First class on Friday September 10. Reading week is after Lecture 24. Regrets in blue, gaps in red.

Topics in Algebraic Topology I: Algebraic Knot Theory and Computation.

The destination will be "a poly-time computable strong knot invariant with good algebraic properties". But you will be taking the course for the journey, not for the destination: What are knots and what are some of the problems around them? Why care about "invariants with good algebraic properties"? What is the "Yang-Baxter equation"? What are "virtual tangles"? What are "Hopf algebras"? Why would a topologist care about computations in Heisenberg algebras more than most physicists? How does Gaussian integration, and how do Feynman diagrams, arise in pure algebra? What is the "Drinfel'd Double Procedure"? Are we there yet?

The professor for this class does not believe anything that he does unless it is coded and the code runs. A useful life skill you will learn here is that even the incredibly abstract can become a computer program, often with no loss to its beauty.

Lecture 1. Course Introduction.

Lecture 2. Knots and the Kauffman bracket. →R1.

Lecture 3. Mathematica and implementing the Kauffman bracket.

Following http://drorbn.net/syd3.

Lecture 4. A faster Kauffman bracket program.

Following http://drorbn.net/syd3, with `ThinPosition` from `WG.nb`.

Lecture 5. Tangles and planar algebras. →R2, →R3.

Lecture 6. Three basic problems: unknotting, genus, ribbon knots. Display a list of ribbon knots.

Lecture 7. Aside: the Seifert algorithm.

Lecture 8. Tangles and the three basic problems.

Lecture 9. The Yang-Baxter approach and the WG algebras.

Lecture 10. π_1 and WG.

Lecture 11. Implementation.

Lecture 12. Virtual tangles and rotational virtual tangles. Meta monoids.

Lecture 13. Quasi-triangular Hopf algebras and the Kerler algebra.

Lecture 14. Implementation.

Lecture 15. The Heisenberg algebra $\mathbb{H}$, hR_0, and the PBW principle.

Lecture 16. Generating functions and hm.

Lecture 17. Gaussians and compositions.

Lecture 18. Implementation, testing.

Lecture 19. Yang-Baxter. An aside on the harmonic oscillator.

Lecture 20. Γ calculus and the Alexander polynomial. →R4.

Lecture 21. hR_ϵ. Gaussian integration.

Lecture 22. Perturbation theory for Gaussian integration.

Lecture 23. Perturbation theory for Gaussian compositions.

Lecture 24. Implementation.

Lecture 25. The Rozansky-Overbay invariants.

Lecture 26. CU_0, QU_0, and the QU_0-calculus.

Lecture 27. Genus using QU_0.

Lecture 28. Fox-Milnor using QU_0?

Lecture 29. CU_ϵ, Wigner contractions, solvable approximation.

Lecture 30. OU tangles and the Drinfel'd Double procedure.

Lecture 31. the Taft algebra?

Following Montgomery, Schneider, "Skew derivations of finite-dimensional algebras and actions of the double of the Taft Hopf algebra"?

Lecture 32. QU_ϵ and P.

Lecture 33. The rest of the QU_ϵ structure.

Lecture 34. Implementation, verification, computation.

Lecture 35. QU_ϵ and genus.

Lecture 36. From QU_ϵ to hR_ϵ.

Regret 1. Khovanov homology.

Regret 2. Khovanov homology for tangles.

Regret 3. Other algebraic structures near knot theory: (monoidal) categories, braid groups. Also mention contraction (circuit) algebras, meta-monoids, meta-Hopf-algebras, quandles, ...

Regret 4. Oh there's so much more on the Alexander polynomial!!!

