Examples

First Draft

In this file, I will try to run through some examples for ratio 1, 2, and 3 with our program so far. For outputs of ratio 2, the array represents different $kmax = n+1, \ldots$. For outputs of ratio 3, each line represents different values of e (the bound on p-valuation of the determinant of the solution), and for each e, the array represents the ratio for $k = 1, \ldots, n$ in order. The following runtime may be inaccurate as (for some unknown reason), running Julia in WSL (Windows Subsystem for Linux) can cause inconsistent runtime (as much as a factor of 3!). The examples are as follows,

1. $p = 2, n = 7, M' = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$.
   For $kmax = 8, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 3//2, 3//2, 3//2]
  0.204897 seconds (2.50 M allocations: 44.790 MiB)
ratio 3:
e = 0 → Rational{Int64}[1, 1, 1, 1, 1, 1, 1//2]
e = 1 → Rational{Int64}[2, 2, 2, 2, 2, 2, 3//2]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2]
  1.923861 seconds (4.88 M allocations: 374.310 MiB, 17.67% gc time, 18.23% compilation time)
```

2. $p = 2, n = 7, M' = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$.
   For $kmax = 8, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 11//8, 11//8, 11//8, 11//8, 11//8, 11//8]
  0.221524 seconds (2.51 M allocations: 50.418 MiB)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//2, 1//2, 1//2, 1//4, 1//8]
e = 1 → Rational{Int64}[2, 1, 1, 1, 1, 3//4, 3//8]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 3//2]
  1.543713 seconds (3.95 M allocations: 346.930 MiB, 16.42% gc time, 22.48% compilation time)
```

3. $p = 2, n = 8, M' = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$.
   For $kmax = 9, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 11//8, 11//8, 11//8, 11//8, 11//8]
  0.794720 seconds (2.78 M allocations: 55.421 MiB, 54.67% gc time)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//2, 1//2, 1//2, 1//2, 1//4, 1//8]
e = 1 → Rational{Int64}[2, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 3//2]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 7//4]
  7.488949 seconds (20.02 M allocations: 1.750 GiB, 22.42% gc time)
```

4. $p = 2, n = 9, M' = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$.
   For $kmax = 10, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 11//8, 11//8, 11//8, 11//8]
  0.104838 seconds (3.08 M allocations: 95.800 MiB)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//4, 1//8]
e = 1 → Rational{Int64}[2, 1, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 3//2]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 7//4]
 16.974221 seconds (80.05 M allocations: 7.025 GiB, 32.06% gc time)
```

5. $p = 2, n = 10, M' = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$.
   For $kmax = 11, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 11//8, 11//8, 11//8]
  0.488224 seconds (3.57 M allocations: 108.721 MiB, 51.89% compilation time)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//4, 1//8]
e = 1 → Rational{Int64}[2, 1, 1, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 3//2]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 7//4]
 76.405462 seconds (320.51 M allocations: 28.136 GiB, 30.85% gc time, 0.45% compilation time)
```

6. $p = 2, n = 11, M' = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}$.
   For $kmax = 12, \ldots, 15$, we get

```
julia> @time compute_ratio12(2, 11, 6, 5, 15); @time compute_ratio3(2, 11, (1, 0, 0, 5))
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 11//8, 11//8]
  0.226014 seconds (3.72 M allocations: 216.782 MiB)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 1//4, 1//8]
e = 1 → Rational{Int64}[2, 1, 1, 1, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 2 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3//2]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 7//4]
358.527506 seconds (1.28 G allocations: 112.536 GiB, 32.57% gc time)
```

7. $p = 2, n = 10, M' = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$.
   For $kmax = 11, \ldots, 15$, we get

```
julia> @time compute_ratio12(2, 10, 10, 9, 15); @time compute_ratio3(2, 10, (1, 0, 0, 9))
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 43//32, 43//32, 43//32]
  1.347063 seconds (3.44 M allocations: 192.650 MiB, 70.82% gc time)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//4, 1//4, 1//4, 1//4, 1//4, 1//8, 1//16, 1//32]
e = 1 → Rational{Int64}[2, 1, 1//2, 1//2, 1//2, 1//2, 1//2, 3//8, 3//16, 3//32]
e = 2 → Rational{Int64}[2, 2, 1, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 15//8, 23//16]
 81.096195 seconds (278.23 M allocations: 27.033 GiB, 31.09% gc time)
```

8. $p = 2, n = 11, M' = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$.
   For $kmax = 12, \ldots, 15$, we get

