Helix Fun

Paul Kotschy

4 May 2023

Compiled on April 2, 2025

THE IDEA FOR THIS SHORT ESSAY came from learning how eukaryote cells probably evolved from simpler prokaryotic cells. In the nucleus (or nuclei) of eukaryotic cells, each chromosome is comprised of a single DNA molecule. Because the molecule is so long, to fit inside the nucleus it usually undergoes four levels of compact helical winding.

So while minding that winding, let's have helix fun!

Eukaryotic cells. The cells of plants, animals, fungi and protozoa are known as eukaryotic cells. They are cells whose whose content is mostly nicely compartmentalised into separate organelles. Plastids, mitochondria, lysosomes, vacuoles, and the cell nucleus are some such organelles. The nucleus organelle contains the cell's chromosomes. Each chromosome is comprised of a single DNA molecule.

Levels of helical winding. DNA molecules can be thousands of times longer than a host cell's diameter. So to fit inside a cell's nucleus (or nuclei), the DNA molecule usually undergoes four levels of compact helical winding. The first level of winding is the familiar helical wind of the DNA molecule itself. In the second level, the DNA strand winds around certain proteins known as histone octamers, thereby forming a connected array of histone-DNA complexes known as nucleosomes. In the third level of winding, this array of nucleosomes winds around itself to form a tight helical fibre. Finally, the fourth level of winding of these fibres forms a so-called supercoil. Fibres of these supercoils are called chromatin, and it is this highly condensed chromatin which constitutes a chromosome. These levels of helical winding are shown schematically in Figure 1.

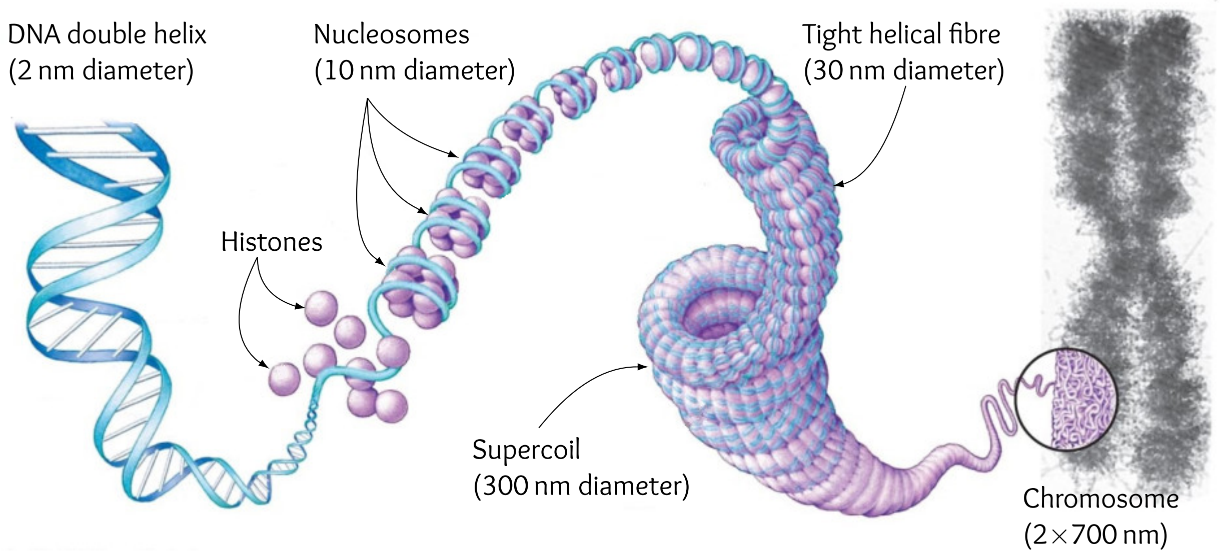

Figure 1: There is a lot of helical winding in a eukaryotic cell's chromosome. The DNA molecule forms a double-helix strand. The strand then winds around histone proteins to form an array of nucleosomes, which in turn wind around themselves forming a tight helical fibre. Finally, this fibre winds to form the supercoiled fibre called chromatin.

There is clearly a lot helical winding (and unwinding) occurring in a eukaryote's cell nucleus. So let's have some helix fun!

