Random Knots from TurboKnots

T. Wolf, Brock University

April 2, 2024

Outline

Outline

Disclaimer

The talk

Disclaimer

The talk
- does not present new mathematics,

Disclaimer

The talk
- does not present new mathematics,
- is about a tool to work on knot diagrams.

Disclaimer

The talk
- does not present new mathematics,
- is about a tool to work on knot diagrams.

TurboKnots has only a fraction of the functionality of packages like KnotTheory or Regina.

Disclaimer

The talk
- does not present new mathematics,
- is about a tool to work on knot diagrams.

TurboKnots has only a fraction of the functionality of packages like KnotTheory or Regina.

One of its strength is its ability to simplify diagrams which is useful, for example, when computing knot invariants.

Outline

Initial Motivation

- Caribou Contests: online math contests for grades K to 12, run 6 times per year for all grades

Initial Motivation

- Caribou Contests: online math contests for grades K to 12, run 6 times per year for all grades
- Online nature allows interactive questions and possibility to introduce new math to school students.

Initial Motivation

- Caribou Contests: online math contests for grades K to 12, run 6 times per year for all grades
- Online nature allows interactive questions and possibility to introduce new math to school students.
- April 2021 contests featured the first interactive knot challenge to unknot/simplify diagrams or reduce their number of crossings for 25,000 students of grades 2 to 12: grade 2 67%, grade 3/4 38%, grade 5/6 51%, grade 7/8 14%, grade 9/10 7.4%, grade 11/12 6.6%

Initial Motivation

- Caribou Contests: online math contests for grades K to 12, run 6 times per year for all grades
- Online nature allows interactive questions and possibility to introduce new math to school students.
- April 2021 contests featured the first interactive knot challenge to unknot/simplify diagrams or reduce their number of crossings for 25,000 students of grades 2 to 12: grade 2 67%, grade 3/4 38%, grade 5/6 51%, grade 7/8 14%, grade 9/10 7.4%, grade 11/12 6.6%
- Questions are announced 2 weeks before each contest to study 'Food for Thought'

Initial Motivation

- Caribou Contests: online math contests for grades K to 12, run 6 times per year for all grades
- Online nature allows interactive questions and possibility to introduce new math to school students.
- April 2021 contests featured the first interactive knot challenge to unknot/simplify diagrams or reduce their number of crossings for 25,000 students of grades 2 to 12: grade 2 67%, grade 3/4 38%, grade 5/6 51%, grade 7/8 14%, grade 9/10 7.4%, grade 11/12 6.6%
- Questions are announced 2 weeks before each contest to study 'Food for Thought'
- May 2023 contests featured an interactive knot colouring challenges for 14,800 students of grades 7 to 12: grade 7/8 44%, grade 9/10 39%, grade 11/12 24%

Outline

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation),

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform, generate,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform, generate, simplify,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform, generate, simplify, compactify,

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform, generate, simplify, compactify, decompose

Basic Tools of TurboKnots

- Initial Goal: general knot workbench to manipulate knot diagrams,
- Runs under linux and linux simulated under windows and Mac,
- Display uses box drawing characters: fast, compact, allows computations done on display and thus compute annimations of deformation sequences
- Initial tools: draw, paint, load, save, export (LaTeX, html, animation), edit, deform, generate, simplify, compactify, decompose and identify diagrams.

Outline

Visualization, Scope

Copying the Terminal: y

Zooming in/out: Ctrl +/- z (after resizing terminal)

Working with a stack of diagrams: + PgUp/Dn Del

Shifting the visible Window: S-↑, S-↓, S-←, S-→

Generating and Simplifying Random Knot Diagrams

r R [Ctrl r]
l b kn/big/128 [⏎]
d s S-1 [🖱]
s S-2 [🖱]
d S-1 [🖱]
d S-2 [🖱]
d S-4 [🖱]
s S-3 [🖱]
d S-3 [🖱]
d S-4 [🖱]
d S-1 [🖱]
d S-5 [🖱]
d S-4 [🖱]
p P [🖱]
[🖱] [[PgDn]] [[⌫]]
l b [⏎] d m [🖱]

Outline

Outline

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Example:
l b kn/bigprime/21 ⏎
c (characterization) c (colouring)
1 (computing the Smith Normal Form)

