

Pensieve header: This is the main Mathematica package that goes along with the paper “A Fast, Strong, Topologically Meaningful and Fun Knot Invariant” by Dror Bar-Natan and Roland van der Veen.

```
In[*]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\Theta"];
```

tex

We start by loading the package `\verb$KnotTheory`$` --- it is only needed because it has many specific knots pre-defined. In this Section and in the next, `$\face$` and `\scalebox{0.66}{$\human$}` mean ``human input” while `$\machine$` means ``computer output”:

```
{\def\nbpdfPostInput{\hfill}
```

pdf

```
In[*]:= Once[<< KnotTheory`]
```

pdf

Loading KnotTheory` version of October 29, 2024, 10:29:52.1301.
Read more at <http://katlas.org/wiki/KnotTheory>.

tex

```
}
```

Next we quietly define the modules `\verb$Rot$`, used to compute rotation numbers, and `\verb$PolyPlot$`, used to plot polynomials as bar codes and as hexagonal QR codes. Neither is a part of the core of the computation of `$\Theta$`, so neither is shown; yet we do show one usage example for each.

pdf

```
In[*]:= (* The definitions of Rot and PolyPlot are suppressed *)
```

tex

```
{\def\nbpdfPostInput{\hfill}
```

```
In[*]:= PD[epd_EPD] := PD@@epd /. {Xi,j => X[j, i + 1, j + 1, i],  $\bar{X}_{i,j}$  => X[j, i, j + 1, i + 1]}
```

```

In[ ]:= Rot[{X_,  $\varphi$ _}] := {X,  $\varphi$ };
Rot[pd_PD] := Module[{n, xs, x, rots, Xp, Xm, front = {1}, k},
  n = Length@pd; rots = Table[0, {2 n + 1}];
  xs = Cases[pd, x_X  $\Rightarrow$  {Xp[x[[4]], x[[1]] PositiveQ@x},
    {Xm[x[[2]], x[[1]]] True}];
  For[k = 1, k  $\leq$  2 n, ++k,
    If[FreeQ[front, -k],
      front = Flatten@Replace[front, k  $\rightarrow$  {xs /. {
        Xp[k, l_] | Xm[l_, k]  $\Rightarrow$  {l + 1, k + 1, -l},
        Xp[l_, k] | Xm[k, l_]  $\Rightarrow$  {++rots[[l]]; {-l, k + 1, l + 1}},
        _Xp | _Xm  $\Rightarrow$  {}
      }}, {1}],
      Cases[front, k | -k] /. {k, -k}  $\Rightarrow$  --rots[[k]];
    ]
  ];
  {xs /. {Xp[i_, j_]  $\Rightarrow$  {+1, i, j}, Xm[i_, j_]  $\Rightarrow$  {-1, i, j}}, rots}];
Rot[K_] := Rot[PD[K]];

```

pdf

```
In[ ]:= Rot[Mirror@Knot[3, 1]]
```

pdf

 KnotTheory: Loading precomputed data in PD4Knots`.

Out[]:=

pdf

```
{{{1, 1, 4}, {1, 3, 6}, {1, 5, 2}}, {0, 0, 0, -1, 0, 0, 0}}
```

tex

```
}
```

We urge the reader to compare the above output with the knot diagram in Figure~\ref{fig:SampleDiagram}.

```

In[ ]:= Options[PolyPlot1] = {Labeled  $\rightarrow$  False};
PolyPlot1[ $\Delta$ _, OptionsPattern[]] := Module[{ $\Delta$ 1, crs, m, maxc, minc, s, rect},
   $\Delta$ 1 = PowerExpand@Expand@ $\Delta$ ;
  rect = {{0, 0}, {1, 0}, {1, 1}, {0, 1}};
  If[ $\Delta$ 1 === 0, Graphics[],
    m = Max[-Exponent[ $\Delta$ 1, T, Min], Exponent[ $\Delta$ 1, T, Max]];
    crs = CoefficientRules[Tm  $\Delta$ 1, {T}];
    maxc = N@Log@Max@Abs[Last /@ crs];
    minc = N@Log@Min@Select[Abs[Last /@ crs], # > 0 &];
    If[minc == maxc, s[_] = 0, s[_] := s[_] = (maxc - Log@c) / (maxc - minc)];
    Graphics[crs /. ({x_}  $\rightarrow$  c_)  $\Rightarrow$  {
      Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
      Tooltip[Polygon[{(x + m - 1 / 2, 0) + #} & /@ rect], c Tx-m],
      If[Not@OptionValue[Labeled], {}],
    }];

```

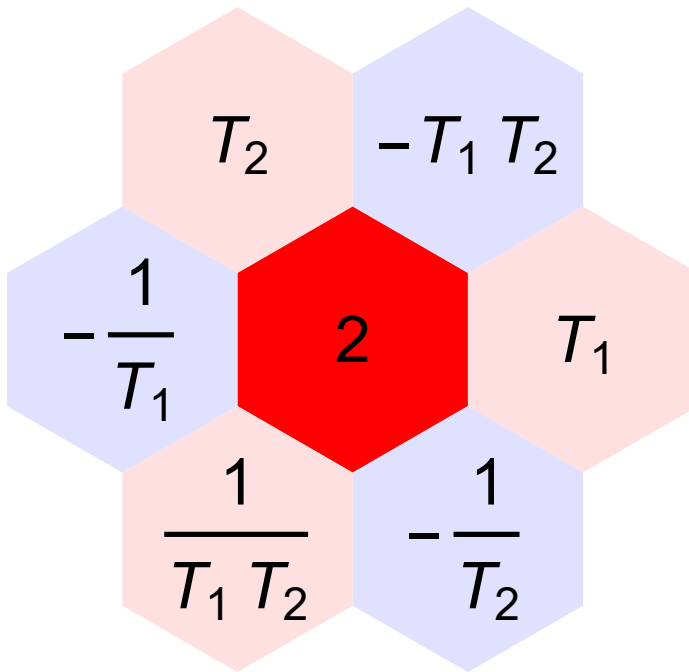
```

      {Black, FontSize → Scaled[ $\frac{1}{(m+1)(6+maxc)}$ ], Text[c Tx-m, {x+m, 0.5}]}]]
    }, AspectRatio → Min[1/5, 1/√(m+1)],
    ImagePadding → None, PlotRangePadding → None]
  ]];
Options[PolyPlot2] = {Labeled → False};
PolyPlot2[ $\theta$ _, OptionsPattern[]] := Module[{ $\theta$ 1, crs, m1, m2, maxc, minc, s, hex, p},
   $\theta$ 1 = PowerExpand@Expand@ $\theta$ ;
  If[ $\theta$ 1 === 0, Graphics[{White, Disk[]}],
  hex = Table[{Cos[ $\alpha$ ], Sin[ $\alpha$ ]} / Cos[2  $\pi$  / 12] / 2, { $\alpha$ , 2  $\pi$  / 12, 2  $\pi$ , 2  $\pi$  / 6}];
  m1 = Max[-Exponent[ $\theta$ 1, T1, Min], Exponent[ $\theta$ 1, T1, Max]];
  m2 = Max[-Exponent[ $\theta$ 1, T2, Min], Exponent[ $\theta$ 1, T2, Max]];
  crs = CoefficientRules[T1m1 T2m2  $\theta$ 1, {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[{(*{Yellow, Disk[{0, 0}, 1 + Cos[2  $\pi$  / 12] Norm[{m1, m2}] / √2]}), (*
    crs /. (({x1_, x2_} → c_) => {
      Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
      p =  $\begin{pmatrix} 1 & -1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix} \cdot \{x1 - m1, x2 - m2\}$ ;
      Tooltip[Polygon[(p + #) & /@ hex], c T1x1-m1 T2x2-m2],
      If[Not@OptionValue[Labeled], {},
        {Black, FontSize → Scaled[ $\frac{1}{\text{Max}[m1, m2] (10 + maxc)}$ ], Text[c T1x1-m1 T2x2-m2, p]}]}
    }, ImagePadding → None, PlotRangePadding → None]
  ]];
Options[PolyPlot] = {Labeled → False, ImageSize → Automatic};
PolyPlot[{ $\Delta$ _,  $\theta$ _, opts__Rule] := GraphicsColumn[
  {PolyPlot1[ $\Delta$ , FilterRules[Join[{opts}, Options[PolyPlot]], Options[PolyPlot1]]],
  PolyPlot2[ $\theta$ , FilterRules[Join[{opts}, Options[PolyPlot]], Options[PolyPlot2]]]},
  Spacings → Scaled@0.08, ImagePadding → None, PlotRangePadding → None,
  FilterRules[Join[{opts}, Options[PolyPlot]], Options[GraphicsColumn]]
];

```

In[]:= PPDemo = PolyPlot2[2 + T₁ - T₁ T₂ + T₂ - T₁⁻¹ + T₁⁻¹ T₂⁻¹ - T₂⁻¹, Labeled → True]

Out[]=

In[]:= **Export**["PPDemo.pdf", PPDemo]

