Pensieve header: Compute the signatures of AllKnots[{3,12}].

(Alt) In[ ]:=

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Talks\\ICERM-2305"];
```

(Alt) In[ ]:=
pdf

```
Once[<< KnotTheory`];
```

pdf

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

(Alt) In[ ]:=
pdf

```
θ[x_] /; NumericQ[x] := UnitStep[x]
```

(Alt) In[ ]:=
pdf

```
ω2[v_][p_] := Module[{q = Expand[p], n, c},
   If[q === 0, 0, c = Coefficient[q, ω, n = Exponent[q, ω]];
    c v^n + ω2[v][q - c (ω + ω^-1)^n]]];
```

(Alt) In[ ]:=
pdf

```
sign[ℰ_] := Module[{n, d, v, p, rs, e, k},
  {n, d} = NumeratorDenominator[ℰ];
  {n, d} /= ω^(Exponent[n,ω]/2+Exponent[n,ω,Min]/2);
  p = Factor[ω2[v]@n * ω2[v]@d /. v → 4 u^2 - 2];
  rs = Solve[p == 0, u, Reals];
  If[rs === {}, Sign[p /. u → 0],
   rs = Union@@(u /. rs);
   Sign[(-1)^(e=Exponent[p,u]) Coefficient[p, u, e]] + Sum[
     k = 0; While[(d = RootReduce[∂_{u,++k} p /. u → r]) == 0];
     If[EvenQ[k], 0, 2 Sign[d]] * θ[u - r],
     {r, rs}]
  ]
 ]
```

(Alt) In[ ]:=
pdf

```
SetAttributes[B, Orderless];
CF[b_B] := RotateLeft[#, First@Ordering[#] - 1] & /@ DeleteCases[b, {}]
```

(Alt) In[ ]:=
pdf

```
CF[ℰ_] := Module[{γs = Union@@Cases[ℰ, γ_ | γ̄_, ∞]},
  Total[CoefficientRules[ℰ, γs] /. (ps_ → c_) :> Factor[c] Times @@ γs^ps]]
```

(Alt) In[ ]:=
pdf

```
CF[{}] = {};
CF[C_List] := Module[{γs = Union@Cases[C, γ_, ∞], γ},
  CF /@ DeleteCases[0][
    RowReduce[Table[∂γ r, {r, C}, {γ, γs}]].γs] ]
```

(Alt) In[ ]:=
pdf

```
(ℰ_)* := ℰ /. {γ̄ → γ, γ → γ̄, ω → ω^-1, c_Complex :> c*};
r_Rule+ := {r, r*}
```

(Alt) In[ ]:=
pdf

```
RulesOf[γi_ + rest_.] := (γi → -rest)+;
CF[PQ[C_, q_]] := Module[{nC = CF[C]},
  PQ[nC, CF[q /. Union @@ RulesOf /@ nC]] ]
```

(Alt) In[ ]:=
pdf

```
CF[Σb_[σ_, pq_]] := ΣCF[b][σ, CF[pq]]
```

(Alt) In[ ]:=
pdf

```
Format[Σb_B[σ_, PQ[C_, q_]]] := Module[{γs},
   γs = γ# & /@ Join @@ b;
   Column[{TraditionalForm@σ,
     TableForm[Join[
       Prepend[""] /@ Table[TraditionalForm[∂c r], {r, C}, {c, γs}],
       {Prepend[""][
         Join @@ (b /. {l_, m___, r_} :> {DisplayForm@RowBox[{"(", l}],
               m, DisplayForm@RowBox[{r, ")"}]}]) /. i_Integer :> γi ]},
       MapThread[Prepend, {Table[TraditionalForm[∂r,c q], {r, γs*}, {c, γs}], γs*}]
      ], TableAlignments → Center]
    }, Center] ];
```

(Alt) In[ ]:=
pdf

```
Σb1_[σ1_, PQ[C1_, q1_]] ⊕ Σb2_[σ2_, PQ[C2_, q2_]] ^:=
  CF@ΣJoin[b1,b2][σ1 + σ2, PQ[C1 ⋃ C2, q1 + q2]];
```

(Alt) In[ ]:=
pdf

```
GTi_,j_@ΣB[{li___,i_,ri___},{lj___,j_,rj___},bs___][σ_, PQ[C_, q_]] :=
 CF@ΣB[{ri,li,j,rj,lj,i},bs][σ, PQ[C ⋃ {γi - γj}, q]]
```

(Alt) In[ ]:=
pdf

