Pensieve header: Roland’s notebook on Mat/Weave knots.

In[ ]:=
```
(*Please ignore*)
CF[ℰ_] := Module[{vs = Union@Cases[ℰ, g__, ∞], ps, c},
   Total[CoefficientRules[Expand[ℰ], vs] /. (ps_ → c_) ⧴ Factor[c] (Times @@ vs^ps)] ];
```

In[ ]:=
```
(*The formulas*)
T_3 = T_1 T_2;
R_1[s_, i_, j_] =
  CF[s (1 / 2 - g_3ii + T_2^s g_1ii g_2ji - g_1ii g_2jj - (T_2^s - 1) g_2ji g_3ii + 2 g_2jj g_3ii - (1 - T_3^s) g_2ji g_3ji -
      g_2ii g_3jj - T_2^s g_2ji g_3jj + g_1ii g_3jj + ((T_1^s - 1) g_1ji (T_2^(2 s) g_2ji - T_2^s g_2jj + T_2^s g_3jj) +
          (T_3^s - 1) g_3ji (1 - T_2^s g_1ii - (T_1^s - 1) (T_2^s + 1) g_1ji + (T_2^s - 2) g_2jj + g_2ij)) / (T_2^s - 1))];
θ[{s0_, i0_, j0_}, {s1_, i1_, j1_}] := CF[s1 (T_1^s0 - 1) (T_2^s1 - 1)^-1
    (T_3^s1 - 1) g_{1,j1,i0} g_{3,j0,i1} ( (T_2^s0 g_{2,i1,i0} - g_{2,i1,j0}) - (T_2^s0 g_{2,j1,i0} - g_{2,j1,j0}))]
Γ_1[φ_, k_] = -φ / 2 + φ g_3kk;
```

In[ ]:=
```
(*The main program*)
θ[K_PD] := θ[Rot[K]]
θ[{Cs_, φ_}] := Module[{n, A, s, i, j, k, Δ, G, ν, α, β, gEval, c, z},
    n = Length[Cs];
    A = IdentityMatrix[2 n + 1];
    Cases[Cs, {s_, i_, j_} ⧴ (A[[{i, j}, {i + 1, j + 1}]] += ( -T^s  T^s - 1 
                                                              0     -1     ))];
    Δ = T^((-Total[φ]-Total[Cs[[All,1]]])/2) Det[A];
    G = Inverse[A];
    gEval[ℰ_] := Factor[ℰ /. g_{ν_,α_,β_} ⧴ (G[[α, β]] /. T → T_ν)];
    z = gEval[Sum_{k1=1}^n Sum_{k2=1}^n θ[Cs[[k1]], Cs[[k2]]]];
    z += gEval[Sum_{k=1}^n R_1 @@ Cs[[k]]];
    z += gEval[Sum_{k=1}^{2 n} Γ_1[φ[[k]], k]];
    {Δ, (Δ /. T → T_1) (Δ /. T → T_2) (Δ /. T → T_3) z} // Factor ];
```

```
(*Example calculation*)
ThetaTrefoil = θ[{{{1, 1, 4}, {1, 5, 2}, {1, 3, 6}}, {0, 0, 0, -1, 0, 0, 0}}]
```