Out[]=

PPDemo.pdf

In[]:= **1 / 2 + T₁ - T₁ T₂ + T₂ - T₁⁻¹ + T₁⁻¹ T₂⁻¹ - T₂⁻¹ // TeXForm**

Out[]//TeXForm=

$$-T_2 T_1 + T_1 + T_2 - \frac{1}{T_2} + \frac{1}{T_2 T_1} - \frac{1}{T_1} + \frac{1}{2}$$

tex

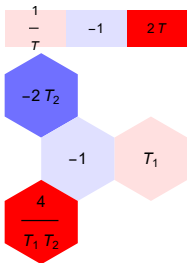
$$\{\text{def}\backslash\text{nbpdfPostInput}\{\backslash\text{hfill}\}$$

pdf

**In[]:= PolyPlot[{2 T - 1 + T⁻¹, -1 + T₁ - 2 T₂ + 4 T₁⁻¹ T₂⁻¹},
ImageSize → 100, Labeled → True]**

Out[]=

pdf



tex

}

The definition of `\verb$CF$` below is a technicality telling the computer how to best store polynomials in the $g_{\nu\alpha\beta}$'s such as F_1 and F_2 . The programs would run just the same without it, albeit a bit more slowly:

pdf

```
In[*]:= CF[ $\mathcal{E}_-$ ] := Expand@Collect[ $\mathcal{E}_-$ ,  $g_{--}$ , F] /. F -> Factor;
```

tex

Next, we decree that $T_3 = T_1 T_2$ and define the three “Feynman Diagram” polynomials F_1 , F_2 , and F_3 :

pdf

```
In[*]:= T3 = T1 T2;
```

pdf

```
In[*]:= F1[{ $s_-$ ,  $i_-$ ,  $j_-$ }] := 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  $g_{2jj} g_{3ii}$  - ( $1 - T_3^s$ )  $g_{2ji} g_{3ji}$  -
   $g_{2ii} g_{3jj}$  -  $T_2^s g_{2ji} g_{3jj}$  +  $g_{1ii} g_{3jj}$  + (( $T_1^s - 1$ )  $g_{1ji}$  ( $T_2^{2s} g_{2ji} - T_2^s g_{2jj} + T_2^s g_{3jj}$ ) +
  ( $T_3^s - 1$ )  $g_{3ji}$  ( $1 - T_2^s g_{1ii} + g_{2ij} + (T_2^s - 2) g_{2jj} - (T_1^s - 1) (T_2^s + 1) g_{1ji}$ )) / ( $T_2^s - 1$ )]
```

pdf

```
In[*]:= F2[{ $s_{\theta_-}$ ,  $i_{\theta_-}$ ,  $j_{\theta_-}$ }, { $s_{1_-}$ ,  $i_{1_-}$ ,  $j_{1_-}$ }] := CF[
   $s_{1_-} (T_1^{s_{\theta_-}} - 1) (T_2^{s_{1_-}} - 1)^{-1} (T_3^{s_{1_-}} - 1) g_{1,j_{1_-},i_{\theta_-}} g_{3,j_{\theta_-},i_{1_-}} ( (T_2^{s_{\theta_-}} g_{2,i_{1_-},i_{\theta_-}} - g_{2,i_{1_-},j_{\theta_-}}) - (T_2^{s_{\theta_-}} g_{2,j_{1_-},i_{\theta_-}} - g_{2,j_{1_-},j_{\theta_-}}) ) )]$ 
```

pdf

```
In[*]:= F3[ $\varphi_-$ ,  $k_-$ ] =  $\varphi g_{3kk} - \varphi / 2$ ;
```

tex

Next comes the main program computing $\Delta(K)$. Fortunately, it matches perfectly with the mathematical description in Section~\ref{sec:MainTheorem}. In line 1 below we use `Rot` to let X and φ be the crossings and rotation numbers of K . In addition we let n be the length of X , namely, the number of crossings in K , and we let the starting value of A be the $(2n+1)$ -times $(2n+1)$ identity matrix. Then in line 2, for each crossing in X we add to A a 2×2 block, in rows i and j and columns $i+1$ and $j+1$, as explain in Equation~\ref{eq:A}. In line 3 we compute the normalized Alexander polynomial Δ as in~\ref{eq:Delta}. In line 4 we let G be the inverse of A . In line 5 we declare what it means to evaluate, `ev`, a formula $\mathcal{E}$ that may contain symbols of the form $g_{\nu\alpha\beta}$: each such symbol is to be replaced by the entry in position α, β of G , but with T replaced with T_{ν} . In line 6 we start computing θ by computing the first summand in~\ref{eq:Main}, which in itself, is a sum over the crossings of the knot. In line 7 we add to θ the double sum corresponding to the second term in~\ref{eq:Main}, and in line 8, we add the third summand of~\ref{eq:Main}. Finally, line 9 outputs a pair: Δ , and the re-normalized version of θ .

pdf

```

In[ ]:=  $\Theta[K\_]$  :=  $\Theta[K]$  = Module[{{X,  $\varphi$ , n, A,  $\Delta$ , G, ev,  $\theta$ , k, k1, k2}},
  (* 1 *) {X,  $\varphi$ } = Rot[K]; n = Length[X]; A = IdentityMatrix[2 n + 1];
  (* 2 *) Cases[X, {s_, i_, j_}  $\Rightarrow$  (A[[{i, j}, {i + 1, j + 1}]] +=  $\begin{pmatrix} -T^s & T^s \\ 0 & -1 \end{pmatrix}$ )]];
  (* 3 *)  $\Delta$  = T(-Total[ $\varphi$ ]-Total[X[[All,1]]])/2 Det[A];
  (* 4 *) G = Inverse[A];
  (* 5 *) ev[ $\mathcal{E}_-$ ] := Factor[ $\mathcal{E}_-$  /. gv-,  $\alpha_-$ ,  $\beta_-$   $\Rightarrow$  (G[[ $\alpha$ ,  $\beta$ ]] /. T  $\rightarrow$  Tv)]];
  (* 6 *)  $\theta$  = ev[Sum[F1[X[[k]]], {k, n}]]];
  (* 7 *)  $\theta$  += ev[Sum[F2[X[[k1]], X[[k2]]], {k1, n}, {k2, n}]]];
  (* 8 *)  $\theta$  += ev[Sum[F3[ $\varphi$ [[k]], k], {k, Length@ $\varphi$ }}];
  (* 9 *) Factor@{ $\Delta$ , ( $\Delta$  /. T  $\rightarrow$  T1) ( $\Delta$  /. T  $\rightarrow$  T2) ( $\Delta$  /. T  $\rightarrow$  T3)  $\theta$ }
];

```

tex

\subsection{Examples} \label{ssec:Examples}

On to examples! Starting with the trefoil knot.

pdf

```

In[ ]:= Expand[ $\Theta$ [Knot[3, 1]]]

```

pdf

KnotTheory: Loading precomputed data in PD4Knots`.

Out[]:=

pdf

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

tex

%\needspace{2cm}

{\def\nbpdfPostInput{\hfill}}

pdf

```

In[ ]:= PolyPlot[ $\Theta$ [Knot[3, 1]], ImageSize -> Tiny]

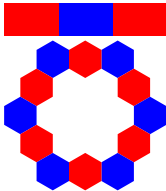
```

pdf

KnotTheory: Loading precomputed data in PD4Knots`.

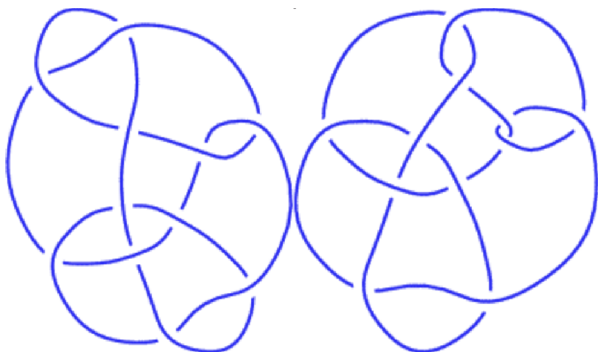
Out[]:=

pdf



tex

}



tex

```
\parpic[r]{\parbox{42mm}{
\includegraphics[width=20mm]{figs/K11n34.png}\hfill\includegraphics[width=20mm]{figs/K11n42.png}
}}
```

Next are the Conway knot 11_{34} and the Kinoshita-Terasaka knot 11_{42} . The two are mutants and famously hard to separate: they both have $\Delta=1$ (as evidenced by their one-bar Alexander bar codes below), and they have the same hyperbolic volume, HOMFLY-PT polynomial, and Khovanov homology. Yet their Θ invariants are different. Note that the genus of the Conway knot is 3, while the genus of the Kinoshita-Terasaka knot is 2. This agrees with the apparent higher complexity of the QR code of the Conway polynomial and with Conjecture~\ref{conj:Genus} below.

```
{\def\nbpdfPostInput{\hfill}
```

pdf

```
In[ ]:= PolyPlot[Theta[Knot[#]], ImageSize -> 120] & /@
{"K11n34", "K11n42"}
```

pdf

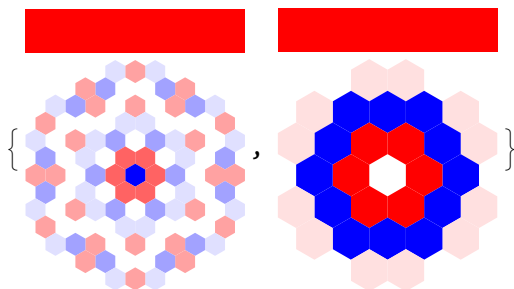
```
... KnotTheory: Loading precomputed data in DTCode4KnotsTo11`.
```

pdf

```
... KnotTheory: The GaussCode to PD conversion was written by Siddarth Sankaran at the University of Toronto in the summer of
2005.
```

