Pensieve header: This notebook contains some testing code for the package KnotTheory`.

KnotTheory` Testing.nb

In[•]:=
```
SetAttributes[Assert, HoldAll];
Assert[fact_] := Module[
   {result = fact},
   If[result === True,
    True,
    Print[
     StringTake[ToString[Hold[fact]], {6, -2}],
     " yields ", fact]
   ]
  ];
```

In[•]:=
```
<< KnotTheory`
```

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

In[•]:=
```
Assert[Jones[Link["L11n458"]][q] == Jones[DTCode[Link["L11n458"]]][q]]
```

KnotTheory: Loading precomputed data in Jones4Links`.

KnotTheory: Loading precomputed data in PD4Links`.

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

Out[•]= True

In[•]:=
```
Assert[Kh[PD[Knot[3, 1]]][q, t] == 1/q^3 + 1/q + 1/(q^9 t^3) + 1/(q^5 t^2)]
```

KnotTheory: Loading precomputed data in PD4Knots`.

KnotTheory: The Khovanov homology program JavaKh–v2 is an update of Jeremy Green's program JavaKh–v1, written by Scott Morrison in 2008 at Microsoft Station Q.

Out[•]= True

In[•]:=
```
Assert[Kh[PD[Knot[4, 1]], Program → "FastKh"][q, t] == 1/q + q + 1/(q^5 t^2) + 1/(q t) + q t + q^5 t^2]
```

KnotTheory: The Khovanov homology program FastKh was written by Dror Bar–Natan.

Out[•]= True

In[•]:=
```
Assert[Last[Kh[TorusKnot[6, 5], Modulus → Null][q, t]] == q^39 t^12 ZMod[2, 5]]
```

Out[•]= True

In[•]:=
```
Assert[Alexander["K11a44", 2][t] === {1 - t + t^2}]
```

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: The program Alexander[K, r] to compute Alexander ideals was written by Jana Archibald at the University of Toronto in the summer of 2005.

Out[•]= True

In[•]:=
```
Assert[
 Alexander["10_99", #][t] & /@ {1, 2} ===
  {{1 - 4 t + 10 t^2 - 16 t^3 + 19 t^4 - 16 t^5 + 10 t^6 - 4 t^7 + t^8}, {1 - 2 t + 3 t^2 - 2 t^3 + t^4}}
]
```

Out[•]= True

In[•]:=
```
Assert[
 ArcPresentation["K11n11"] === ArcPresentation[{12, 2}, {1, 10}, {3, 9},
   {5, 11}, {9, 12}, {4, 8}, {2, 5}, {11, 7}, {8, 6}, {7, 4}, {10, 3}, {6, 1}]
]
```

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

Out[•]= True

In[•]:=
```
Assert[
 BR[ArcPresentation[
    {12, 2}, {1, 10}, {3, 9}, {5, 11},
    {9, 12}, {4, 8}, {2, 5}, {11, 7}, {8, 6}, {7, 4}, {10, 3}, {6, 1}]
  ] === BR[4, {-1, 2, 3, 3, -2, -1, -2, -1, 2, 3, 3, 3, 2}]
]
```

KnotTheory: Vogel's algorithm was implemented by Dan Carney in the summer of 2005 at the University of Toronto.

Out[•]= True

In[•]:=
```
Assert[(HFKHat[Knot[8, 19]][t, m] /. m → 1/t) == 4/t^3 + 1/t^2]
```

KnotTheory: Loading precomputed data in HFKHat4KnotsTo11`.

Out[•]= True

```
In[•]:= Assert[(HFKHat[BR[Knot[8, 19]]][t, m] /. m → 1/t) == 4/t^3 + 1/t^2]
```

KnotTheory: The minimum braids representing the knots with up to 10 crossings were provided by Thomas Gittings. See arXiv:math.GT/0401051.