Some helix fun. Suppose we are interested in describing a solid helix which coils about an axis which is embedded in Euclidean space E^3, and which axis aligns with the $\hat{\mathbf{3}}$ basis vector. Suppose, too, that we would like to describe a particular path in E^3 which is bound to the two-dimensional surface of the solid helix. Such a helix is shown in Figure 2.[1]

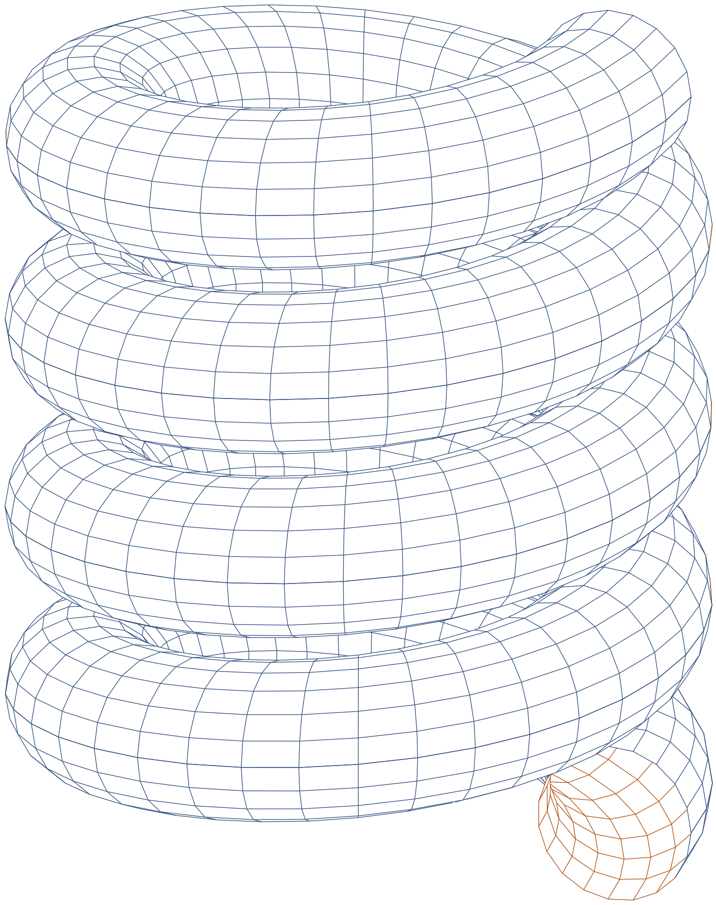

Figure 2: Surface of a solid helix which coils about an axis which aligns with the $\hat{\mathbf{3}}$ vector in Figure 3. The surface is defined by the set in (1) on page 3 with $r = 1.0$, $\rho = 0.3$, and $\delta = 0.2$.

A typical cross-sectional disc of the helix is shown in Figure 3. In the figure, the path in E^3 of the center of the solid helix is shown in brickred, and a typical point in E^3 along this path is denoted $\mathbf{c}(\theta)$. The cross-sectional circle of the solid helix surface which passes through $\mathbf{c}(\theta)$ is shown. It's radius is denoted ρ. As the central path coils about the axis with increasing θ, it prescribes a circle of radius r in the $\hat{\mathbf{1}}\hat{\mathbf{2}}$ plane.

A typical point on the edge of this cross-section, i.e., on the surface of the solid helix, is denoted $\mathbf{h}(\theta, \psi)$. The coordinates of $\mathbf{h}(\theta, \psi)$ depend on the angles θ and ψ:

$$\mathbf{h}(\theta, \psi) \;=\; h_1(\theta, \psi)\hat{\mathbf{1}} \;+\; h_2(\theta, \psi)\hat{\mathbf{2}} \;+\; h_3(\theta, \psi)\hat{\mathbf{3}}$$

where by inspection of the above figure

$$\begin{aligned} h_1(\theta, \psi) &= r\cos\theta + \rho\cos\psi\cos\theta = \cos\theta(r + \rho\cos\psi) \\ h_2(\theta, \psi) &= \sin\theta(r + \rho\cos\psi) \\ h_3(\theta, \psi) &= l(\theta) + \rho\sin\psi \end{aligned}$$

[1]This document was typeset using LaTeX. Figures 2 and 4 were drawn using GNUPLOT, the commands for which were embedded directly into the LaTeX source document using Paul Kotschy's PKSHELL. Figure 3 was drawn using the venerable Ti*k*Z graphics system for TeX.