Out[]:=

pdf



tex

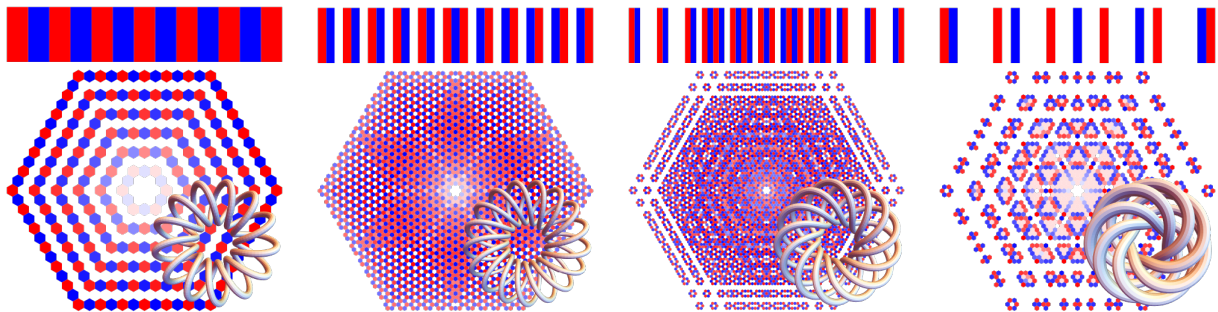
```
}
```

Torus knots have particularly nice-looking Θ invariants. Here are the torus knots $T_{13/2}$, $T_{17/3}$, $T_{13/5}$, and $T_{7/6}$:

pdf

```
In[ ]:= ImageCompose[PolyPlot[ $\Theta$ [TorusKnot @@ #], ImageSize -> 480],
  TubePlot[TorusKnot @@ #, ImageSize -> 240], {Right, Bottom}, {Right, Bottom}] & /@
  {{13, 2}, {17, 3}, {13, 5}, {7, 6}} // GraphicsRow
```

Out[]=
pdf



tex

The next line shows the computation time in seconds for the 132-crossing torus knot $T_{22/7}$ on a 2024 laptop, without actually showing the output. The output plot is in Figure~\ref{fig:T227}.

{\def\nbpdfPostInput{\hfill}}

pdf

```
In[ ]:= AbsoluteTiming[ $\Theta$ [TorusKnot[22, 7]]];]
```

Out[]=
pdf

```
{1020.73, Null}
```

```
In[ ]:= AbsoluteTiming[ $\Theta$ [Mirror@TorusKnot[22, 7]]];]
```

Out[]=

```
{1783.26, Null}
```

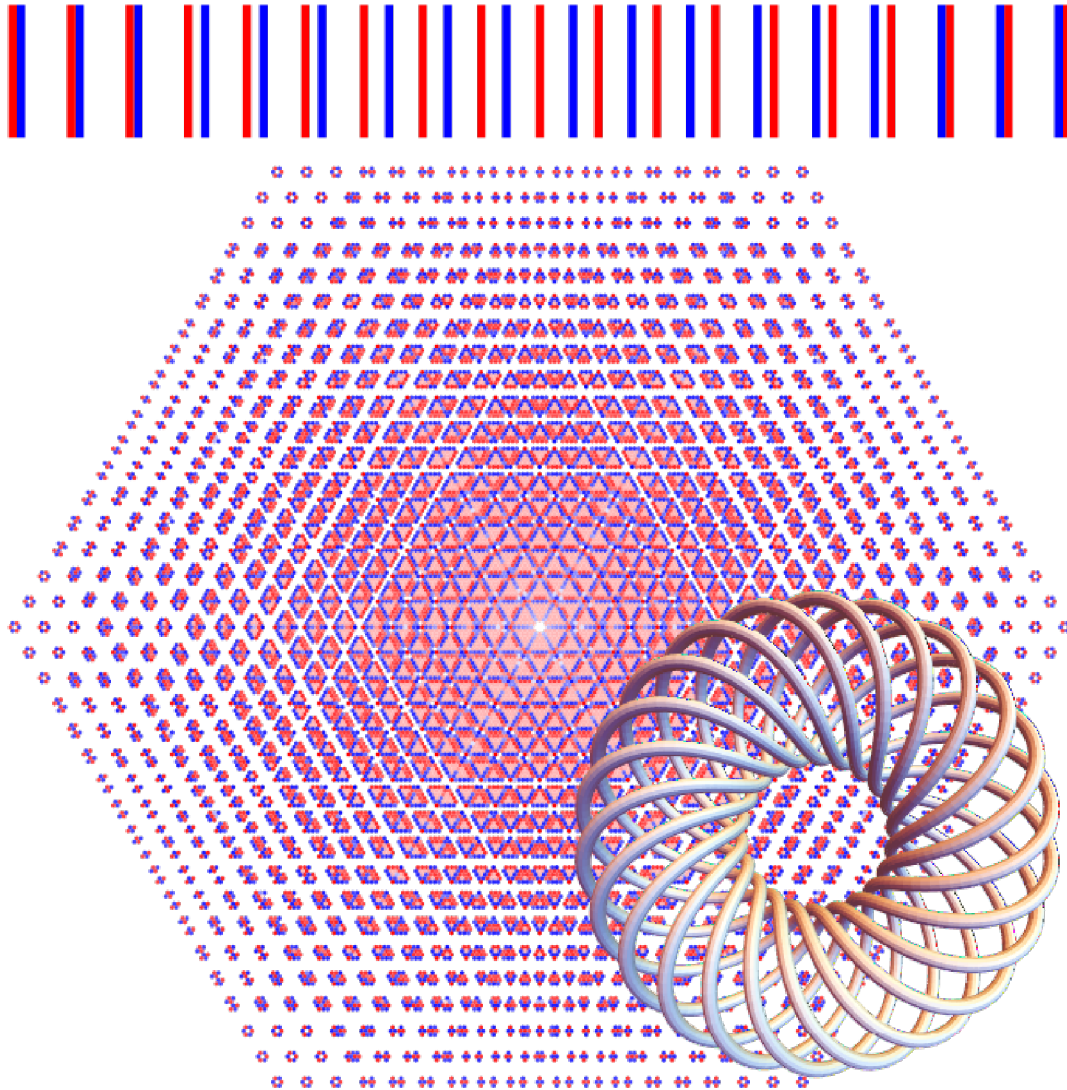
tex

```
}
\begin{figure}
```

pdf

```
In[ ]:= ImageCompose [PolyPlot[ $\Theta$ [TorusKnot[22, 7]], ImageSize → 720],
  TubePlot[TorusKnot[22, 7], ImageSize → 360], {Right, Bottom}, {Right, Bottom}]
```

```
Out[ ]:=
pdf
```



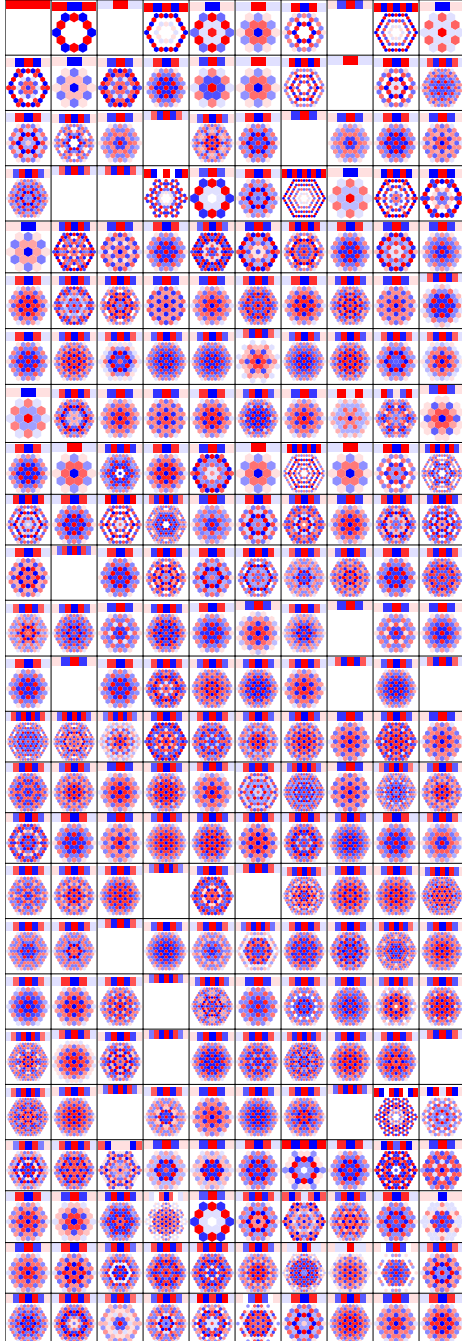
tex

```
\caption{The 132-crossing torus knot  $T_{22/7}$  and a plot of its  $\Theta$  invariant} \label{fig:T227}
\end{figure}
```

```
In[ ]:= tab250 = {{1, 0}} ~ Join ~ Table[ $\Theta$ [K], {K, AllKnots[{3, 10}]}];
```

```
g250 = GraphicsGrid[
  Partition[PolyPlot /@ tab250, 10],
  Spacings -> 0, Dividers -> All, ImagePadding -> None, PlotRangePadding -> None
]
```

Out[]=



In[]:= **Export**["Theta4Ro1fsen.pdf", g250]