```
ratio 1: 2//1
ratio 2: Rational{Int64}[3//2, 11//8, 43//32, 43//32]
  0.248298 seconds (3.86 M allocations: 192.079 MiB)
ratio 3:
e = 0 → Rational{Int64}[1, 1//2, 1//4, 1//4, 1//4, 1//4, 1//4, 1//4, 1//8, 1//16, 1//32]
e = 1 → Rational{Int64}[2, 1, 1//2, 1//2, 1//2, 1//2, 1//2, 1//2, 3//8, 3//16, 3//32]
e = 2 → Rational{Int64}[2, 2, 1, 1, 1, 1, 1, 1, 1, 3//4, 3//8]
e = 3 → Rational{Int64}[2, 2, 2, 2, 2, 2, 2, 2, 2, 15//8, 23//16]
339.682632 seconds (1.11 G allocations: 108.158 GiB, 28.39% gc time)
```

9. $p = 3, n = 6, M' = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$.
   For $kmax = 7, \ldots, 12$, we get

```
julia> @time compute_ratio12(3, 6, 5, 4, 12); @time compute_ratio3(3, 6, (1, 0, 0, 4))
ratio 1: 3//2
ratio 2: Rational{Int64}[7//6, 7//6, 7//6, 7//6, 7//6, 7//6]
  2.191727 seconds (21.09 M allocations: 362.384 MiB, 41.41% gc time)
ratio 3:
e = 0 → Rational{Int64}[1//2, 1//2, 1//2, 1//2, 1//2, 1//6]
e = 1 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 7//6]
e = 2 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 3//2]
e = 3 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 3//2]
 29.607283 seconds (129.95 M allocations: 11.200 GiB, 32.86% gc time)
```

10. $p = 3, n = 6, M' = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$.
    For $kmax = 7, \ldots, 12$, we get

```
julia> @time compute_ratio12(3, 6, 11, 10, 12); @time compute_ratio3(3, 6, (1, 0, 0, 10))
ratio 1: 3//2
ratio 2: Rational{Int64}[7//6, 61//54, 61//54, 61//54, 61//54, 61//54]
  1.338297 seconds (21.27 M allocations: 360.481 MiB, 24.64% gc time)
ratio 3:
e = 1 → Rational{Int64}[3//2, 1//2, 1//2, 1//2, 7//18, 7//54]
e = 2 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 7//6]
 14.071798 seconds (54.52 M allocations: 5.431 GiB, 33.14% gc time)
```

11. $p = 3, n = 7, M' = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$.
For $kmax = 8, \ldots, 12$, we get

```
julia> @time compute_ratio12(3, 7, 11, 10, 12); @time compute_ratio3(3, 7, (1, 0, 0, 10))
ratio 1: 3//2
ratio 2: Rational{Int64}[7//6, 61//54, 61//54, 61//54, 61//54]
  3.058086 seconds (24.41 M allocations: 496.669 MiB, 43.92% gc time)
ratio 3:
e = 1 → Rational{Int64}[3//2, 1//2, 1//2, 1//2, 1//2, 7//18, 7//54]
e = 2 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 3//2, 7//6]
134.824826 seconds (490.85 M allocations: 48.683 GiB, 28.83% gc time)
```

12. $p = 3, n = 7, M' = \begin{pmatrix} 1 & 0 \\ 0 & 28 \end{pmatrix}$.
For $kmax = 8, \ldots, 12$, we get