```
Cordon_i_@Σ_B[{li___,i_,ri___},bs___][σ_,PQ[C_,q_]]:=
 Module[{ϕ=∂_γi C, λ=∂_γ̄i,γi q, nσ=σ, nC, nq, p},
  {p}=FirstPosition[(#=!=0)&/@ϕ, True, {0}];
  {nC, nq}=Which[
    p>0, {C, q}/.(γi→-C[[p]]/ϕ[[p]])^+/.(γi→0)^+,
    λ=!=0, (nσ+=sign[λ]; {C, q/.(γi→-(∂_γ̄i q)/λ)^+/.(γi→0)^+}),
    λ===0, {C⋃{∂_γ̄i q}, q/.(γi→0)^+}];
  CF@Σ_B[Most@{ri,li},bs][nσ, PQ[nC, nq]/.(γ_Last@{ri,li}→γ_First@{ri,li})^+] ]
```

(Alt) In[ ]:=
pdf

```
c_i_,j_@t:Σ_B[{li___,i_,ri___},{___,j_,___},___][__]:=t//GT_j,First@{ri,li}//Cordon_j
```

(Alt) In[ ]:=
pdf

```
c_i_,j_@t:Σ_B[{___,i_,j_,___},___][__]:=Cordon_j@t
c_i_,j_@t:Σ_B[{j_,___,i_},___][__]:=Cordon_j@t
c_i_,j_@t:Σ_B[{___,j_,i_,___},___][__]:=Cordon_i@t
c_i_,j_@t:Σ_B[{i_,___,j_},___][__]:=Cordon_i@t
```

(Alt) In[ ]:=
pdf

```
mc[E_]:=E//.
  t:Σ_B[{___,i_,___},{___,j_,___},___][__]|Σ_B[{___,i_,j_,___},___][__]|Σ_B[{j_,___,i_},___][__]/;
    i+j==0:>c_i,j@t
```

(Alt) In[ ]:=
pdf

```
Kas@P_i_,j_:=CF@Σ_B[{i,j}][0, PQ[{}, 0]];
TL@P_i_,j_:=CF@Σ_B[{i,j}][0, PQ[{}, 0]]
```

(Alt) In[ ]:=
pdf

```
Kas[x:X[i_, j_, k_, l_]]:=Kas@If[PositiveQ[x], X_-i,j,k,-l, X̄_-j,k,l,-i];
Kas[(x:X|X̄)_fs__]:=Module[{v=2u^2-1, p, γs, m},
  γs=γ_#&/@{fs}; p=(x===X);
  m=If[p, ((v u 1 u)(u 1 u 1)(1 u v u)(u 1 u 1)), -((v u 1 u)(u 1 u 1)(1 u v u)(u 1 u 1))];
  CF@Σ_B[{fs}][If[p, -1, 1], PQ[{}, γs*.m.γs]]]
```

(Alt) In[ ]:=
pdf

```
TL[x : X[i_, j_, k_, l_]] := TL@If[PositiveQ[x], X_{-i,j,k,-l}, X̄_{-j,k,l,-i}];
TL[(x : X | X̄)_{fs__}] := Module[{t = 1 - ω, r, γs, m},
  r = t + t*; γs = γ_# & /@ {fs};
  m = If[x === X,
    ( -r  -t  2t  t*
      -t*  0   t*  0
      2t*  t   -r  -t*
       t   0   -t  0 ),
    (  r   -t  -2t*  t*
      -t*  0    t*   0
      -2t  t    r    -t*
       t   0    -t   0 )];
  CF@Σ_{B[{fs}]}[0, PQ[{}, γs*.m.γs]]]
```

(Alt) In[ ]:=
pdf

```
Kas[K_] := Fold[mc[#1 ⊕ #2] &, Σ_{B[]}[0, PQ[{}, 0]], List @@ (Kas /@ PD@K)];
KasSig[K_] := KasSig[K] = Expand[Kas[K][[1]] / 2]
```

(Alt) In[ ]:=
pdf