Pensieve header: The WG Algebra with care for conjugacy invariant sets and for tangles.

```
Once[<< KnotTheory`];
HL[ℰ_] :=
  Style[ℰ, Background → If[TrueQ@ℰ, ■, ■]];
χ[cond_] := If[TrueQ[cond], 1, 0];
```

```
Loading KnotTheory` version
  of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
```

```
DeclareGroup[G_Symbol, S_{n_}] := Module[{α, β, e},
  G /: Ord[G] = n!;
  G /: Elements[G] =
   PermutationCycles /@ (Permutations@Range@n);
  Do[G /: g[G, α] = e = Elements[G]⟦α⟧;
   G /: ind[G, e] = α,
   {α, Ord[G]} ];
  Do[G /: m[G, α, β] =
    ind[G, g[G, α]~PermutationProduct~g[G, β]],
   {α, Ord[G]}, {β, Ord[G]} ];
  Do[G /: inv[G, α] =
    ind[G, InversePermutation[g[G, α]]],
   {α, Ord[G]} ] ]
```

```
ConjugacyClasses[G_Symbol] :=
 Module[{sea = Range@Ord@G, ccs = {}, cc},
  While[Length@sea > 0,
   cc =
    Union[
     Table[m[G, inv[G, α], m[G, First@sea, α]],
      {α, Ord@G}]];
   AppendTo[ccs, cc];
   sea = Complement[sea, cc];
  ]; ccs]
```

```
g[α_] := g[$G, α]; ind[α_] := ind[$G, α];
m[α_, β_] := m[$G, α, β];
inv[α_] := inv[$G, α];
```

```
m_{i_,j_→k_}[ℰ_] :=
 Expand[ℰ] /. W_i[α_, β_] W_j[γ_, δ_] ⧴
   χ[m[α, β] == m[β, γ]] W_k[α, m[β, δ]]
```

```
(* $CIS is a Conjugation Invariant Set *)
R_{i_,j_} := Sum[W_i[α, 1] W_j[β, α], {α, $CIS},
   {β, $CIS}];
R̄_{i_,j_} := Sum[W_i[α, 1] W_j[β, inv@α], {α, $CIS},
   {β, $CIS}];
η_{i_} := Sum[W_i[α, 1], {α, $CIS}];
```

```
DeclareGroup[$G = S3, S_3];
Table[m[i, j], {i, Ord[$G]}, {j, Ord[$G]}] //
 MatrixForm
```

1	2	3	4	5	6
2	1	4	3	6	5
3	5	1	6	2	4
4	6	2	5	1	3
5	3	6	1	4	2
6	4	5	2	3	1

```
Elements[$G]
```

```
{Cycles[{}], Cycles[{{2, 3}}],
 Cycles[{{1, 2}}], Cycles[{{1, 2, 3}}],
 Cycles[{{1, 3, 2}}], Cycles[{{1, 3}}]}
```

```
ConjugacyClasses[$G]
```

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

```
Table[{
  Short[lhs = R_{1,2} R_{4,3} R_{5,6} // m_{1,4→1} // m_{2,5→2} // m_{3,6→3}],
  rhs = R_{2,3} R_{1,4} R_{5,6} // m_{1,5→1} // m_{2,6→2} // m_{3,4→3};
  HL[lhs == rhs]},
 {$CIS, ConjugacyClasses[$G]} ]
```

```
{{W_1[1, 1] W_2[1, 1] W_3[1, 1], True},
 {W_1[2, 1] W_2[2, 2] W_3[2, 1] + W_1[3, 1] W_2[3, 3] W_3[2, 1] +
   <<23>> + W_1[2, 1] W_2[3, 2] W_3[6, 5] +
   W_1[3, 1] W_2[6, 3] W_3[6, 5], True},
 {W_1[5, 1] W_2[4, 5] W_3[4, 1] + W_1[4, 1] W_2[5, 4] W_3[4, 1] +
   <<1>> + <<1>> + <<1>> + <<1>> + W_1[5, 1] W_2[5, 5]
    W_3[5, 4] + W_1[4, 1] W_2[4, 4] W_3[5, 5], True}}
```

```
Table[
 lhs = R_{1,2} R̄_{3,4} // m_{1,3→1} // m_{2,4→2};
 rhs = η_1 η_2 // Expand; HL[lhs == rhs],
 {$CIS, ConjugacyClasses[$G]} ]
```

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

```
Table[
 lhs = R_{1,2} R̄_{3,4} // m_{1,3→1} // m_{4,2→2}; rhs = η_1 η_2 // Expand;
 HL[lhs == rhs],
 {$CIS, ConjugacyClasses[$G]} ]
```

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

```
Z[K_] := Module[{z}, Sum[
   z = Expand[Times @@ PD[K] /.
      x : X[i_, j_, k_, l_] ⧴
       If[PositiveQ@x, R_{l,i}, R̄_{j,i}]];
   Do[z = z // m_{c⟦1⟧,c⟦j⟧→c⟦1⟧}, {c, Skeleton[K]},
    {j, 2, Length@c}];
   z,
   {$CIS, ConjugacyClasses[$G]}]]
```

Edge-vertex convention: an oriented edge carries the same label as the vertex ending it.

```
ZF[K_] := ZF[PD@K];
ZF[pd_PD] := Module[{z, done, st, c, mn, k},
  Sum[
   z = 1; done = {};
   st = Range[2 Length@pd];
   Do[
    z *= c /. X[i_, j_, _, l_] :>
       If[PositiveQ@c, mn = {i, l}; R_{l,i}, mn = {i, j};
        R̄_{j,i}];
    Do[
     If[MemberQ[done, k + 1], z = z // m_{k,k+1→k};
      st = st /. k + 1 → k];
     If[MemberQ[done, k - 1],
      z = z // m_{st⟦k-1⟧,k→st⟦k-1⟧};
      st = st /. k → st⟦k - 1⟧],
     {k, mn}];
    done = done ⋃ mn,
    {c, List @@ pd}];
   z,
   {$CIS, ConjugacyClasses[$G]}] ]
```

```
ThinPosition[K_] := Module[{todo, done, pd, c},
  todo = List @@ PD@K; done = {}; pd = PD[];
  While[todo =!= {},
   AppendTo[pd,
    c = RandomChoice@
      MaximalBy[todo, Length[done ⋂ List @@ #] &]];
   todo = DeleteCases[todo, c];
   done = done ⋃ List @@ c];
  pd
 ]
```

```
ZEF[K_] := ZF@ThinPosition@K;
```