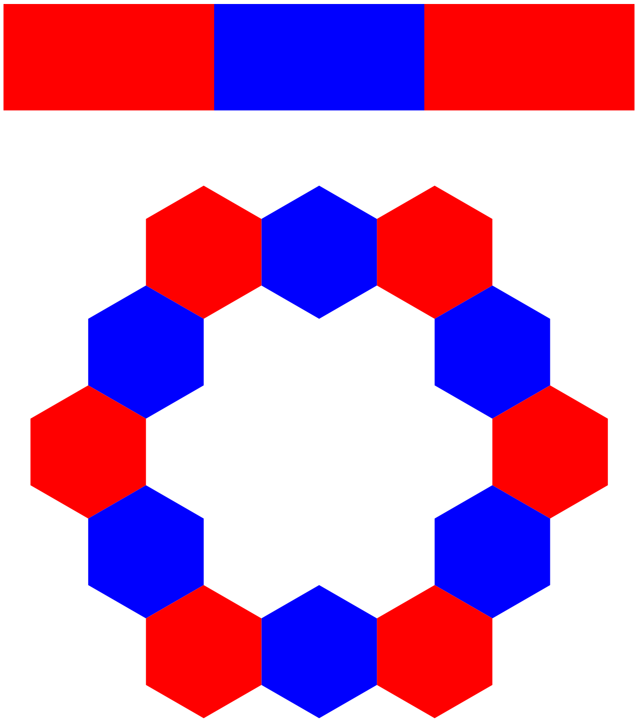
Out[ ]=

$$\left\{\frac{1-T+T^2}{T},\ \frac{1-T_1+T_1^2-T_2-T_1^3T_2+T_2^2+T_1^4T_2^2-T_1T_2^3-T_1^4T_2^3+T_1^2T_2^4-T_1^3T_2^4+T_1^4T_2^4}{T_1^2T_2^2}\right\}$$

In[∘]:=
```
(*For plotting the output nicely*)
PolyPlot[{Δ_, θ_}] :=
  Module[{crs, m, m1, m2, maxc, minc, s, , rect, hex}, GraphicsColumn[{
     rect = {{0, 0}, {1, 0}, {1, 1}, {0, 1}};
     hex = Table[{Cos[α], Sin[α]} / Cos[2 π / 12] / 2, {α, 2 π / 12, 2 π, 2 π / 6}];
     If[Expand[Δ] === 0, Graphics[],
      crs = CoefficientRules[T^(m=-Exponent[Δ,T,Min]) Δ, {T}];
      maxc = N@Log@Max@Abs[Last /@ crs];
      minc = N@Log@Min@Select[Abs[Last /@ crs], # > 0 &];
      If[minc == maxc, s[_] = 0, s[c_] := s[c] = (maxc - Log@c) / (maxc - minc)];
      Graphics[crs /. ({x_} → c_) :→ {
          Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
          Polygon[({x + m - 1 / 2, 0} + #) & /@ rect] }, AspectRatio → 1 / 5 ]
     ],
     If[Expand[θ] === 0, Graphics[{White, Disk[]}],
      crs = CoefficientRules[T1^(m1=-Exponent[θ,T1,Min]) T2^(m2=-Exponent[θ,T2,Min]) θ, {T1, T2}];
      maxc = N@Log@Max@Abs[Last /@ crs];
      minc = N@Log@Min@Select[Abs[Last /@ crs], # > 0 &];
      If[minc == maxc, s[_] = 0, s[c_] := s[c] = (maxc - Log@c) / (maxc - minc)];
      Graphics[{{White, Disk[{0, 0}, 1 + Cos[2 π / 12] Norm[{m1, m2}] / Sqrt[2]]},
        crs /. ({x1_, x2_} → c_) :→ {
           Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
           Polygon[(({{1, -1 / 2}, {0, Sqrt[3] / 2}}).{x1 - m1, x2 - m2} + #) & /@ hex] }
        }]
     ]
    }, Spacings → 0]];
```