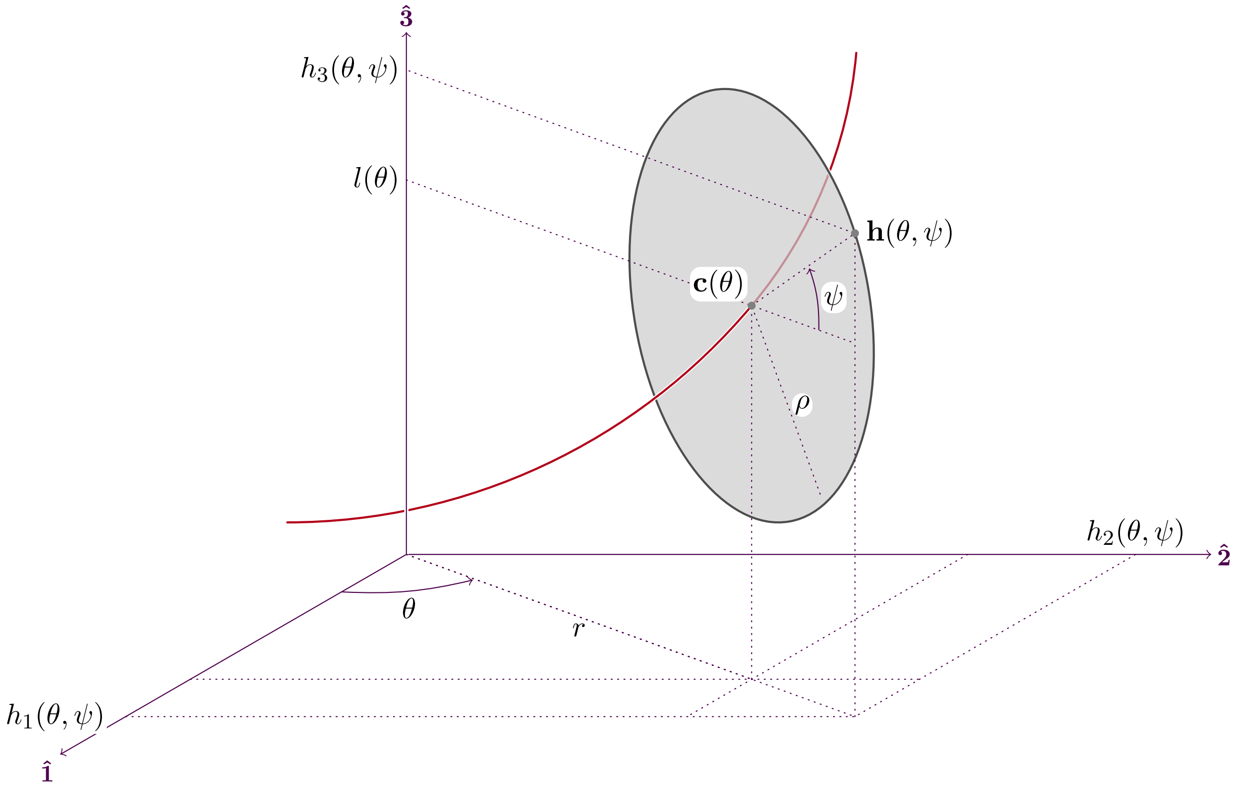

Figure 3: Typical cross-sectional disc of a solid helix located at a point $\mathbf{c}(\theta)$ along the central path in Euclidean space of the solid helix. A typical point on the edge of the disc, i.e., on the helix surface, is denoted $\mathbf{h}(\theta, \psi)$.

and where $l(\theta)$ is some as yet unspecified function which characterises the "intensity"of helical coiling. We are free to specify $l(\theta)$.