Out[]=

Theta4Ro1fsen.pdf

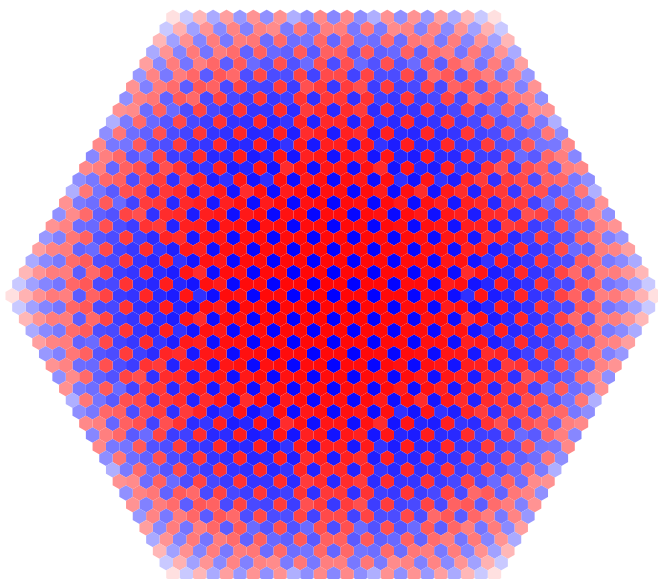
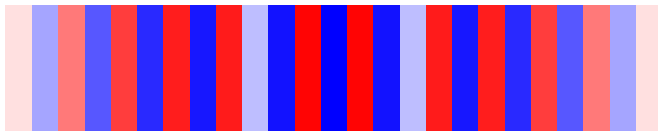
See also <https://drorbn.net/AcademicPensieve/Projects/HigherRank/DunfieldKnots/>.

```
In[*]:= DunfieldKnots = ReadList["../../People/Dunfield/nmd_random_knots"] /.  $k\_Integer \mapsto k + 1$ ;  
DK[n_] := DunfieldKnots[[n - 2]];
```

```
In[•]:= AbsoluteTiming[th =  $\Theta$ [DK[50]]];  
PolyPlot[th]
```

```
Out[ ]= {21.4991, Null}
```

Out[•]=

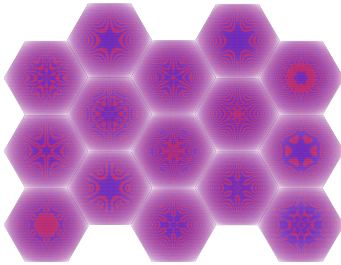



```

In[ ]:= Top15Images =
  Import["C:\\drorbn\\AcademicPensieve\\Projects\\HigherRank\\DunfieldKnots\\PP" <>
    ToString[#] <> ".png"] & /@
    {300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 317};
  Show[Top15 = Graphics[{
    {0, 0, 1}, {1, 0, 2}, {2, 0, 3}, {0, 1, 4}, {1, 1, 5}, {2, 1, 6}, {-1, 2, 7}, {0, 2, 8},
    {1, 2, 9}, {-1, 3, 10}, {0, 3, 11}, {1, 3, 12}, {-2, 4, 13}, {-1, 4, 14}, {0, 4, 15}
  },
  PlotRange -> {{-0.6, 4.1}, {-0.5, 3}}, ImageSize -> 1200] /. {x_, y_, k_} ->
  Inset[
    SetAlphaChannel[Top15Images[[k]], Graphics[{White,
      Polygon[Table[{Cos[t], Sin[t]}, {t, 0, 5  $\pi$  / 3,  $\pi$  / 3}]]], Background -> Black]],
    x {0, 1} + y {  $\sqrt{3}$  / 2, 1 / 2}, Center, 1.2
  ], ImageSize -> Small]

```

Out[]=



```

In[ ]:= Export["figs/Top15.png", ImageCrop[Top15]]

```

Out[]=

figs/Top15.png

```

In[ ]:= First15Data =
  (Get["C:\\drorbn\\AcademicPensieve\\Projects\\HigherRank\\DunfieldKnots\\D" <>
    StringPadLeft[ToString[#], 3, "0"] <> ".m"] [[
    2, 2]] /. {T1 -> T1, T2 -> T2}) & /@ Range[3, 17];

In[ ]:= First15Images = PolyPlot2 /@ First15Data;

```

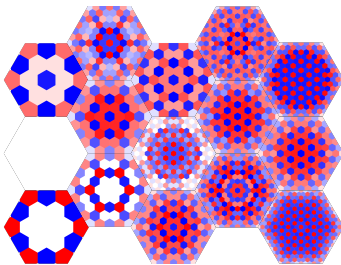


```

In[ ]:= Show[First15 = Graphics[{
  {0, 0, 1}, {1, 0, 2}, {2, 0, 3}, {0, 1, 4}, {1, 1, 5}, {2, 1, 6}, {-1, 2, 7}, {0, 2, 8},
  {1, 2, 9}, {-1, 3, 10}, {0, 3, 11}, {1, 3, 12}, {-2, 4, 13}, {-1, 4, 14}, {0, 4, 15}
},
PlotRange -> {{-0.6, 4.1}, {-0.5, 3}}, ImageSize -> 1200] /. {x_, y_, k_} ->
Inset[
  SetAlphaChannel[First15Images[[k]], Graphics[{White,
    Polygon[Table[{Cos[t], Sin[t]}, {t, 0, 5  $\pi$  / 3,  $\pi$  / 3}]]], Background -> Black]],
  x {0, 1} + y {  $\sqrt{3}$  / 2, 1 / 2}, Center, 1.2
], ImageSize -> Small]

```

Out[]:=



Top15Data =

```

(Get["C:\\drorbn\\AcademicPensieve\\Projects\\HigherRank\\DunfieldKnots\\D" <> ToString[
  #] <> ".m"]][2, 2] /. {T1 -> T1, T2 -> T2}) & /@
{300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 317};

```

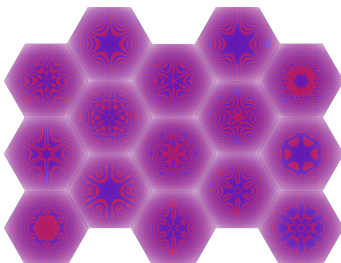
Top15Images = PolyPlot2 /@ Top15Data;

```

In[ ]:= Show[Top15 = Graphics[{
  {0, 0, 1}, {1, 0, 2}, {2, 0, 3}, {0, 1, 4}, {1, 1, 5}, {2, 1, 6}, {-1, 2, 7}, {0, 2, 8},
  {1, 2, 9}, {-1, 3, 10}, {0, 3, 11}, {1, 3, 12}, {-2, 4, 13}, {-1, 4, 14}, {0, 4, 15}
},
PlotRange -> {{-0.6, 4.1}, {-0.5, 3}}, ImageSize -> 1200] /. {x_, y_, k_} ->
Inset[
  SetAlphaChannel[Top15Images[[k]], Graphics[{White,
    Polygon[Table[{Cos[t], Sin[t]}, {t, 0, 5  $\pi$  / 3,  $\pi$  / 3}]]], Background -> Black]],
  x {0, 1} + y {  $\sqrt{3}$  / 2, 1 / 2}, Center, 1.2
], ImageSize -> Small]

```

Out[]:=



```
In[ ]:= Export["figs/Top15.pdf", ImageCrop[Top15]]
Out[ ]:=
figs/Top15.pdf
```

Θ with numerical inputs

tex

We note that if T_1 and T_2 are assigned specific rational numbers and if the program for Θ is slightly modified so as to compute each G_{ν} separately (rather than computing G symbolically and then substituting T to T_{ν}), then the program becomes significantly more efficient, for inverting a numerical matrix is cheaper than inverting a symbolic matrix (but then one obtains numerical answers and the beauty and the topological significance (Section~\ref{sec:SandM}) are lost). The Mathematica notebook that accompanies this paper, \cite[Theta.nb]{Self}, contains the required modified program as well as a few computational examples. One finds that with $T_1=22/7$ and $T_2=21/13$, the invariant Θ can be computed for knots with 600 crossings, and that for knots with up to 15 crossings, its separation power remains the same.

If T_1 and T_2 are assigned approximate real values, say π and e computed to 100 decimal digits, then Θ can be computed on knots with 1,000 crossings and, for knots with up to 15 crossings it remains very strong. But approximate real numbers are a bit thorny. It is hard to know how far one needs to compute before deciding that two such numbers are equal, and when two such numbers appear unequal, it is hard to tell if that is merely because they were computed differently and different roundings were applied. Thorns and snares are in the way of the perverse; He who guards his soul will be far from them (Proverbs 22:5)\footnote{\quad.\cjrL{/swmr np/sw yr.hq}}.

```
In[ ]:= Θ_{t1,t2}[K_] := Θ_{t1,t2}[K] = Module[{t, X, φ, n, A, Δ, G, ev, θ = 0},
  {X, φ} = Rot[K]; n = Length[X];
  A = IdentityMatrix[2 n + 1];
  Cases[X, {s_, i_, j_} :> (A[[{i, j}, {i + 1, j + 1}]] += (
    -T^s T^s - 1
    0 -1
  ))];
  {t1, t2, t3} = {t1, t2, t1 t2};
  Do[Δ_v = t_v^(-Total[φ] - Total[X[[All, 1]])/2) Det[A /. T -> t_v], {v, 3}];
  Do[G_v = Inverse[A /. T -> t_v], {v, 3}];
  ev[ε_] := ε /. {T_v -> t_v, g_{v, α, β} -> G_v[[α, β]]};
  Do[θ += ev@F1[X[[k]], {k, n}];
  Do[θ += ev@F2[X[[k1]], X[[k2]], {k1, n}, {k2, n}];
  Do[θ += ev@F3[φ[[k]], k], {k, 2 n}];
  {Δ1, Δ2, Δ3, Δ1 Δ2 Δ3 θ}
];
```

```

In[ ]:= With[{t1 = 22 / 7, t2 = 21 / 13, K = Knot[10, 101]},
  {Theta[t1, t2][K], Theta[K] /. {T1 -> t1, T2 -> t2, T -> t1 t2}}]

Out[ ]:=
  { { 88 663, 32 215, 72 840 385, 9 005 331 897 478 423 747 }, { 72 840 385, 9 005 331 897 478 423 747 } }
    { 3388, 10 647, 736 164, 433 730 593 728 432 }, { 736 164, 433 730 593 728 432 } }

In[ ]:= Monitor[
  Table[
    Echo[NumberOfKnots[{3, n}] - Length@Union@Table[Theta[22/7, 21/13][K], {K, AllKnots[{3, n]}]}],
    {n, 10, 15}
  ], K]

KnotTheory: Loading precomputed data in PD4Knots`.

» 0

KnotTheory: Loading precomputed data in DTCode4KnotsTo11`.

KnotTheory: The GaussCode to PD conversion was written by Siddarth Sankaran at the University of Toronto in the summer of
2005.

» 3

KnotTheory: Loading precomputed data in KnotTheory/12A.dts.

General: Further output of KnotTheory::loading will be suppressed during this calculation.

» 19
» 194
» 1118
» 6758

Out[ ]:=
  {0, 3, 19, 194, 1118, 6758}