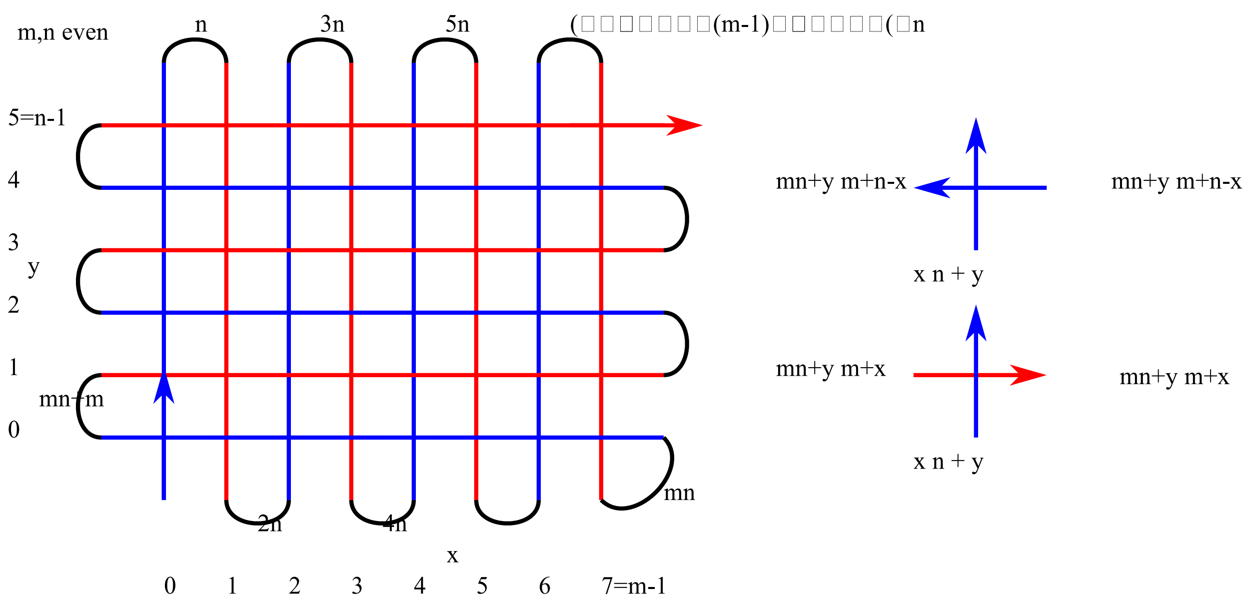
In[•]:=
```
PolyPlot[ThetaTrefoil]
```

Out[•]=



## Some more complicated knots

A mat-shaped knot based on an m by n rectangular grid. The integers m,n should be even and the crossings can be chosen arbitrarily.
Probably all knots can be brought into this form.



```
In[•]:= PositiveQ[X[a_, b_, c_, d_]] := If[b < d, False, True]
Rot[pd_PD] := Module[{n, xs, x, rots, Xp, Xm, front = {1}, k},
   n = Length@pd; rots = Table[0, {2 n}];
   xs = Cases[pd, x_X :> { Xp[x⟦4⟧, x⟦1⟧]  PositiveQ@x
                          Xm[x⟦2⟧, x⟦1⟧]  True        ];
   For[k = 1, k ≤ 2 n, ++k,
    If[FreeQ[front, -k],
     front = Flatten@Replace[front, k → (xs /. {
            Xp[k, l_] | Xm[l_, k] :> {l + 1, k + 1, -l},
            Xp[l_, k] | Xm[k, l_] :> (++rots⟦l⟧; {-l, k + 1, l + 1}),
            _Xp | _Xm :> {}
           }), {1}],
     Cases[front, k | -k] /. {k, -k} :> --rots⟦k⟧;
    ]
   ];
   {xs /. {Xp[i_, j_] :> {+1, i, j}, Xm[i_, j_] :> {-1, i, j}}, rots} ];
Rot[K_] := Rot[PD[K]];
```

```
In[•]:= (*m,n, even, by default all xings positive.  The
 list negxs flips the sign of those crossings.*)
Mat[m_, n_, negxs_ : {}] := Module[{pd}, pd = (Flatten@Table[
        Switch[
         {EvenQ[x], EvenQ[y]},
         {True, True},
         X[m n + y m + m - x - 1, n x + y + 1, m n + y m + m - x, n x + y],
         {True, False},
         X[n x + y, m n + y m + x + 1, n x + y + 1, m n + y m + x],
         {False, True},
         X[n x + n - y - 1, m n + y m + m - x, n x + n - y, m n + y m + m - x - 1],
         {False, False},
         X[m n + y m + x, n x + n - y, m n + y m + x + 1, n x + n - y - 1]
        ]
        , {x, 0, m - 1}, {y, 0, n - 1}]
     ) /. {X[a_, b_, c_, d_] :> X[a + 1, b + 1, c + 1, d + 1]};
   Do[pd⟦i⟧ = (pd⟦i⟧ /. {X[a_, b_, c_, d_] :> X[d, a, b, c]}), {i, negxs}];
   PD @@ pd
  ]
(*Randomly assign signs to all crossings*)
RandMat[m_, n_] := Mat[m, n, RandomSample[Range[m n], RandomInteger[{0, m n}]]]
```