Suppose we require a compact solid helix which does not self-intersect with each helical coil instance. Then the specification of $l(\theta)$ will need to reflect this. The ψ angular parameter traces a point's position $\mathbf{h}(\theta, \psi)$ on the solid helix's surface relative to the central helix trajectory, which trajectory itself is traced by the angular parameter θ. The $\rho \sin \psi$ term in $h_3(\theta, \psi)$ has range $[-\rho, \rho]$. Therefore, for a compact non-self-intersecting solid helix we must demand that

$$l(\theta + 2\pi) - l(\theta) \geq 2\rho$$

A simple function satisfying this condition is

$$l(\theta) = \frac{(1+\delta)\rho}{\pi}\,\theta \qquad \text{with } \delta \geq 0$$

With this, the two-dimensional surface over the solid helix would be parametrised by the angular parameters θ and ψ as the set of all points

$$\begin{aligned}\{\mathbf{h}(\theta, \psi) \;&=\; \cos\theta(r + \rho\cos\psi)\,\hat{\mathbf{1}} \;+\; \sin\theta(r + \rho\cos\psi)\,\hat{\mathbf{2}} \;+\; \left(\frac{1+\delta}{\pi}\,\theta + \sin\psi\right)\rho\,\hat{\mathbf{3}} \\ &\mid\; \theta, \psi \in \mathbb{R},\; \delta \geq 0\}\end{aligned} \tag{1}$$

Suppose now that instead of the two-dimension surface of the solid helix, we are interested in a specific path across the surface. Choosing a path is equivalent to reducing the number of degrees of freedom in (1) from 2 to 1. And that means to specify a functional relationship between θ and ψ. For example, we could demand that θ traces out n helical coil revolutions for every m spins traced by ψ. That is, we could demand that

$$\theta(\psi) = (\frac{n}{m})\psi \qquad \text{for } m, n \in \mathbb{N}$$

Then clearly $\theta(2m\pi) = 2n\pi$. The one-dimensional path over the surface of the solid helix would then be parametrised by the angular parameter ψ as the set of all points

$$\begin{aligned}\{\mathbf{h}(\psi) \;&=\; \cos\theta(\psi)(r+\rho\cos\psi)\,\hat{\mathbf{1}} \;+\; \sin\theta(\psi)(r+\rho\cos\psi)\,\hat{\mathbf{2}} \;+\; \left(\frac{1+\delta}{\pi}\,\theta(\psi)+\sin\psi\right)\rho\,\hat{\mathbf{3}} \\ &\mid\; \theta(\psi)=\frac{n}{m}\psi,\quad m,n\in\mathbb{N},\quad \psi\in\mathbb{R}\}\end{aligned} \tag{2}$$

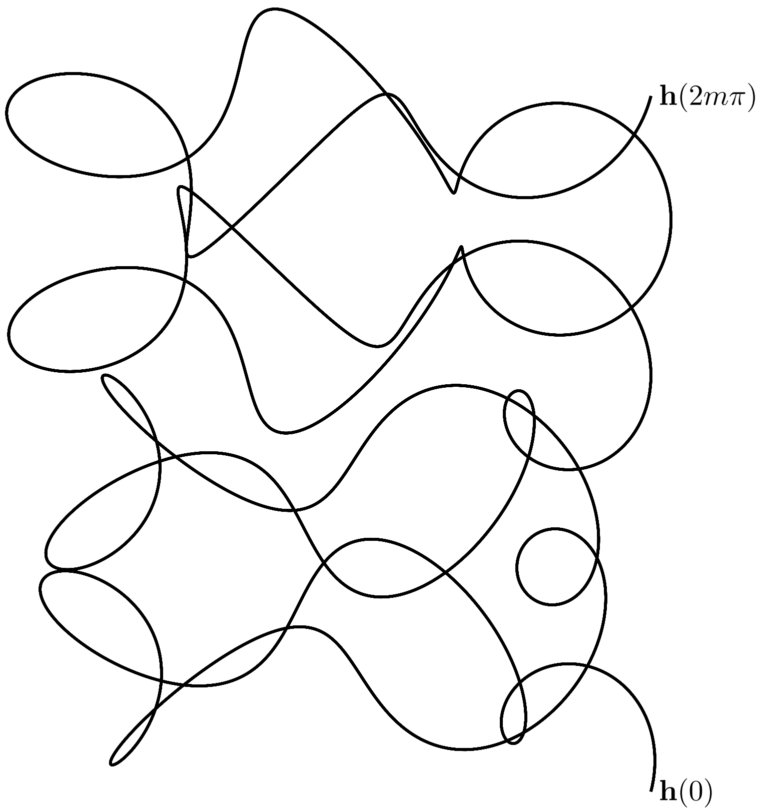

Figure 4: Path on the surface of a solid helix which coils about an axis which aligns with the $\hat{\mathbf{3}}$ vector in Figure 3. The path is defined by (2) with $r = 1.0$, $\rho = 0.3$, $\delta = 0.2$, $m = 17$ and $n = 4$.