KnotTheory: The HFKHat program was written by Jean–Marie Droz in 2007 at the University of Zurich, based on methods of Anna Beliakova's arXiv:07050669.

```
Hold[(HFKHat[BR[Knot[8, 19]]][t, m] /. m -> -) == 1/t - 4/t^3 + 1/t^2]
```

yields $Failed[t, 1/t] == 4/t^3 + 1/t^2

```
In[•]:= Assert[(Plus @@ (AlternatingQ /@ AllKnots[8])) == 3 False + 18 True]
Out[•]= True

In[•]:= Assert[AlternatingQ[Knot[0, 1]] == True]
Out[•]= True

In[•]:= Assert[
 MultivariableAlexander[Link["L7a7"]][t] == MultivariableAlexander[PD[Link["L7a7"]]][t]
]
```

KnotTheory: Loading precomputed data in MultivariableAlexander4Links`.

KnotTheory: Loading precomputed data in PD4Links`.

KnotTheory: The multivariable Alexander program "MVA2" was written by Jana Archibald at the University of Toronto in 2007–2008.

```
Out[•]= True

In[•]:= Flip[X[i_, j_, k_, l_]] := If[l == j + 1 || j - l > 1, X[j, k, l, i], X[l, i, j, k]];
VassilievCube[{}, pdcomb_] := pdcomb;
VassilievCube[{i_, rest___}, pdcomb_] := VassilievCube[{rest},
  Expand[pdcomb - (pdcomb /. pd_PD :> MapAt[Flip, pd, i])]
 ]

In[•]:= Assert[Equal[
  Series[
     VassilievCube[{1, 2, 5, 7}, PD[#]]
       /. pd_PD :> MultivariableAlexander[pd][t] /. t[i_] → E^(h x[i]),
     {h, 0, 2}
    ] & /@ AllLinks[7],
  {O[h]^3, O[h]^3, O[h]^3, O[h]^3, O[h]^3, O[h]^3, O[h]^3, O[h]^3, O[h]^3}
 ]]
Out[•]= True
```

```
In[•]:= Assert[Equal[
          Series[
           VassilievCube[{1, 2, 5, 7}, PD[Link[7, Alternating, 1]]]
             /. pd_PD :> MultivariableAlexander[pd, Program → "MVA1"][t] /.
            t[i_] → E^(h x[i]),
           {h, 0, 2}
          ],
          -2 x[1] h + (-2 x[1]^2 - x[1] x[2]) h^2 + O[h]^3
         ]]
```

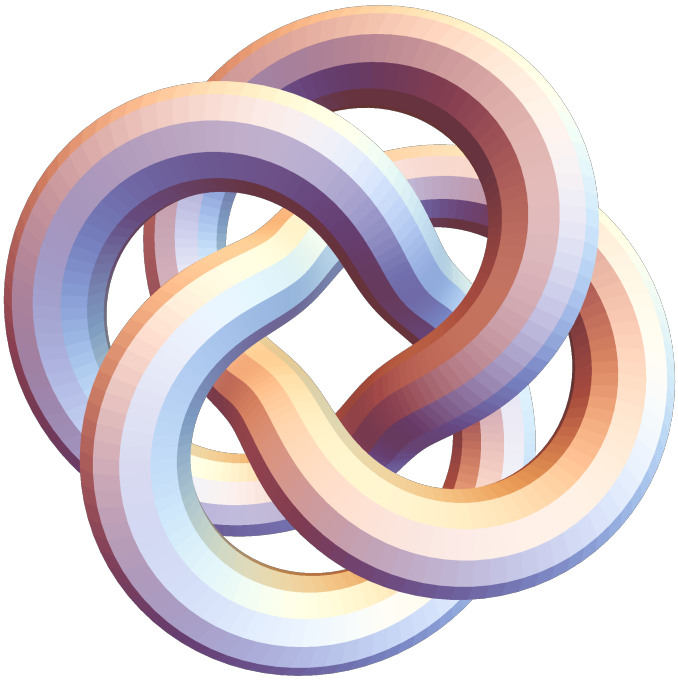
KnotTheory: The multivariable Alexander program "MVA1" was written by Dan Carney at the University of Toronto in the summer of 2005.

```
Out[•]= True
```

The Test Part - Graphics

```
In[•]:= TubePlot[TorusKnot[4, 3]] // Show
```

Out[•]=



In[•]:=
```
Draw[ArcPresentation[Knot[9, 17]]]
```

Out[•]=