```

The following line crashes my laptop at n=700:

```

In[ ]:= Do[Echo@{n -> AbsoluteTiming[Theta[22/7, 21/13][DK[n]]}], {n, 100, 1000, 100}]

» 100 -> {9.20528, Null}
» 200 -> {37.1709, Null}
» 300 -> {139.323, Null}
» 400 -> {302.169, Null}
» 500 -> {733.375, Null}
» 600 -> {1412.65, Null}

```

```

In[ ]:= Monitor[
  {r1, r2} = Echo@N[{ $\pi$ , e}, dec = 50];
  Table[
    Echo[NumberOfKnots[{3, n}] -
      CountDistinct[Table[N[ $\Theta_{r1, r2}[K]$ , dec / 2], {K, AllKnots[{3, n]}]}]],
    {n, 10, 15}
  ], K]
» {3.1415926535897932384626433832795028841971693993751,
  2.7182818284590452353602874713526624977572470937000}
KnotTheory: Loading precomputed data in PD4Knots`.
» 0
KnotTheory: Loading precomputed data in DTCode4KnotsTo11`.
KnotTheory: The GaussCode to PD conversion was written by Siddarth Sankaran at the University of Toronto in the summer of
2005.
» 3
KnotTheory: Loading precomputed data in KnotTheory/12A.dts.
General: Further output of KnotTheory::loading will be suppressed during this calculation.
» 19
» 194
» 1109

In[ ]:= {r1, r2} = Echo@N[{ $\pi$ ,  $(1 + \sqrt{5})/2$ }, 100];
Do[Echo@{n → AbsoluteTiming[ $\Theta_{r1, r2}[DK[n]]$ ]}, {n, 100, 1000, 100}]
» {3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342-
117068,
1.618033988749894848204586834365638117720309179805762862135448622705260462818902449707207204189-
391137}
» 100 → {9.01107,
{4.482694362539232574701053839023169156272666361182388272903975674512417607420687354252103602-
61600679 × 106,
0.244367383866191990261729892936648366149104794465977442793484954678783685700567609433815255-
918660195,
2.706952929746772312394426937040678227229265547817193763952417910648141113947504349750076963-
49948412 × 1014,
1.443704996840666397466273784548155745622699561616229361436865096139843495949644947520006361-
298 × 1023}}

```

- » 200 → {36.0862,
 $\{-1.18727928492276706294240090063426906154686664062527997214092273693001799386124242960017118-400903192 \times 10^{15},$
29.69733733512682042134919627767284853510615664777543150845466699703712301732014411109043238-30657499,
4.213310805165255704368320868588111718310775905052612595914168253125102258845625258951845746-8340826 $\times 10^{33},$
 $-8.767457545110443094371252345597285108955181262228377612094942779558996661644657488432465303 \times 10^{51}\}$
- » 300 → {96.1869,
 $\{7.089173830085219905811553245250111990338390614324722457082370098012200645341705504058909340-4000286 \times 10^{25},$
1948.913630402733766531350862358562463878167441528527273566238171944495124761031540093517042-14658773,
 $-1.33349844665939007633042426892172508607635776126078918014549695660423435859111597134532707-198548183 \times 10^{47},$
 $-5.46267641256745676321217034385814384366000216115107383187532679762227457908551406022223211-54 \times 10^{77}\}$
- » 400 → {172.213,
 $\{3.510317794718962927783888612709425469998150367103719530720444214445837272060967526789827639-01488886 \times 10^{34},$
-71 442.2820846361218315904628902695556633720818555781045215902189692033369072078182079228897-263471608,
1.250129191946950361728605727952624888457100602171629278629672791823693261702467547702298605-11918013 $\times 10^{61},$
 $-1.249120936663355769285241631147335772617272668159319393987793146120053795734625832941200194 \times 10^{103}\}$
- » 500 → {272.045,
 $\{-4.81947032694393372217453252761909737806222499074775701108126283299727702000598942348918975-41046368 \times 10^{40},$
-6163.94577659006187441397278546133479998399635263249378794746287883186373729527624524062494-73268976,
 $-1.17967870349703616140837217663005221839821870579038171080896664112458253285787403620547555-935313580 \times 10^{75},$
 $5.11834358728383779787798908406640950258337289166047659558613646738160019195618933287466323 \times 10^{121}\}$
- » 600 → {406.829,
 $\{1.775004209073710307946864700201455267664789805583665223763240994196661723179840198975621678-87974702 \times 10^{54},$
-158 221.695338938356362894027301097503111800949845436638716747406159426743325874082030743557-209927869,
1.106423112818815991230309449811148227700323143028690570923106911316948993892232373977509162-10682640 $\times 10^{96},$
 $-4.826558318437098777176777176865792197108347922562413147370461572223240349161081767460734475 \times 10^{156}\}$

» 700 → { 559.931,
 { 4.912566456455316265842124803991000772082026358612247239553053134779197564503416152939172938-
 52491182 × 10⁵⁴,
 525067.4174923157401982289231216004315331783629597705965571539912684099315928113378489149700-
 6086641,
 -1.91134318833867773324518283277963655961335342008703955242280786098847010769046173286331872-
 930017154 × 10¹⁰³,
 2.0218282568762719747028265740400144519185583649500352996483670201255757634239436504614062 ×
 10¹⁶⁵ } }
 » 800 → { 788.663,
 { -3.64209906546810405054314776627034014014347211577656030417886207193452102865845662778367875-
 559563008 × 10⁷⁰,
 1.85006699063327845828229222335817334963551339123665743363716759073170655206678358262188470-
 12873805 × 10¹⁰,
 1.847802888663950568224860403603941613173156090300018710014535575548285705118839736590137367-
 33984127 × 10¹²⁴,
 -6.26104794796003774066297049718339769045866752760731385197446809366983977065449768463488638 ×
 10²⁰⁵ } }
 » 900 → { 1051.99,
 { 3.149167024101946932854378218635776138150149040688977287592146617054513053236022784918143921-
 57637288 × 10⁷⁸,
 -6.30882506670073103396873943652464908329350583695804004638524043076215076085213755538182887-
 522626537 × 10¹¹,
 -1.28777433815718988970611843935202559444290130074261671385253305724241996009345445367202897-
 198207778 × 10¹³⁵,
 -2.146158762183508670025586081413880382995702252636942836684683733135724625044921755478919620
 × 10²²⁶ } }
 » 1000 → { 1273.9,
 { 2.734657895518667018900602961142453380719671208278606063934869793602703478489594458720693611-
 8231180 × 10⁹³,
 2.601212381597743315805429010690529233958795582806190049007690642268584885943951890340110391-
 3057200 × 10¹⁴,
 -2.43425704022046777862249539637673490800236373023383574216931773598144673992230066149317162-
 85089385 × 10¹⁵⁶,
 -4.13556916837375875480427190896073476953710456343545123116309822206018949073744760406845414 ×
 10²⁶⁶ } }

In[]:= **dec = 100;**

num300 =

**Get["C:\\drorbn\\AcademicPensieve\\Projects\\HigherRank\\DunfieldKnots\\D300.m"][[2, 2]] /.
 {T1 → N[π , dec], T2 → N[(1 + $\sqrt{5}$) / 2, dec]}**

Out[]:=

-5.4626764125674567632121703 × 10⁷⁷

```
In[ ]:= {r1, r2} = Echo@N[{ $\pi$ , (1 +  $\sqrt{5}$ ) / 2}];  $\Theta_{r1, r2}$ [DK[300]]
» {3.14159, 1.61803}

Out[ ]:= {7.08917  $\times 10^{25}$ , 1948.91, -1.3335  $\times 10^{47}$ , -5.46268  $\times 10^{77}$ }
```