To do: Implement and verify Δ etc.

Pensieve header: The Kerler Algebra and the Alexander polynomial.

```
Once[<< KnotTheory`];
HL[ℰ_] :=
  Style[ℰ, Background → If[TrueQ@ℰ, ■, ■]];
```

```
Loading KnotTheory` version
  of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
```

```
MT = ( □   a    b    c    d    ka   kb   kc   kd
       a   a    b    0    0    ka   kb   0    0
       b   0    0    a    b    0    0    -ka  -kb
       c   c    d    0    0    -kc  -kd  0    0
       d   0    0    c    d    0    0    kc   kd
       ka  ka   kb   0    0    a    b    0    0
       kb  0    0    ka   kb   0    0    -a   -b
       kc  kc   kd   0    0    -c   -d   0    0
       kd  0    0    kc   kd   0    0    c    d  );

ℰ_ // m_{i_,j_→k_} :=
  Expand[ℰ] /.
   Flatten@
    Table[MT⟦α, 1⟧_i MT⟦1, β⟧_j →
      (MT⟦α, β⟧ /. v : (a | b | c | d | ka | kb | kc | kd) :→
        v_k), {α, 2, 9}, {β, 2, 9}];
```

```
KBasis[{i_}] := {a_i, b_i, c_i, d_i, ka_i, kb_i, kc_i, kd_i};
KBasis[{i_, is__}] :=
 Flatten@Outer[Times, KBasis[{i}], KBasis[{is}]]
```

```
η_i_ := a_i + d_i;
ɣ_i_ := ka_i + kd_i;
```

```
lhs = η_1 KBasis[{2}] // m_{1,2→1};
HL[lhs == KBasis[{1}]]
```

True

```
lhs = η_1 KBasis[{2}] // Expand // m_{1,2→1};
HL[lhs == KBasis[{1}]]
```

True

```
Short[lhs = KBasis[{1, 2, 3}] // m_{1,2→1} // m_{1,3→1}]
rhs = KBasis[{1, 2, 3}] // m_{2,3→2} // m_{1,2→1};
lhs == rhs // HL
```

$\{a_1,\ b_1,\ 0,\ 0,\ ka_1,\ kb_1,\ 0,\ 0,\ 0,\ 0,\ a_1,\ b_1,\ 0,\ 0,\ -ka_1,\ \ll 482 \gg,\ d_1,\ 0,\ 0,\ -kc_1,\ -kd_1,\ 0,\ 0,\ 0,\ 0,\ c_1,\ d_1,\ 0,\ 0,\ kc_1,\ kd_1\}$

True

```
R_{i_,j_} := a_i a_j + d_i a_j + T a_i d_j - (1 - T) kc_i kb_j - T d_i d_j
```

```
Short[lhs = R_{1,2} R_{4,3} R_{5,6} // m_{1,4→1} // m_{2,5→2} // m_{3,6→3}];
rhs = R_{2,3} R_{1,4} R_{5,6} // m_{1,5→1} // m_{2,6→2} // m_{3,4→3};
lhs == rhs // HL
```

True

```
R̄_{i_,j_} := a_i a_j + d_i a_j + T^-1 a_i d_j - (1 - T^-1) kc_i kb_j -
  T^-1 d_i d_j
```

```
Short[lhs = R_{1,2} R̄_{3,4} // m_{1,3→1} // m_{2,4→2}]
rhs = η_1 η_2 // Expand;
lhs == rhs // HL
```

$a_1 a_2 + a_2 d_1 + a_1 d_2 + d_1 d_2$

True

```
Short[lhs = R_{1,2} R̄_{3,4} // m_{1,3→1} // m_{4,2→2}]
rhs = η_1 η_2 // Expand;
Simplify[lhs - rhs]
```

$a_1 a_2 + a_2 d_1 + a_1 d_2 + d_1 d_2 - 2\, kb_2\, kc_1 + 2\, T\, kb_2\, kc_1$

$2 \times (-1 + T)\, kb_2\, kc_1$

```
lhs = R_{1,4} R̄_{5,2} ɣ_3 // m_{2,4→2} // m_{1,3→1} // m_{1,5→1}
rhs = ɣ_1 η_2 // Expand;
lhs == rhs // HL
```

$a_2\, ka_1 + d_2\, ka_1 + a_2\, kd_1 + d_2\, kd_1$

True

RVK and Z

RVK, rot, Z modified from 2016-09/OneSmidgen.nb. See also in AP/Projects/SL2Invariant/.