```
TL[K_] := Fold[mc[#1 ⊕ #2] &, Σ_{B[]}[0, PQ[{}, 0]], List @@ (TL /@ PD@K)] /.
   θ[c_ + u] /; Abs[c] ≥ 1 :↦ θ[c];
TLSig[K_] := TLSig[K] = TL[K][[1]]
```

In[ ]:= `TLSig[Knot[3, 1]]`

Out[ ]=

$$-2\,\theta\left[-\frac{\sqrt{3}}{2}+u\right]+2\,\theta\left[\frac{\sqrt{3}}{2}+u\right]$$

In[ ]:= `Monitor[Table[K → TLSig[K], {K, AllKnots[{3, 10}]}], K]`

Out[ ]=

In[ ]:= `Table[K → TLSig[K], {K, AllKnots[{3, 10}]}] >> TLSigData10.m`

In[ ]:= `CountDistinct[Table[TLSig[K], {K, AllKnots[{3, 10}]}]]`

Out[ ]=

141

In[ ]:= `CountDistinct[Table[Alexander[K][T], {K, AllKnots[{3, 10}]}]]`

Out[ ]=

211

In[ ]:= `CountDistinct[Table[{TLSig[K], Alexander[K][T]}, {K, AllKnots[{3, 10}]}]]`

Out[ ]=

224

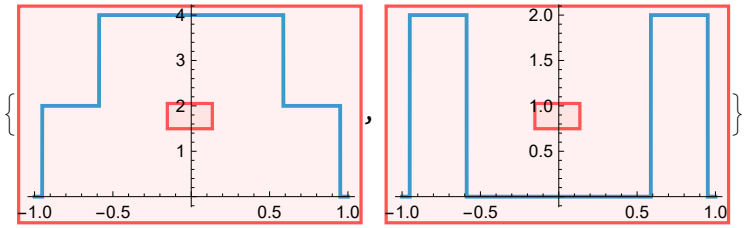
```
In[•]:= vals = Sort@Table[{Alexander[K][T], TLSig[K], K}, {K, AllKnots[{3, 10}]}];
        Table[
          pair = vals[[{k, k + 1}]];
          If[pair[[1, 1]] === pair[[2, 1]] ∧ pair[[1, 2]] =!= pair[[2, 2]], Last /@ pair,],
          {k, NumberOfKnots[10] - 1}
         ] // DeleteCases[Null]
```

```
Out[•]=
{{Knot[7, 4], Knot[9, 2]}, {Knot[10, 147], Knot[8, 11]},
 {Knot[10, 136], Knot[8, 21]}, {Knot[5, 1], Knot[10, 132]}, {Knot[7, 5], Knot[10, 130]},
 {Knot[10, 25], Knot[10, 56]}, {Knot[10, 87], Knot[10, 98]}, {Knot[8, 5], Knot[10, 141]},
 {Knot[10, 127], Knot[10, 150]}, {Knot[10, 59], Knot[9, 40]}}
```

```
In[•]:= pair = {Knot[5, 1], Knot[10, 132]};
        Alexander[#][T] & /@ pair
        Plot[TLSig[#], {u, -1, 1}] & /@ pair
```

```
Out[•]=
{1 + 1/T^2 - 1/T - T + T^2, 1 + 1/T^2 - 1/T - T + T^2}
```

```
Out[•]=
```




```
In[•]:= Monitor[Table[K → TLSig[K], {K, AllKnots[{3, 12}]}], K]
```

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.

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

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

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

```
Out[•]=
```

```
In[•]:= Table[K → TLSig[K], {K, AllKnots[{3, 12}]}] >> TLSigData12.m
```

```
In[•]:= Monitor[Table[K → TLSig[K], {K, AllKnots[{3, 13}]}], K]
```

KnotTheory: Loading precomputed data in KnotTheory/13N.dts.

```
Out[•]=
```

```
Table[K → TLSig[K], {K, AllKnots[{3, 13}]}] >> "Data/3-13.m"
```

```
In[•]:= NumberOfKnots[14, Alternating]
```

```
Out[•]=
19 536
```

```
Monitor[Table[K → TLSig[K], {K, AllKnots[14, Alternating][[1 ;; 10 000]]}] >>
  "Data/14a1-10000.m", K]
```

```
Monitor[Table[K → TLSig[K], {K, AllKnots[14, Alternating][[10 001 ;; 19 536]]}] >>
  "Data/14a10001-19536.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/14A.dts.