Proof of R3b

```
exec
nb2tex$TeXFileName = "Invariance-R3.tex";
```

tex

First, we implement the Kronecker δ -function, the g -rules for a crossing (s, i, j) , and the g -rules for a list of crossings X :

pdf

```
In[ ]:=  $\delta_{\alpha, \beta} := \text{If}[\alpha == \beta, 1, 0];$ 
gRules[{s_, i_, j_}] := { $g_{v_j \beta} \mapsto g_{vj^* \beta} + \delta_{j \beta}$ ,  $g_{v_i \beta} \mapsto T_v^s g_{vi^* \beta} + (1 - T_v^s) g_{vj^* \beta} + \delta_{i \beta}$ ,
 $g_{v_\alpha i^*} \mapsto T_v^s g_{v \alpha i} + \delta_{\alpha i^*}$ ,  $g_{v_\alpha j^*} \mapsto g_{v \alpha j} + (1 - T_v^s) g_{v \alpha i} + \delta_{\alpha j^*}$ };
gRules[X__List] := Union@@Table[gRules[c], {c, {X}}]
```

tex

We then let X_l be the three crossings in the left-hand-side of the R3b move, as in Figure~\ref{fig:R3}, we let A_l be the A^l term of~\eqref{eq:ABC}, and we let lhs be the result of applying the g -rules for the crossings in X_l to A_l . We print only a ``\verb\$Short\$'' version of lhs because the full thing would cover about 2.5 pages:

pdf

```
In[ ]:= Xl = {{1, j, k}, {1, i, k^+}, {1, i^+, j^+}};
Al = Sum[F1[c], {c, Xl}] + Sum[F2[c0, c1], {c0, Xl}, {c1, Xl}];
lhs = Simplify[Al /. gRules @@ Xl];
Short[lhs, 5]
```

Out[]//Short=

pdf

$$\begin{aligned}
& - \frac{1}{2(1 - T_2)} \\
& (3 - 3T_2 + \ll 129 \gg + 2(1 - T_2)(1 + T_2(T_2 g_{2, (i^+)^+, i} - (-1 + T_2) g_{2, (j^+)^+, i} - (-1 + T_2) g_{2, (k^+)^+, i}) \\
& (1 + (1 - T_1 T_2) g_{3, (k^+)^+, j} + g_{3, (k^+)^+, k}))
\end{aligned}$$

tex

We do the same for A^r , except this time, without printing at all:

pdf

```
In[ ]:= Xr = {{1, i, j}, {1, i^+, k}, {1, j^+, k^+}};
Ar = Sum[F1[c], {c, Xr}] + Sum[F2[c0, c1], {c0, Xr}, {c1, Xr}];
rhs = Simplify[Ar /. gRules @@ Xr];
```

tex

We then compare lhs with rhs . The output, $True$, tells us that we have proven~\eqref{eq:R3A}:

```
{\def\nbpdfPostInput{\hfill}}
```

pdf

In[]:= **Simplify**[lhs == rhs]

Out[]:=

pdf

True

tex

We show that $B^l = B^r$ by following exactly the same procedure. Note that we ignore the summation over c_y and instead treat c_y as a fixed crossing (s,m,n) . If an equality is proven for every fixed c_y , it is of course also proven for the sum over c_y in Y .

pdf

```
In[ ]:= lhs = Sum[F2[c0, {s, m, n}], {c0, X1}] /. gRules @@ X1;
rhs = Sum[F2[c0, {s, m, n}], {c0, Xr}] /. gRules @@ Xr;
Simplify[lhs == rhs]
```

Out[]:=

pdf

True

tex

Similarly we prove that $C^l = C^r$, and this concludes the proof of Proposition~\ref{prop:R3}.

pdf

```
In[ ]:= lhs = Sum[F2[{s, m, n}, c1], {c1, X1}] /. gRules @@ X1;
rhs = Sum[F2[{s, m, n}, c1], {c1, Xr}] /. gRules @@ Xr;
Simplify[lhs == rhs]
```

Out[]:=

pdf

True

tex

```
}
\vskip -7mm
\endpar{\ref{prop:R3}}
```

\begin{remark} \label{rem:E} The computations above were carried out for generic $g_{\{\nu\alpha\beta\}}$ and for a generic $c_y = (s,m,n)$; namely, without specifying the knot diagrams in full, and hence without assigning specific values to $g_{\{\nu\alpha\beta\}}$, and without specifying m and n . Under these conditions the three parts of~\eqref{eq:ABC} cannot mix (namely, terms from, say, A^h cannot cancel terms in B^h or C^h), and so it would have been enough to show that $E^l = E^r$, where E^h combines A^h and B^h and C^h (and a few harmless further terms) by adding c_y to the summation corresponding to A^h :

```
\[
E^h = \sum_{c \in \{c^h_1, c^h_2, c^h_3, c_y\}} F^h_1(c)
+ \sum_{c_0, c_1 \in \{c^h_1, c^h_2, c^h_3, c_y\}} F^h_2(c_0, c_1).
\]
```

But that's a simpler computation:

pdf

```
In[ ]:= ESum[X_] := (Sum[F1[c], {c, X}] + Sum[F2[c0, c1], {c0, X}, {c1, X}]) /. gRules @@ X;
```



```

tex
{\def\nbpdfPostInput{\hfill}

pdf
In[ ]:= Xl = {{1, j, k}, {1, i, k+}, {1, i+, j+}};
Xr = {{1, i, j}, {1, i+, k}, {1, j+, k+}};
Simplify[ESum[Append[Xl, {s, m, n}]] == ESum[Append[Xr, {s, m, n}]]]

Out[ ]:=
pdf
True

tex
}\end{remark}\endpar{\ref{rem:E}}

```

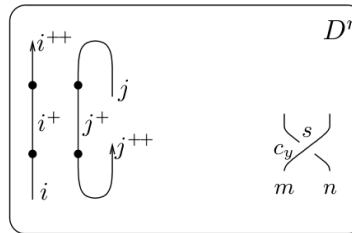
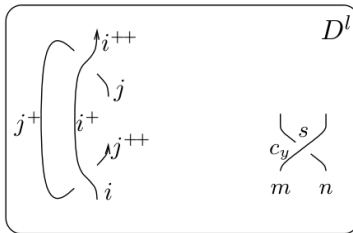
Proof of $R2c^+$ and of $R2c^-$

```

exec
nb2tex$TeXFileName = "Invariance-R2c.tex";

tex
\noindent{\em Proof.} For  $R2c^+$  we follow the same logic as in the proof of Proposition~\ref{prop:R3}, as simplified by Remark~\ref{rem:E}. We start with the figure that replaces Figure~\ref{fig:R3} (note the null vertices in  $D^l$  and their minimal effect as in Lemma~\ref{lem:NullVertices} and Remark~\ref{rem:F4Null}):
\[\{
\def\i{{i$}} \def\j{{j$}} \def\k{{k$}} \def\m{{m$}} \def\n{{n$}} \def\s{{s$}}
\input{figs/R2cE.pdf_t}
\}

```



As in Remark~\ref{rem:E}, we let E^l and E^r be the sums corresponding to the diagrams D^l and D^r above:

```

\[\begin{aligned}
E^l &= \sum_{\text{makebox}[0pt]{\scriptstyle c \in \{c^l_1, c^l_2, c_y\}}} F^l_1(c) \\
&\quad + \sum_{\text{makebox}[0pt]{\scriptstyle c_0, c_1 \in \{c^l_1, c^l_2, c_y\}}} F^l_2(c_0, c_1) \\
&\quad + F^l_3(jp)|_{\varphi(jp)=1}, \\
&\quad \text{quad} \\
E^r &= F^r_1(c_y) + F^r_2(c_y, c_y) + F^r_3(jp)|_{\varphi(jp)=1}.
\end{aligned}
\]

```

We need to show that $E^l = E^r$ after all relevant g -rules are applied to both sides.

To compute these E sums we first have to extend the `ESum` routine to accept also a list R of

pairs (φ, k) of the form (rotation number, edge label):

pdf

```
In[*]:= ESum[X_, R_] :=
  (Sum[F1[c], {c, X}] + Sum[F2[c0, c1], {c0, X}, {c1, X}] + Sum[F3@@r, {r, R}]) /.
  gRules @@ X;
```

tex

We then compute E^A (and apply the relevant g -rules) by calling `\verb$ESum$` with crossings $(-1, i, j)$, $(1, i, j)$, and (s, m, n) , and a rotation number of 1 on edge j :

pdf

```
In[*]:= E1 = Simplify[ESum[{{-1, i, j+}, {1, i+, j}, {s, m, n}}, {{1, j+}}]];
Short[E1, 5]
```

Out[*]//Short=

pdf

$$-\frac{1}{2(-1+T_2^2)}(1+s+2s(T_1T_2)^sg_{3,m^+,m}+2g_{3,(j^+)^+,j}-T_2^s(1+s-2sg_{1,n^+,m}g_{2,n^+,m}+2sg_{2,m^+,m}(1+g_{3,n^+,n})+2g_{3,(j^+)^+,j}))$$

tex

The computation of E^r is simpler, as it only involves the generic (s, m, n) and the rotation $(1, j)$. We implement the g -rules for null vertices as in Equations~\eqref{eq:NullCarRules} and~\eqref{eq:NullCounterRules}, compute E^r , and then compare E^A with E^r to conclude the invariance under $R2c^+$:

pdf

```
In[*]:= gRules[j_] := {g_{v,j,\beta} \Rightarrow \delta_{j,\beta} + g_{v,j^+,\beta}, g_{v,\alpha,j^+} \Rightarrow \delta_{\alpha,j^+} + g_{v,\alpha,j}}
```

pdf

```
In[*]:= Er = ESum[{{s, m, n}}, {{1, j+}}] /. (Union @@ gRules /@ {i, i+, j, j+});
Simplify[E1 == Er]
```