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Example:
l b kn/bigprime/21 ⏎
c (characterization) c (colouring)
1 (computing the Smith Normal Form)
$\to 1049879229 = 3 \times 7 \times 23 \times 2173663$
$3 = 3$
$3 = 3$
$3 = 3$

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Example:
l b kn/bigprime/21 ⏎
c (characterization) c (colouring)
1 (computing the Smith Normal Form)

$\rightarrow 1049879229 = 3 \times 7 \times 23 \times 2173663$

$3 = 3$

$3 = 3$

$3 = 3$

$\rightarrow$ Colouring number 3 has multiplicity 4.

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Example:

l b kn/bigprime/21 ⏎

c (characterization) c (colouring)

1 (computing the Smith Normal Form) ↺

$\rightarrow$ $1049879229 = 3 \times 7 \times 23 \times 2173663$

$3 = 3$

$3 = 3$

$3 = 3$

$\rightarrow$ Colouring number 3 has multiplicity 4.

c 2 3 (colour dia with number 3) ⏎ 2 [⏎] ↺

Knot Colouring I

Computing the Smith Normal Form of coeff matrix gives colouring numbers.

Example:
l b kn/bigprime/21 [↵]
c (characterization) c (colouring)
1 (computing the Smith Normal Form) [⟲]
→ $1049879229 = 3 \times 7 \times 23 \times 2173663$
$3 = 3$
$3 = 3$
$3 = 3$
→ Colouring number 3 has multiplicity 4.

c 2 3 (colour dia with number 3) [↵] 2 [[↵]] [⟲]
→ May be useful to visualize tangles or intertwined prime knots

Knot Colouring II

l b kn/bigprime/38

c c 1

Knot Colouring II

l b kn/bigprime/38 [↵]
c c 1 [⟲] [⟲]
$\rightarrow 18167191515 = 3 \times 3 \times 3 \times 3 \times 3 \times 5 \times 11 \times 1359311$

Knot Colouring II

I b kn/bigprime/38 ⏎
c c 1 ⟲ ⟲
$\rightarrow$ $18167191515 = 3 \times 3 \times 3 \times 3 \times 3 \times 5 \times 11 \times 1359311$
$\rightarrow$ Colouring numbers 3 and $3^5 = 243$ have both multiplicity 1.

Knot Colouring III

l b kn/bigprime/39 ⏎
c c 1

Knot Colouring III

l b kn/bigprime/39 ⏎
c c 1 ⟲ ⟲
$\rightarrow$ 921474469 is a prime number

Knot Colouring III

l b kn/bigprime/39 ⏎

c c 1 ↺ ↺

→ 921474469 is a prime number

Colour smaller knots visually: n n 8_18 ⏎ ⏎

c c 1 ↺

c 2 3 ⏎ (colour diagram) 2 ↺ ↺

Knot Colouring III

l b kn/bigprime/39 ↵
c c 1 ⟲ ⟲
→ 921474469 is a prime number

Colour smaller knots visually: n n 8_18 ↵ ↵
c c 1 ⟲
c 2 3 ↵ (colour diagram) 2 ⟲ ⟲

Display of the growth of colouring invariants: a o ↵

Outline

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

After each skein relation a complete simplification of the whole link is done which performs

- generalized beneficiary Reidemeister 1,2,3 moves,

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

After each skein relation a complete simplification of the whole link is done which performs

- generalized beneficiary Reidemeister 1,2,3 moves,
- reducing pass moves,

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

After each skein relation a complete simplification of the whole link is done which performs

- generalized beneficiary Reidemeister 1,2,3 moves,
- reducing pass moves,
- recognition of disconnected link components and link decomposition when being connected through a Hopf link,

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

After each skein relation a complete simplification of the whole link is done which performs

- generalized beneficiary Reidemeister 1,2,3 moves,
- reducing pass moves,
- recognition of disconnected link components and link decomposition when being connected through a Hopf link,
- prime knot extractions of prime knots lying above or below the remainder of the link,

HOMFLY-PT computations I

... with all their simplifications are based on a graph encoding without coordinates.