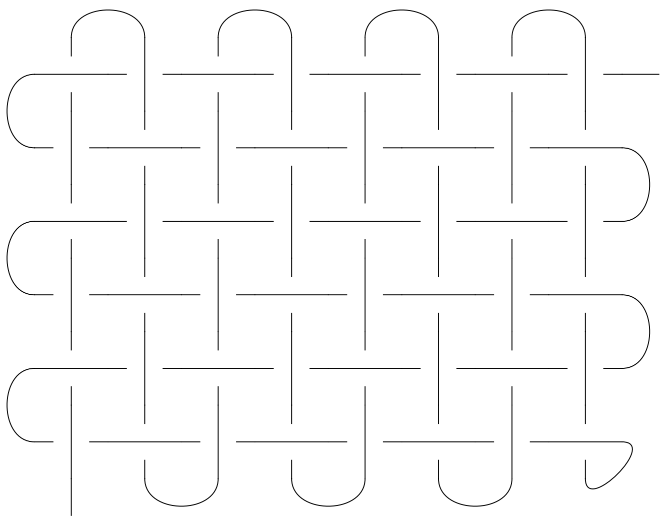
In[∘]:=
```
DrawXing[x_, y_, cut_] :=
 If[cut === "v", {Line[{{x, y - 1/2}, {x, y - 1/4}}], Line[{{x, y + 1/4}, {x, y + 1/2}}],
   Line[{{x - 1/2, y}, {x + 1/2, y}}]}, {Line[{{x, y - 1/2}, {x, y + 1/2}}],
   Line[{{x - 1/2, y}, {x - 1/4, y}}], Line[{{x + 1/4, y}, {x + 1/2, y}}]}]
```

In[ ]:=
```
DrawMat[m_, n_, negxs_ : {}] := Graphics[{
   Table[
    Switch[
     {EvenQ[x], EvenQ[y]},
     {True, True},
     DrawXing[x, y, If[MemberQ[negxs, y m + x + 1], "v", "h"]],
     {True, False},
     DrawXing[x, y, If[MemberQ[negxs, y m + x + 1], "h", "v"]],
     {False, True},
     DrawXing[x, y, If[MemberQ[negxs, y m + x + 1], "h", "v"]],
     {False, False},
     DrawXing[x, y, If[MemberQ[negxs, y m + x + 1], "v", "h"]]
    ]
    , {x, 0, m - 1}, {y, 0, n - 1}],
   Table[{Arrowheads[{{0, .7}}],
     Arrow[BezierCurve[{{x, -1/2}, {x, -1}, {x + 1, -1}, {x + 1, -1/2}}]]}, {x, 1, m - 3, 2}],
   Table[{Arrowheads[{{0, .7}}], Arrow[
      BezierCurve[{{x, n - 1/2}, {x, n}, {x + 1, n}, {x + 1, n - 1/2}}]]}, {x, 0, m - 2, 2}],
   Table[{Arrowheads[{{0, .7}}],
     Arrow[BezierCurve[{{-1/2, y}, {-1, y}, {-1, y + 1}, {-1/2, y + 1}}]]}, {y, 0, n - 2, 2}],
   Table[{Arrowheads[{{0, .7}}], Arrow[
      BezierCurve[{{m + -1/2, y}, {m, y}, {m, y + 1}, {m + -1/2, y + 1}}]]}, {y, 1, n - 3, 2}],
   BezierCurve[{{m - 1, -1/2}, {m - 1, -1}, {m, 0}, {m - 1/2, 0}}],
   {Arrowheads[{{0, 1}}], Arrow[{{m - 1/2, n - 1}, {m, n - 1}}]},
   {Arrowheads[{{0, 0.5}}], Arrow[{{0, -1}, {0, -1/2}}]}

  }]
DrawRandMat[m_, n_] := DrawMat[m, n, RandomSample[Range[m n], RandomInteger[m n]]]
```