```
ratio 1: 3//2
ratio 2: Rational{Int64}[7//6, 61//54, 547//486, 547//486, 547//486]
  2.819348 seconds (24.47 M allocations: 697.799 MiB, 40.05% gc time)
ratio 3:
e = 1 → Rational{Int64}[3//2, 1//2, 1//6, 1//6, 7//54, 7//162, 7//486]
e = 2 → Rational{Int64}[3//2, 3//2, 1//2, 1//2, 1//2, 7//18, 7//54]
e = 3 → Rational{Int64}[3//2, 3//2, 3//2, 3//2, 3//2, 79//54, 187//162]
185.602098 seconds (667.66 M allocations: 70.430 GiB, 25.88% gc time, 0.08% compilation time)
```

13. $p = 3, n = 7, M' = \begin{pmatrix} 1 & 6 \\ 3 & 1 \end{pmatrix}$.
For $kmax = 8, \ldots, 12$, we get

```
julia> @time compute_ratio12(3, 7, 2, -17, 13); @time compute_ratio3(3, 7, (1, 6, 3, 1))
ratio 1: 5//4
ratio 2: Rational{Int64}[13//12, 13//12, 13//12, 13//12, 13//12, 13//12]
  6.089627 seconds (63.18 M allocations: 971.752 MiB, 40.53% gc time)
ratio 3:
e = 0 → Rational{Int64}[1//4, 1//4, 1//4, 1//4, 1//4, 1//4, 1//12]
e = 1 → Rational{Int64}[5//4, 5//4, 5//4, 5//4, 5//4, 5//4, 13//12]
e = 2 → Rational{Int64}[5//4, 5//4, 5//4, 5//4, 5//4, 5//4, 5//4]
234.993731 seconds (746.23 M allocations: 63.090 GiB, 31.04% gc time)
```

14. $p = 3, n = 7, M' = \begin{pmatrix} 1 & 18 \\ 9 & 1 \end{pmatrix}$.
For $kmax = 8, \ldots, 12$, we get

```
julia> @time compute_ratio12(3, 7, 2, -143, 13); @time compute_ratio3(3, 7, (1, 18, 9, 1))
ratio 1: 3//2
ratio 2: Rational{Int64}[7//6, 7//6, 7//6, 7//6, 7//6, 7//6]
 15.672939 seconds (71.52 M allocations: 1.117 GiB, 32.67% gc time)
ratio 3:
e = 0 → Rational{Int64}[5//12, 1//12, 1//12, 1//12, 1//12, 1//36, 1//108]
e = 1 → Rational{Int64}[17//12, 5//12, 5//12, 5//12, 5//12, 13//36, 13//108]
e = 2 → Rational{Int64}[17//12, 17//12, 17//12, 17//12, 17//12, 17//12, 41//36]
408.090835 seconds (695.21 M allocations: 68.204 GiB, 23.15% gc time)
```

15. $p = 5, n = 5, M' = \begin{pmatrix} 1 & 0 \\ 0 & 6 \end{pmatrix}$.
    For $kmax = 6, \ldots, 9$, we get

```
ratio 1: 5//4
ratio 2: Rational{Int64}[21//20, 21//20, 21//20, 21//20]
 12.574975 seconds (47.73 M allocations: 826.920 MiB, 37.25% gc time, 4.95% compilation time)
ratio 3:
e = 0 → Rational{Int64}[1//4, 1//4, 1//4, 1//4, 1//20]
e = 1 → Rational{Int64}[5//4, 5//4, 5//4, 5//4, 21//20]
488.282807 seconds (901.64 M allocations: 85.564 GiB, 29.30% gc time, 0.17% compilation time)
```

16. $p = 5, n = 5, M' = \begin{pmatrix} 1 & 10 \\ 5 & 1 \end{pmatrix}$.
    For $kmax = 6, \ldots, 9$, we get

```
julia> @time compute_ratio12(5, 5, 2, -49, 9); @time compute_ratio3(5, 5, (1, 10, 5, 1))
ratio 1: 7//6
ratio 2: Rational{Int64}[31//30, 31//30, 31//30, 31//30]
 16.534435 seconds (44.76 M allocations: 700.388 MiB, 41.18% gc time)
ratio 3:
e = 0 → Rational{Int64}[1//6, 1//6, 1//6, 1//6, 1//30]
e = 1 → Rational{Int64}[7//6, 7//6, 7//6, 7//6, 31//30]
593.111332 seconds (857.06 M allocations: 78.484 GiB, 28.04% gc time)
```