) +
                            geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                            fill = "#FFFFFF00", col = "grey") +
                            geom_rect(xmin = plot.dat$x[481], xmax = plot.dat$x[481] + A[1
                            ymin = plot.dat$y[481], ymax = plot.dat$y[481] - B[2],
                            fill = "#FFFFFF00", col = "orange")+
                            geom_point(col = "magenta")+
                            ylim(c(-14, 7)) +
                            xlim(c(-7, 15))

## create off-diag rectangles
p_both <- p1 +  geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                            fill = "yellow", alpha = .2) +
                            ## outlines
                            geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + B[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - A[2],
                            fill = "#FFFFFF00", col = "lightblue", lwd = 2) +
                            geom_rect(xmin = plot.dat$x, xmax = plot.dat$x + A[1],
                            ymin = plot.dat$y, ymax = plot.dat$y - B[2],
                            fill = "#FFFFFF00", col = "lightyellow") +
                            geom_rect(xmin = plot.dat$x[481], xmax = plot.dat$x[481] + B[1
                            ymin = plot.dat$y[481], ymax = plot.dat$y[481] - A[2],
                            fill = "#FFFFFF00", col = "blue", lwd = 2) +
                            geom_rect(xmin = plot.dat$x[481], xmax = plot.dat$x[481] + A[1
                            ymin = plot.dat$y[481], ymax = plot.dat$y[481] - B[2],
                            fill = "#FFFFFF00", col = "yellow") +
                            geom_point(col = "magenta") +
                            ylim(c(-14, 7)) +
                            xlim(c(-7, 15))
## add edges after
p1 + p2 + p_both
```



1 Construction

1.1 Code outline

Input:

- A $(2+k) \times (2+k)$ matrix M

Step 1

- calculate k
- Create three new matrices:
  - M_{TOP} is the $2 \times (2+k)$ matrix consisting of the first two rows of M
  - M_{BOT} is the $k \times (2+k)$ matrix consisting of the remaining k rows of M
  - M' is the $(2+k) \times (2+k)$ matrix created by stacking M_{TOP} and $-M_{BOT}$

Step 2

- For each $\sigma \in \binom{[2+k]}{2}$, define $S_\sigma(M)$ as follows:

  > **Notation:** $\binom{[n]}{k}$
  >
  > The notation is shorthand for the set of all subsets of $\{1, 2, \ldots, n\}$ of size k.
  > For example, if $n = 3$ and $k = 2$,
  >
  > $$\binom{[n]}{k} = \binom{[3]}{2} = FIXME$$

  - Start with M',
    - Consider i^{th} column for $i \in \{1, 2, \ldots, k+2\}$. If :
      - $i \in \sigma$: zero out bottom k entries
      - $i \notin \sigma$: zero out the top 2 entries

- After this process, you will end up with $\frac{1}{2}k^2 + \frac{3}{2}k + 1$ matrices

$$M' = \begin{bmatrix} 2 & 3 & -4 & 2 & 5 \\ 6 & 1 & 8 & 0 & 4 \\ 3 & -3 & 4 & 6 & 0 \\ -5 & -5 & 4 & 2 & 3 \\ 1 & 8 & 1 & 2 & 2 \end{bmatrix}$$

$$S_{\{2,5\}} = \begin{bmatrix} 0 & 3 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & 4 & 0 & 0 \\ -5 & 0 & 4 & 0 & 3 \\ 1 & 0 & 1 & 0 & 2 \end{bmatrix}$$

Figure 1.1: Here, $k = 3$ and $\sigma = \{2, 5\}$.

Step 3

Consider the matrices $\{S_\sigma\}$. Each of these matrices corresponds to a $2 + k$-dimensional shape (the fundamental parallelepiped $\Pi\left(S_\sigma\right)$ defined by the columns of that matrix).

The fundamental parallelepiped $\Pi(S_\sigma)$

Note that

$$\Pi\left(S_\sigma\right) = \{S_\sigma \cdot (x_1, \ldots, x_{k+2})'\}$$

Think of the x_i as representing the interval $[0, 1]$, so that you are getting a shape. In other words, Π is the Minkowski sum of the columns of S_σ.

For each σ, want to visualize a collection of parallelepipeds, $\Pi\left(S_\sigma\right) + Mz$, where $z \in \mathbb{Z}^{k+2}$. (FIXME: in practice we won't want to generate individual plots because there will be too many, we just want to emphasize a given S_σ in the plots discussed next.)

The desired end product is to create two visualizations: one of all positive S_σ and one of all negative S_σ.

What about $det(S_\sigma) = 0$?

The determinant measures volume. So, in this case, the volume of the parallelepiped is 0, so there is nothing to visualize.

Also, that parallelepiped is more than 3D so there will need to be a choice made in which slices to take for visualization.