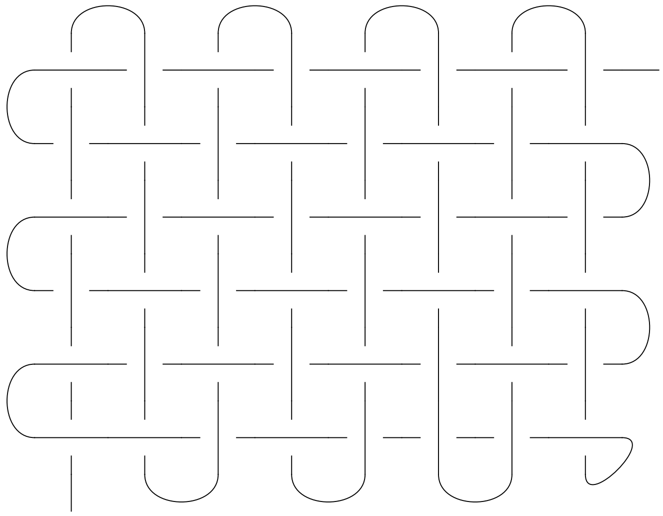
In[∘]:= **DrawMat[8, 6]**

Out[∘]=

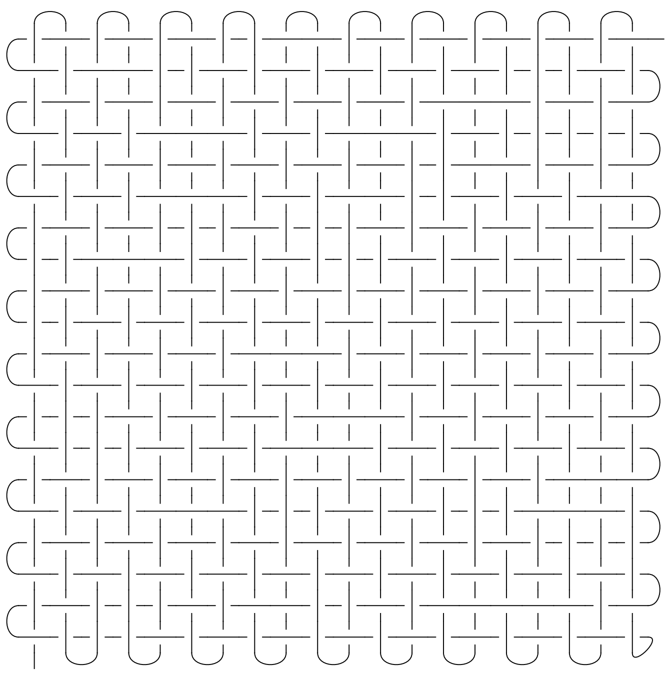


In[•]:=
```
DrawMat[8, 6, {1, 6}]
DrawRandMat[20, 20]
```

Out[•]=

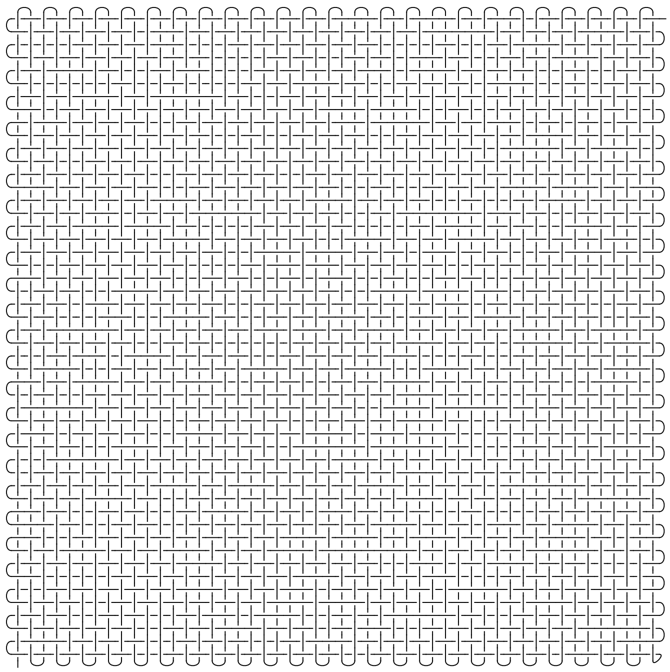


Out[•]=



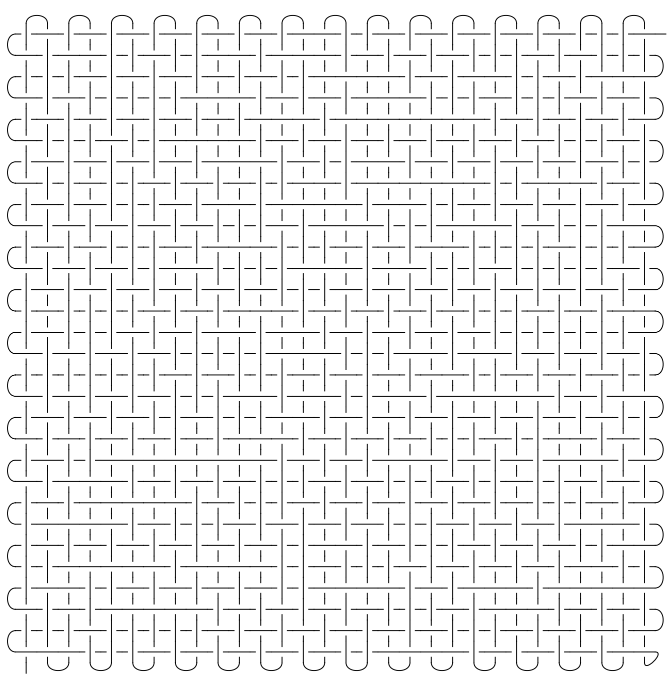