Some details of the code below are at http://drorbn.net/bbs/show?shot=Dror-160920-151350.jpg.

```
RVK::usage =
  "RVK[xs, rots] represents a Rotational
    Virtual Knot with a list of n Xp/Xm
    crossings xs and a length 2n list of
    rotation numbers rots. Crossing sites
    are indexed 1 through 2n, and rots⟦k⟧
    is the rotation between site k-1 and
    site k. RVK is also a casting operator
    converting to the RVK presentation from
    other knot presentations.";
```

```
RVK[pd_PD] :=
  Module[{n, xs, x, rots, front = {0}, 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 = 0, k < 2 n, ++k,
    If[k == 0 ∨ FreeQ[front, -k],
     front = Flatten@Replace[front, k → (xs /. {
            Xp[k + 1, l_] | Xm[l_, k + 1] ⧴
             {l, k + 1, 1 - l},
            Xp[l_, k + 1] | Xm[k + 1, l_] ⧴
             (++rots〚l〛;
              {1 - l, k + 1, l}),
            _Xp | _Xm ⧴ {}
           }), {1}],
     Cases[front, k | -k] /.
       {k, -k} ⧴ --rots〚k + 1〛;
    ]];
   RVK[xs, rots] ];
RVK[K_] := RVK[PD[K]];
```

```
rot_i_[n_] := { η_i   EvenQ[n]
                γ_i   OddQ[n]
```

```
Z[K_] := Z[RVK@K];
Z[rvk_RVK] := Module[{ξ, done, st, c, χ, i, j, k},
  ξ = 1; done = {}; st = Range[2 Length[rvk〚1〛]];
  Do[
   {i, j} = List @@ c;
   χ = (c /. {_Xp ⧴ R_{i,j}, _Xm ⧴ R̄_{i,j}}) (ka_0 - kd_0) //
     m_{j,0→j};
   Do[χ = (rot_0[rvk〚2, k〛] χ) // m_{0,k→k},
    {k, {i, j}}];
   ξ *= χ;
   Do[
    If[MemberQ[done, k + 1], ξ = ξ // m_{k,k+1→k};
     st = st /. k + 1 → k];
    If[MemberQ[done, k - 1], ξ = ξ // m_{st〚k-1〛,k→st〚k-1〛};
     st = st /. k → st〚k - 1〛],
    {k, {i, j}}];
   done = done ⋃ {i, j},
   {c, rvk〚1〛}
  ];
  Factor@ξ
 ]
```

```
K = Knot[8, 17]; Factor@Alexander[K][T]
z = Z[K]
```

$$-\frac{1-4\,T+8\,T^2-11\,T^3+8\,T^4-4\,T^5+T^6}{T^3}$$

$$-\frac{\left(1-4\,T+8\,T^2-11\,T^3+8\,T^4-4\,T^5+T^6\right)\,(a_1+d_1)}{T^4}$$

```
Timing@
 Union@Table[Simplify[Alexander[K][T] (a_1 + d_1) / Z[K]],
   {K, AllKnots[{3, 10}]}]
```

$$\left\{44.8281,\ \left\{1,\ \frac{1}{T^5},\ \frac{1}{T^4},\ \frac{1}{T^3},\ \frac{1}{T^2},\ \frac{1}{T},\ T,\ T^2,\ T^3,\ T^4,\ T^5,\ T^6\right\}\right\}$$

```
ThinPosition[K_] := Module[{todo, done, pd, c},
  todo = List @@ PD@K; done = {}; pd = PD[];
  While[todo =!= {},
   AppendTo[pd,
    c = RandomChoice@
      MaximalBy[todo, Length[done ⋂ List @@ #] &]];
   todo = DeleteCases[todo, c];
   done = done ⋃ List @@ c];
  pd ]
```

```
Timing@
 Union@
  Table[Simplify[Alexander[K][T] (a_1 + d_1) / ZF[K]],
   {K, AllKnots[{3, 10}]}]
```

$$\left\{11.3281,\ \left\{1,\ \frac{1}{T^5},\ \frac{1}{T^4},\ \frac{1}{T^3},\ \frac{1}{T^2},\ \frac{1}{T},\ T,\ T^2,\ T^3,\ T^4,\ T^5,\ T^6\right\}\right\}$$