Computed drawing with TEX, Ti*k*Z and Gnuplot. The following script for GNUPLOT's `gnuplot` command was used to compute Figure 2.

For details on GNUPLOT's pm3d, `fill`, `set style line`, and `set linetype`, refer to `/usr/local/share/examples/gnuplot/hidden_compare.dem`.

```
reset
#
#set terminal pdfcairo size 16cm,18cm
#set output "helix-surface.pdf"
#
set terminal cairolatex size 12cm,13cm
set output "helix-surface.tex"
#
unset border
#set border 0
```

```
unset tics
unset key
unset colorbox
set xyplane at 0
set view equal xyz
set view 70,20,1.5,
#
#set pm3d depthorder border linecolor rgb "0x707070" linewidth 0.5
#set pm3d nolighting
#set style fill transparent solid 0.5
##set hidden3d back
##set hidden3d back offset 1
set hidden
#
#set xlabel "x"
#set ylabel "y"
#set zlabel "z"
#set style function dots
#set autoscale z
##set isosamples 100
#set isosamples 100
set isosamples 150,20
#
set parametric
set dummy theta,psi
#
# Define my own line styles.
#
set style line 101 linecolor "black"
set style line 102 linecolor "cyan"
#
# Define my own line types.
#
#set linetype 1 linecolor rgb "black"
# This is close to ’pkdocblue’.
set linetype 1 linecolor rgb "0x305080"
#set linetype 2 linecolor rgb "0x707070"
# This is close to ’pkdocorange’.
set linetype 2 linecolor rgb "0xb05010"
# This is close to ’pkdocbrown’.
#set linetype 2 linecolor rgb "0x805030"
#
spins=1
revs=4
set urange [0:revs*2.0*pi]
set vrange [0:spins*2.0*pi]
#
radius=1.0
rho=0.3
delta=0.2
helixx(theta,psi) = cos(theta)*(radius + rho*cos(psi))
helixy(theta,psi) = sin(theta)*(radius + rho*cos(psi))
helixz(theta,psi) = ((1+delta)*theta/pi + sin(psi))*rho
#
splot helixx(theta,psi),helixy(theta,psi),helixz(theta,psi) with lines linetype 1
```