In[ ]:= **DrawRandMat[50, 50]**

Out[ ]=



In[ ]:= **DrawRandMat[30, 30]**

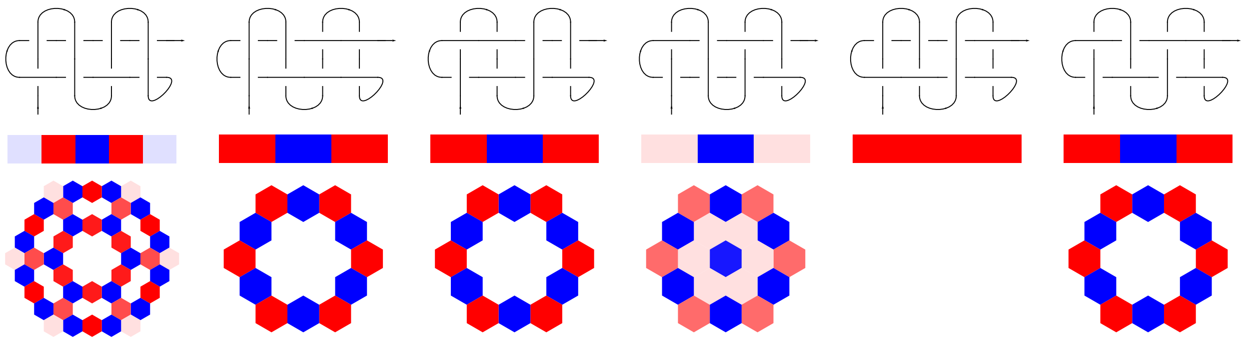
Out[ ]=



In[∘]:=
```
RandMatPair[m_, n_] := Module[{nxs = RandomSample[Range[m n], RandomInteger[m n]]},
  {DrawMat[m, n, nxs], PolyPlot@θ@Mat[m, n, nxs]}
 ]
```