Each crossing has a handedness and is made of an under- and over-pass which each have a predecessor, a successor and opposite pass.

After each skein relation a complete simplification of the whole link is done which performs

- generalized beneficiary Reidemeister 1,2,3 moves,
- reducing pass moves,
- recognition of disconnected link components and link decomposition when being connected through a Hopf link,
- prime knot extractions of prime knots lying above or below the remainder of the link,
- direct HOMFLY-PT polynomial substitution of the Hopf Link and of knots $3_1, 4_1, 5_1, 5_2$.

HOMFLY-PT computations II

File name in bigprime/	# of crossings	TurboKnots time	Regina time
23	49	2 s	0 s
47	54	11 s	0 s
38	64	21 s	0.5 s
21	68	1:10 min	0.5 s
20	83	5:20 min	21 s

Regina is faster for larger prime knots

HOMFLY-PT computations III

File name in composed/	# of crossings	TurboKnots time	Regina time
pktest	78	0 s	0 s
pk0	6	0 s	0 s
pk1	21	0 s	0 s
pk2	25	0 s	0 s
pk3	38	0 s	0.4 s
pk4	38	0 s	0.6 s
pk5	52	0 s	3.8 s
pk6	78	12 s	21 s

TurboKnots is slightly faster for composite knots

HOMFLY-PT computations IV

File name in ukn/0/	# of crossings	TurboKnots time	Regina time
TuzunSikora	21	0 s	0 s
SikoraTuzun	23	0 s	0 s
Gordian Gordian	141	0 s	-

These are unknots

HOMFLY-PT computations V

File name in big/	# of crossings	TurboKnots time	Regina time
98	754	0 s	2.2 s
0	794	0 s	1.3 s
48	816	0 s	1 s
102	832	0 s	1 s
44	867	0 s	1 s
86	869	0 s	2.4 s
76	873	0 s	2 s
38	888	0 s	1.4 s

TurboKnots is faster for knots that can be heavily simplified.

HOMFLY-PT computations VI

... read from disk all 313230 knots with up to 15 crossings and compute and verify HOMFLYPT computation in 3:10 min,

HOMFLY-PT computations VI

... read from disk all 313230 knots with up to 15 crossings and compute and verify HOMFLYPT computation in 3:10 min,

.. simplify the Gordian (un-)knot diagram with 141 crossings and compute the polynomial to 1 in about 1 ms.

Outline

Computing Unknotting Numbers I

Usual definition: The unknotting number of a knot is n if there exists a diagram where n crossings switched (at the same time!) give the unknot.

Computing Unknotting Numbers I

Usual definition: The unknotting number of a knot is n if there exists a diagram where n crossings switched (at the same time!) give the unknot.

It had been conjectured by Bernhard-Jablan that a minimal unknotting sequence could be determined starting with some minimal crossing diagram for a knot. This has been shown to be false by Brittenham and Hermiller.

Computing Unknotting Numbers II

The 'unknotting number' computed in TurboKnots is recursively defined. It is n if there exists a minimal diagram of that knot (with minimal number of crossings) that has at least one crossing which being switched results in a knot with unknotting number $n - 1$.

Computing Unknotting Numbers II

The 'unknotting number' computed in TurboKnots is recursively defined. It is n if there exists a minimal diagram of that knot (with minimal number of crossings) that has at least one crossing which being switched results in a knot with unknotting number $n - 1$.

This gives the correct value if for each knot at least one minimal diagram has a crossing which being switched lowers the unknotting number.

Computing Unknotting Numbers III

Procedure:

- Know all minimal diagrams for all knots. That can be many, e.g. knot $15n_{2100}$ has (at least) 3986.

Computing Unknotting Numbers III

Procedure:

- Know all minimal diagrams for all knots. That can be many, e.g. knot $15n_{2100}$ has (at least) 3986.
- After each switch simplify the diagram to be minimal. This requires P0 moves otherwise, for example, for $12a_{39}$, $12a_{818}$, $12n_{47}$ an unknotting number of 3 is obtained instead of 2.