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

```
In[•]:= NumberOfKnots[14, NonAlternating]
```

```
Out[•]=
27 436
```

```
In[•]:= Monitor[Table[K → TLSig[K], {K, AllKnots[14, NonAlternating][[1 ;; 10 000]]}] >>
  "Data/14n1-10000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/14N.dts.

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

```
In[•]:= Monitor[Table[K → TLSig[K], {K, AllKnots[14, NonAlternating][[10 001 ;; 20 000]]}] >>
  "Data/14n10001-20000.m", K]
```

```
In[•]:= Monitor[Table[K → TLSig[K], {K, AllKnots[14, NonAlternating][[20 001 ;; 27 436]]}] >>
  "Data/14n20001-27436.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/14N.dts.

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

```
In[•]:= NumberOfKnots[15, Alternating]
```

```
Out[•]=
85 263
```

```
(Alt) In[•]:=
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[1 ;; 10 000]]}] >>
  "Data/15a1-10000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

(Alt) In[ ]:=
```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[10 001 ;; 20 000]]}] >>
  "Data/15a10001-20000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

(Alt) In[ ]:=
```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[20 001 ;; 30 000]]}] >>
  "Data/15a20001-30000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

(Alt) In[ ]:=
```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[30 001 ;; 35 000]]}] >>
  "Data/15a30001-35000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

(Alt) In[ ]:=
```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[35 001 ;; 40 000]]}] >>
  "Data/15a35001-40000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

(Alt) In[ ]:=
```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[40 001 ;; 50 000]]}] >>
  "Data/15a40001-50000.m", K]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[50 001 ;; 60 000]]}] >>
  "Data/15a50001-60000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[60 001 ;; 70 000]]}] >>
  "Data/15a60001-70000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[70 001 ;; 80 000]]}] >>
  "Data/15a70001-80000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, Alternating][[80 001 ;; 85 263]]}] >>
  "Data/15a10001-85263.m", K]
```

In[ ]:=
```
NumberOfKnots[15, NonAlternating]
```

Out[ ]=
```
168 030
```

```
Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[1 ;; 10 000]]}] >>
  "Data/15n1-10000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[10 001 ;; 20 000]]}] >>
  "Data/15n10001-20000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[20 001 ;; 30 000]]}] >>
  "Data/15n20001-30000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[30 001 ;; 40 000]]}] >>
  "Data/15n30001-40000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[40 001 ;; 50 000]]}] >>
  "Data/15n40001-50000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[50 001 ;; 60 000]]}] >>
  "Data/15n50001-60000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[60 001 ;; 70 000]]}] >>
  "Data/15n60001-70000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[70 001 ;; 80 000]]}] >>
  "Data/15n70001-80000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[80 001 ;; 90 000]]}] >>
  "Data/15n80001-90000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[90 001 ;; 100 000]]}] >>
  "Data/15n90001-100000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[100 001 ;; 110 000]]}] >>
  "Data/15n100001-110000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[110 001 ;; 120 000]]}] >>
  "Data/15n110001-120000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[120 001 ;; 130 000]]}] >>
  "Data/15n120001-130000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[130 001 ;; 140 000]]}] >>
  "Data/15n130001-140000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[140 001 ;; 150 000]]}] >>
  "Data/15n140001-150000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[150 001 ;; 160 000]]}] >>
  "Data/15n150001-160000.m", K]

Monitor[Table[K → TLSig[K], {K, AllKnots[15, NonAlternating][[160 001 ;; 168 030]]}] >>
  "Data/15n160001-168030.m", K]
```

(Alt) In[ ]:=

```
Ks = AllKnots[{3, 15}];
$M = 100;
Monitor[
  Do[
   Get[f] /. (K_ → V_) :> (TLSig[K] = V),
   {f, FileNames["Data/*.m"]}
  ],
  "Reading " <> f
 ];
Monitor[
 Do[
  f = "Data/15X" <> ToString[r] <> "mod" <> ToString[$M] <> ".m";
  If[! FileExistsQ[f],
   Put[
    Table[
     (K = Ks[[k + 1]]) → TLSig[K],
     {k, r, Length[Ks] - 1, $M}
    ],
    f
   ]
  ],
  {r, 0, $M - 1}
 ],
 {f, k + 1, K}
]
```

KnotTheory: Loading precomputed data in KnotTheory/15A.dts.

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

KnotTheory: Loading precomputed data in KnotTheory/15N.dts.

(Alt) In[ ]:=

```
Monitor[
  Do[
   Get[f] /. (K_ → V_) :> (TLSig[K] = V),
   {f, FileNames["Data/*.m"]}
  ],
  "Reading " <> f
 ];
((# → TLSig[#]) & /@ AllKnots[{3, 15}]) >> "TLSigData.m"
```