Out[*]=

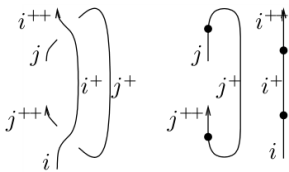
pdf

True

tex

For $R2c^+$ we allow ourselves to be even more concise:

```
\parpic[r]{\def i{{i$}} \def j{{j$}} \input{figs/uR2cm.pdf_t}
\pickskip{1}
```



exec

```
nb2tex$PDFwidth == 1.8;
```

pdf

```
In[ ]:= E1 = ESum[{ {1, i, j^+}, {-1, i^+, j}, {s, m, n}}, {{-1, j^+}}];
Er = ESum[{ {s, m, n}}, {{-1, j^+}}] /. (Union@gRules /@ {i, i^+, j, j^+});
Simplify[Er == E1]
```

Out[]:=

pdf

True

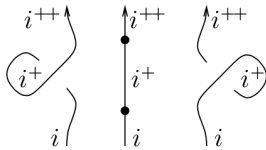
tex

\endpar{\ref{prop:R2c}}

exec

```
nb2tex$PDFWidth += 1.8;
nb2tex$TeXFileName = "Invariance-R1s.tex";
```

Proofs of R1l and R1r



tex

\noindent{\em Proof.} We aim to use the same approach and conventions as in the previous two proofs but hit a minor snag. The $\$g\$$ -rules for R1l include

```
\[
g_{\{ip\}\beta} = \delta_{\{ip\}\beta} + T g_{\{ipp\},\beta} + (1-T) g_{\{ip\},\beta}
\quad\text{and}\quad
g_{\{\alpha,\ip\}} = g_{\{\alpha i\}} + (1-T) g_{\{\alpha\}\ip} + \delta_{\{\alpha,\ip\}},
\]
```

and if these are implemented as simple left to right replacement rules, they lead to infinite recursion. Fortunately, these rules can be rewritten in the form

```
\[
g_{\{ip\}\beta} = T^{-1} \delta_{\{ip\}\beta} + g_{\{ipp\},\beta}
\quad\text{and}\quad
g_{\{\alpha,\ip\}} = T^{-1} g_{\{\alpha i\}} + T^{-1} \delta_{\{\alpha,\ip\}},
\]
```

which makes perfectly valid replacement rules. We thus redefine:

pdf

```
In[ ]:= gRules[{1, i^+, i}] = {g_{v_i\beta} -> g_{v_i^+\beta} + \delta_{i\beta}, g_{v_{i^+}\beta} -> g_{v(i^+)\beta} + T^{-1} \delta_{i^+\beta},
g_{v_{\alpha}(i^+)} -> T g_{v\alpha i} + \delta_{\alpha(i^+)}, g_{v_{\alpha}i} -> T^{-1} g_{v\alpha i} + T^{-1} \delta_{\alpha i}};
```

tex

The same issue does not arise for R1r (!), and thus the following lines conclude the proof:

```
{\def\nbpdfPostInput{\hfill}}
```

pdf

```
In[*]:= E1 = ESum[{{1, i+, i}, {s, m, n}}, {{1, i+}}];
Em = ESum[{{s, m, n}}];
Er = ESum[{{1, i, i+}, {s, m, n}}, {{-1, i+}}]; Simplify[E1 == Em == Er]
```

Out[*]=

pdf

True

tex

\endpar{\ref{prop:R1s}}

exec

nb2tex\$TeXFileName = "Invariance-Sw.tex";

Proof of Sw

The diagram illustrates the proof of the Sw relation. It shows two configurations of four strands labeled i , j , i^+ , and j^+ . In the left configuration, i and j are at the bottom, and i^+ and j^+ are at the top. i connects to i^+ and j connects to j^+ . In the right configuration, i and j are at the bottom, and i^+ and j^+ are at the top. i connects to j^+ and j connects to i^+ . The crossings are labeled with $\varphi = 1$ and $\varphi = -1$.

tex

\noindent{\em Proof.} This one is routine:

pdf

```
In[*]:= E1 = ESum[{{1, i, j}, {s, m, n}}];
Er = ESum[{{1, i, j}, {s, m, n}}, {{-1, i}, {-1, j}, {1, i+}, {1, j+}}];
Simplify[E1 == Er]
```

Out[*]=

pdf

True

tex

\endpar{\ref{prop:Sw}}

Proof of Polynomiality

```
In[*]:= R1 = Simplify[
  Residue[F1[{s0, i0, j0}], {T2, 1}] /. {g2,α-,α- → 1, g2,j0,i0 → 1 - g2,i0,j0, g3,α-,β- → g1,α,β}]
```

Out[*]=

$$- \left((-1 + T_1^{s_0}) g_{1,j_0,i_0} (g_{1,i_0,i_0} + 2 (-1 + T_1^{s_0}) g_{1,j_0,i_0} - g_{1,j_0,j_0}) \right)$$

```
In[*]:= R2 = Simplify[Residue[F2[{s0, i0, j0}, {s1, i1, j1}], {T2, 1}] /. g3,α-,β- → g1,α,β]
```

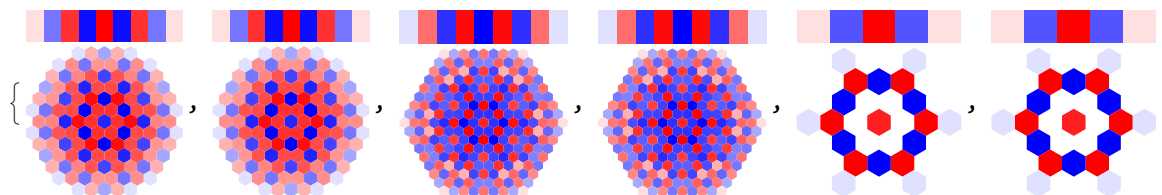
Out[*]=

$$(-1 + T_1^{s_0}) (-1 + T_1^{s_1}) g_{1,j_0,i_1} g_{1,j_1,i_0} (g_{2,i_1,i_0} - g_{2,i_1,j_0} - g_{2,j_1,i_0} + g_{2,j_1,j_0})$$

The deficit for 11 crossing knots

```
In[ ]:= PolyPlot[Θ[#], ImageSize → Tiny] & /@ {Knot["K11a44"], Knot["K11a47"],
  Knot["K11a57"], Knot["K11a231"],
  Knot["K11n73"], Knot["K11n74"]}
```

Out[]:=



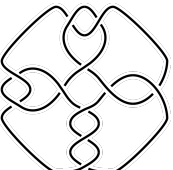
```
In[ ]:= Equal @@@ Partition[Θ /@ {Knot["K11a44"], Knot["K11a47"],
  Knot["K11a57"], Knot["K11a231"], Knot["K11n73"], Knot["K11n74"]}, 2]
```

Out[]:=

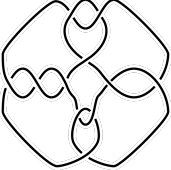
```
{True, True, True}
```

```

In[ ]:= Pair1 = ImageCompose[
  ImageCompose[PolyPlot[ $\Theta$ [Knot["K11a44"]], ImageSize -> 480],

  ColorQuantize[ImageResize[, 230], {Black, White}],

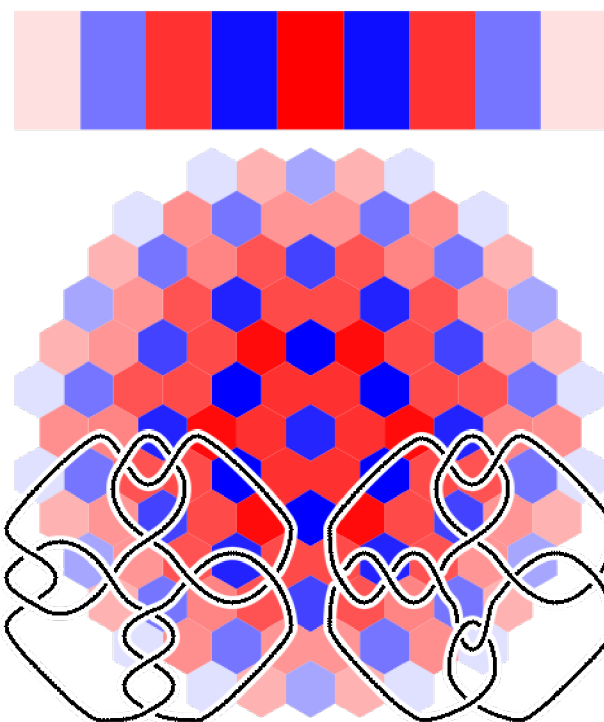
  {Left, Bottom}, {Left, Bottom}],

  ColorQuantize[ImageResize[, 230], {Black, White}],

  {Right, Bottom}, {Right, Bottom}]

```

Out[]=



```

In[ ]:= Export["figs/Pair1.pdf", Pair1]

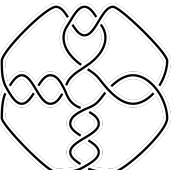
```

Out[]=

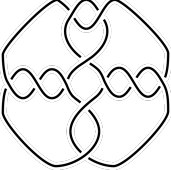
figs/Pair1.pdf

```

In[ ]:= Pair2 = ImageCompose[
  ImageCompose[PolyPlot[ $\Theta$ [Knot["K11a57"]], ImageSize -> 480],

  ColorQuantize[ImageResize[, 230], {Black, White}],

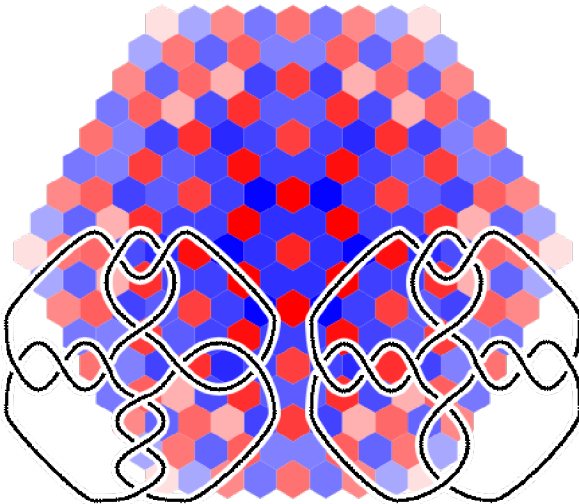
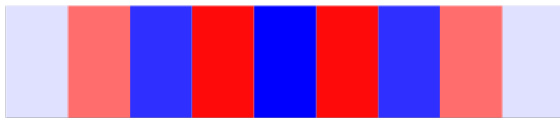
  {Left, Bottom}, {Left, Bottom}],

  ColorQuantize[ImageResize[, 230], {Black, White}],

  {Right, Bottom}, {Right, Bottom}]