Computing Unknotting Numbers III

Procedure:

- Know all minimal diagrams for all knots. That can be many, e.g. knot $15n_{2100}$ has (at least) 3986.
- After each switch simplify the diagram to be minimal. This requires P0 moves otherwise, for example, for $12a_{39}$, $12a_{818}$, $12n_{47}$ an unknotting number of 3 is obtained instead of 2.
- Identify the prime knot or composite knot through its prime knots and look their unknotting number up in the database. Continue recursively if their unknotting number is not known yet.

Computing Unknotting Numbers III

Procedure:

- Know all minimal diagrams for all knots. That can be many, e.g. knot $15n_{2100}$ has (at least) 3986.
- After each switch simplify the diagram to be minimal. This requires P0 moves otherwise, for example, for $12a_{39}$, $12a_{818}$, $12n_{47}$ an unknotting number of 3 is obtained instead of 2.
- Identify the prime knot or composite knot through its prime knots and look their unknotting number up in the database. Continue recursively if their unknotting number is not known yet.
- Store the computed unknotting number in the database.

Computing Unknotting Numbers IV

Comments:

We also apply the unproven but commonly accepted hypothesis that the unknotting number of a composite knot is the sum of the unknotting numbers of its prime knots.

Computing Unknotting Numbers IV

Comments:

We also apply the unproven but commonly accepted hypothesis that the unknotting number of a composite knot is the sum of the unknotting numbers of its prime knots.

The recent computation (Apr 2023) is based on simplification programs using the layout of the diagram (coordinates). Computations performed on the graph encoding would be much faster.

Computing Unknotting Numbers IV

Comments:

We also apply the unproven but commonly accepted hypothesis that the unknotting number of a composite knot is the sum of the unknotting numbers of its prime knots.

The recent computation (Apr 2023) is based on simplification programs using the layout of the diagram (coordinates). Computations performed on the graph encoding would be much faster.

As a by-product for each knot not only the unknotting number but also the minimal and the maximal number of simplifying crossings of all minimal diagrams are determined and stored.

Computing Unknotting Numbers V

Rare cases which become more frequent with increasing crossing number:

- Minimal diagrams have no single simplifying switch.

Computing Unknotting Numbers V

Rare cases which become more frequent with increasing crossing number:

- Minimal diagrams have no single simplifying switch.

  Example: $11n_{64}$ has 14 minimal diagrams. For 2 diagrams all switches result in knots with unknotting number 2, 6 diagrams have 2 switches which result in a knot with unknotting number 1 and 6 diagrams have a single switch resulting in a knot with unknotting number 1.

Computing Unknotting Numbers V

Rare cases which become more frequent with increasing crossing number:

- Minimal diagrams have no single simplifying switch.

  Example: $11n_{64}$ has 14 minimal diagrams. For 2 diagrams all switches result in knots with unknotting number 2, 6 diagrams have 2 switches which result in a knot with unknotting number 1 and 6 diagrams have a single switch resulting in a knot with unknotting number 1.

- An unknotting number reducing switch does not reduce the crossing number.

Computing Unknotting Numbers V

Rare cases which become more frequent with increasing crossing number:

- Minimal diagrams have no single simplifying switch.

  Example: $11n_{64}$ has 14 minimal diagrams. For 2 diagrams all switches result in knots with unknotting number 2, 6 diagrams have 2 switches which result in a knot with unknotting number 1 and 6 diagrams have a single switch resulting in a knot with unknotting number 1.

- An unknotting number reducing switch does not reduce the crossing number.

  Example: 10_{139} with $u = 4$ has a diagram where 2 simplifying switches each change the knot into 10_{161} with $u = 3$ which has a diagram where 2 simplifying switches each change the knot into 10_{145} with $u = 2$.

Outline

Downloads

TurboKnots:
Download: `wget https://cariboutests.com/games/knots/TurboKnots.tar.gz`
Unpack: `tar xfz TurboKnots.tar.gz`
Call: `./TurboKnots`
Help: `./TurboKnots '?'`

Colouring numbers and multiplicities:
https://cariboutests.com/games/knots/colour3-15-N.txt

HOMFLYPT polynomials:
https://cariboutests.com/games/knots/HOMFLY3-15.txt

Unknotting numbers:
https://cariboutests.com/games/knots/uk3-15.txt

Overview of the 3 data files:
https://cariboutests.com/games/knots/readme.txt

The End

Thank you!