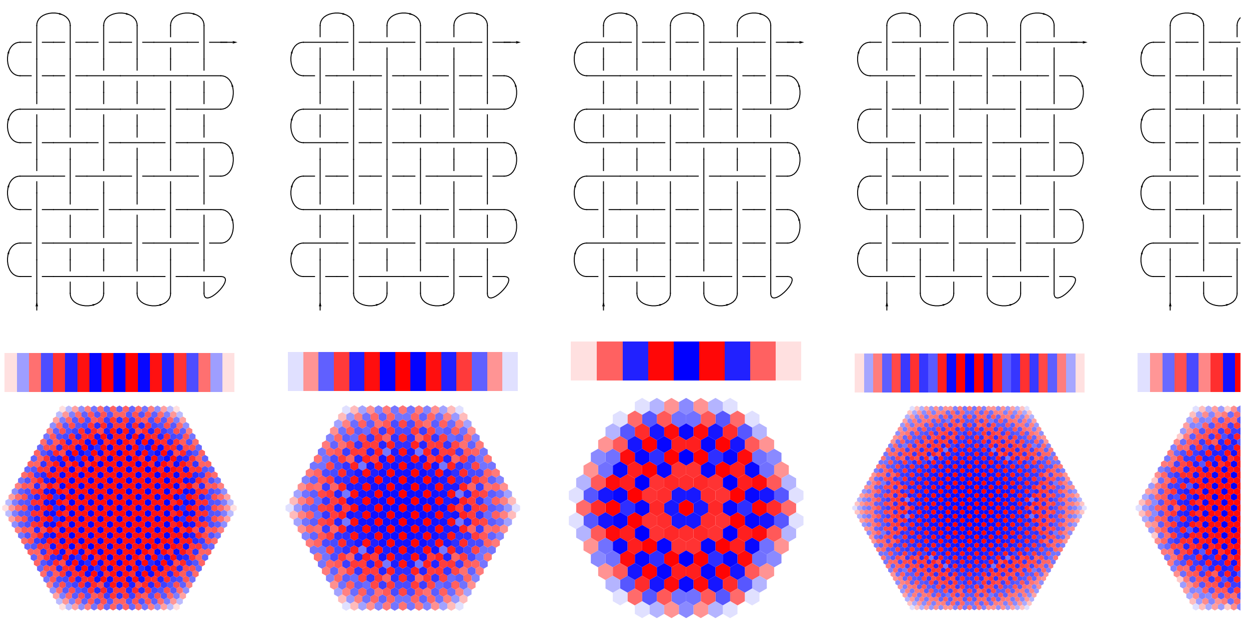
In[∘]:=
```
GraphicsGrid[Table[RandMatPair[4, 2], {j, 7}] // Transpose]
```

Out[∘]=



In[∘]:=
```
GraphicsGrid[Table[RandMatPair[6, 8], {j, 7}] // Transpose]
```

Out[∘]=



```
(*The knots Mat[m,n] are both alternating and also positive,
and thus fibered. This seems to make their theta look especially simple. The
 sides of the hexagon (i.e. the coefficient of the maximal power of T2) are
 in this case conjectured to be always equal to half the signature times
 the Alexander polynomial in T1. Below we print the first few square ones.*)
```

In[ ]:= 
```
GraphicsGrid@Table[{DrawMat[n, n], PolyPlot@θ@Mat[n, n]}, {n, 2, 10, 2}]
```

Out[ ]=