```

Out[]:=



```

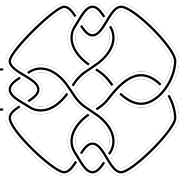
In[ ]:= Export["figs/Pair2.pdf", Pair2]

```

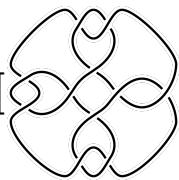
Out[]:=

figs/Pair2.pdf

```
In[ ]:= Pair3 = ImageCompose[
  ImageCompose[PolyPlot[Theta[Knot["K11n73"]]], ImageSize -> 480],

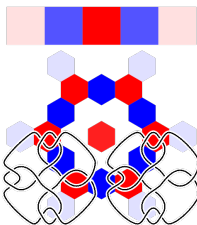
  ColorQuantize[ImageResize[, 230], {Black, White}],

  {Left, Bottom}, {Left, Bottom}],

  ColorQuantize[ImageResize[, 230], {Black, White}],

  {Right, Bottom}, {Right, Bottom}]
```

Out[]:=



```
In[ ]:= Export["figs/Pair3.pdf", Pair3]
```

Out[]:=

figs/Pair3.pdf

exec

```
nb2tex$TeXFileName = "GST48.tex";
```

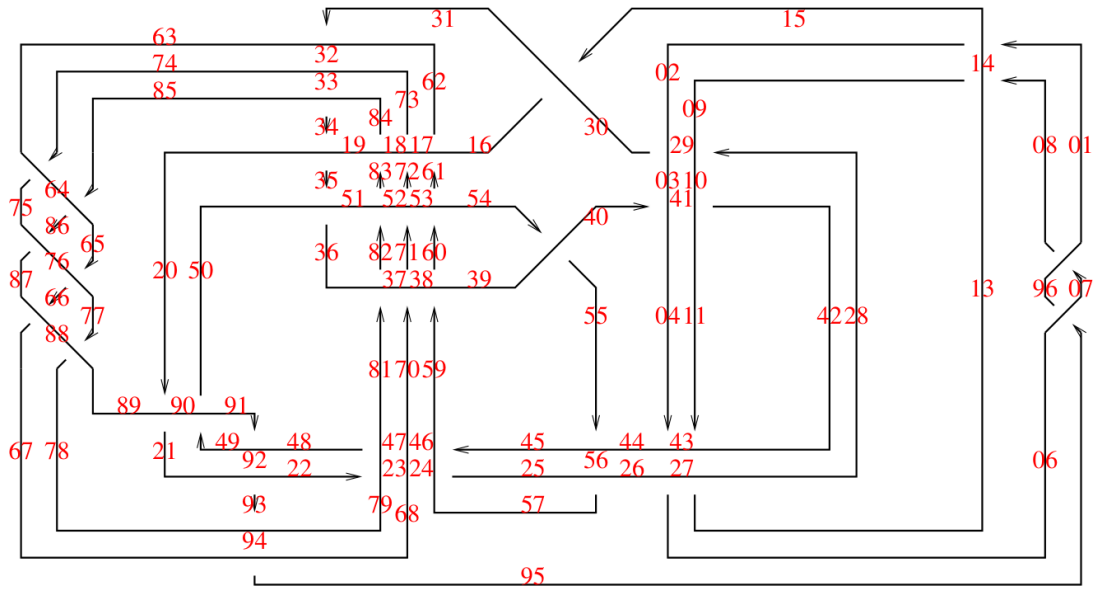
The GST₄₈ Knot

tex

```
{\def\nbpdfPostInput{\hfill}
{\def\nbpdfPreOutput{\noindent\hspace{-7mm}}}
```

exec

```
nb2tex$PDFWidth *= 0.61;
```

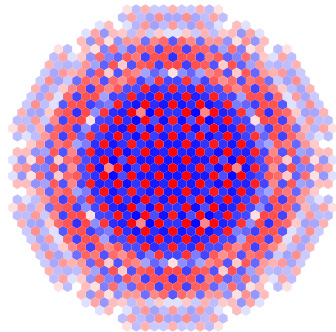
pdf

```
In[ ]:= GST48 = EPD [X14,1, X̄2,29, X3,40, X43,4, X̄26,5, X6,95, X96,7, X13,8, X̄9,28, X10,41, X42,11, X̄27,12,
X30,15, X̄16,61, X̄17,72, X̄18,83, X19,34, X̄89,20, X̄21,92, X̄79,22, X̄68,23, X̄57,24, X̄25,56, X62,31,
X73,32, X84,33, X̄50,35, X36,81, X37,70, X38,59, X̄39,54, X44,55, X58,45, X69,46, X80,47, X48,91,
X90,49, X51,82, X52,71, X53,60, X̄63,74, X̄64,85, X̄76,65, X̄87,66, X̄67,94, X̄75,86, X̄88,77, X̄78,93];
AbsoluteTiming@PolyPlot[{Δ48, θ48} = Θ[GST48], ImageSize → Small]
```

Out[]:=
pdf



{14.5823, }



tex

}

exec

```
nb2tex$PDFwidth /= 0.61;
```

pdf

```
In[ ]:= {Exponent[Δ48, T], ⌈Exponent[θ48, T1] / 2⌋}
```

Out[]:=
pdf

```
{8, 10}
```

tex

}

exec

```
nb2tex$TeXFileName = "KS.tex";
```

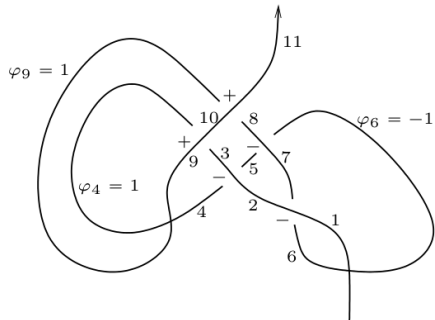
The rotational virtual knot KS

tex

```
{\def\nbpdfPostInput{\hfill}
{\def\nbpdfPreOutput{\noindent\hspace{-7mm}}}
```

exec

```
nb2tex$PDFWidth *= 0.8;
```

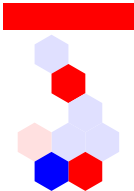


pdf

```
In[*]:= KS = {{ {-1, 1, 6}, {-1, 2, 4}, {1, 9, 3}, {-1, 7, 5}, {1, 10, 8}},
               {0, 0, 0, 1, 0, -1, 0, 0, 1, 0, 0}};
PolyPlot[0[KS], ImageSize -> Tiny]
```

Out[*]=

pdf



tex

```
}
```

exec

```
nb2tex$PDFWidth /= 0.8;
```

tex

```
}
```

```
In[*]:= ConnectedSum[Knot[3, 1], Knot[5, 1]] // PD
```

Out[*]=

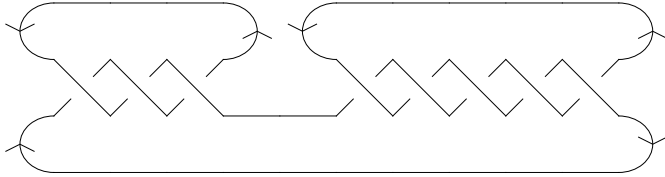
```
PD[X[11, 14, 12, 15], X[13, 16, 14, 1], X[15, 12, 16, 13],
   X[1, 6, 2, 7], X[3, 8, 4, 9], X[5, 10, 6, 11], X[7, 2, 8, 3], X[9, 4, 10, 5]]
```

```
In[ ]:= ConnectedSum[Knot[3, 1], Knot[5, 1]] // DrawMorseLink
```

KnotTheory: MorseLink was added to KnotTheory` by Siddarth Sankaran at the University of Toronto in the summer of 2005.

KnotTheory: DrawMorseLink was written by Siddarth Sankaran at the University of Toronto in the summer of 2005.

```
Out[ ]:=
```



```
In[ ]:= rθ[K_] := rθ[K] = Module[{Δ, θ},
  {Δ, θ} = θ[K];
  Factor[(Δ /. T → T1)-1 (Δ /. T → T2)-1 (Δ /. T → T3)-1 θ]
]
```

```
In[ ]:= {K1, K2} = {Knot[3, 1], Knot[5, 1]};
Factor[ $\frac{r\theta[\text{ConnectedSum}[K1, K2]]}{r\theta[K1] + r\theta[K2]}$ ]
```

```
Out[ ]:=
```

1

```
In[ ]:= Monitor[
  Union@Flatten@Table[Simplify[rθ[ConnectedSum[K1, K2]] == rθ[K1] + rθ[K2]],
    {K1, AllKnots[{3, 8]}}, {K2, AllKnots[{3, 8]}},
    {K1, K2}]
```

```
Out[ ]:=
```

{True}