The following Ti*k*Z script was used to compute Figure 3.

```
\begingroup
\def\oneAngle{-150}
```

```
\def\oneLength{1}
\def\twoLength{1}
\def\threeLength{1}
\def\cxLength{0.2*\linewidth}
\def\cyLength{0.45*\linewidth}
\def\czLength{0.3*\linewidth}
\def\cxyAngle{-20}
\def\crossSectRadius{0.45*\czLength}
\def\helixRadius{3*\crossSectRadius}
\def\cTohAngle{35}
%
\def\setPoint{\pktikzSetUncircledPoint}
%\def\setPoint{\pktikzSetLabelledPoint}
%
\begin{PkTikzpicture}[helixpath/.style={helixpathcolor,
                                        thick,
                                        pktikzshadowed},
                      helixsection/.style={draw=crosssectioncolor,
                                           thick,
                                           fill=crosssectioncolor!20!white,
                                           %fill=crosssectionlightcolor,
                                           opacity=0.7,
                                           pktikzshadowed}]
   %\draw[help lines] (-0.2,-0.2) grid (\twoLength.1,\threeLength.1);
   %
   \setPoint{(0,0)}{origin};
   %
   % Preliminary basis positions 'one', 'two' and 'three'.
   %
   \setPoint{(\oneAngle:\oneLength)}{one};
   \setPoint{(right:\twoLength)}{two};
   \setPoint{(up:\threeLength)}{three};
   %
   % Begin with the centre 'c's 'x', 'y' and 'z' coordinates.
   %
   \setPoint{(\oneAngle:\cxLength)}{cx};
   \setPoint{(right:\cyLength)}{cy};
   \setPoint{(up:\czLength)}{cz};
   %
   % Point 'c(theta)' located at the solid helix surface centre.
   %
   \setPoint{(cx) +(right:\cyLength)}{cxy};
   \setPoint{(cz) +(\cxyAngle:1)}{u};
   \setPoint{(cxy) +(up:1)}{v};
   \setPoint{(intersection of cz--u and cxy--v)}{c};
   %
   % Back half of path of 'c(theta)'
   %
   \draw[helixpath]
      (c) arc[start angle=-40,end angle=-5,radius=0.9*\helixRadius];
   %
   % Helix surface cross-section drawn as a rotated ellipse.
   %
   \draw[helixsection,name path=crossSectPath]
      (c) circle[x radius=0.7*\crossSectRadius,y radius=1.3*\crossSectRadius,rotate=10];
   %
   % Point 'h(theta,psi)' located on the solid helix surface.
   %
   \path[name path=psiRay] (c) -- +(\cTohAngle:\crossSectRadius);
   \path[name intersections={of=crossSectPath and psiRay}]
      (intersection-1) coordinate(u);
   \setPoint{(u)}{h};
```

```
\setPoint{(h) +(down:1)}{hxy};
\setPoint{(intersection of origin--cxy and h--hxy)}{hxy};
\setPoint{(hxy) +(left:1)}{hx};
\setPoint{(intersection of origin--one and hxy--hx)}{hx};
\setPoint{(hxy) +(\oneAngle+180:1)}{hy};
\setPoint{(intersection of origin--two and hxy--hy)}{hy};
\setPoint{(h) +(\cxyAngle+180:1)}{hz};
\setPoint{(intersection of origin--three and h--hz)}{hz};
%
% Front half of path of 'c(theta)'
%
\draw[helixpath]
   (c) arc[start angle=-40,end angle=-90,radius=1.2*\helixRadius];
%
% Basis vectors '1', '2' and '3'.
%
\setPoint{(hx)   ([turn]0:1)}{one};
\setPoint{(hy)   ([turn]0:1)}{two};
\setPoint{(hz) ([turn]0:0.5)}{three};
\draw[pktikzbasisvector,<->]
   (one) node[below left]{$\one$}
   -- (origin) -- (two) node[right]{$\two$};
\draw[pktikzbasisvector]
   (origin) -- (three) node[above]{$\three$};
%
% Dimensions, angles and labels begin.
%
% Radius 'rho'.
%
\path[name path=rhoRay] (c) -- +(-70:1.4*\crossSectRadius);
\path[name intersections={of=crossSectPath and rhoRay}]
   (intersection-1) coordinate(rho);
%
\draw[pktikzdimension]
   (hx) node[pktikzlabel,left=1ex]{$h_1(\theta,\psi)$} -- (hxy)
        -- (hy) node[pktikzlabel,above]{$h_2(\theta,\psi)$}
   (cx) -- (intersection of cx--cxy and hy--hxy)
   (cy) -- (intersection of cy--cxy and hx--hxy)
   (origin) -- node[pktikzlabel,below]{$r$} (cxy)
   (origin) -- (hxy)
   (hxy) -- (h) -- (hz) node[pktikzlabel,left]{$h_3(\theta,\psi)$}
   (cxy) -- (c)
   (cz) node[pktikzlabel,left]{$l(\theta)$} -- (intersection of cz--c and hxy--h)
   (c) -- (h)
   (c) -- node[pktikzlabel,right]{$\rho$} (rho);
\pktikzDrawLabelledPoint[fill=black]%
                        {(c)}%
                        [above left]%
                        {$\pktikzVector{c}(\theta)$};
\pktikzDrawLabelledPoint[fill=black]%
                        {(h)}%
                        [right=0.3ex]%
                        {$\pktikzVector{h}(\theta,\psi)$};
%
% The 'psi' angle.
%
\path[name path=czcLine] (c) -- +(\cxyAngle:\crossSectRadius);
\path[name intersections={of=crossSectPath and czcLine}]
   (intersection-1) coordinate(u);
\draw[pktikzangle]
   ($ (c)!0.55!(u) $)
      to[bend right=10] node[pktikzlabel,right]{$\psi$} ($ (c)!0.55!(h) $);
```

```
   %
   % The 'theta' angle.
   %
   \draw[pktikzangle]
      ($ (origin)!0.3!(cx) $)
         to[bend right=8] node[pktikzlabel,below]{$\theta$} ($ (origin)!0.2!(cxy) $);
   %
   % Dimensions, angles and labels end.
   %
\end{PkTikzpicture}
\endgroup
\begin{PktdListing}
%
% 6May23---Below is NB. Place it in my 'tikz-pictures' package. It
% demonstrates the use of '[turn]' coordinate definitions, and of
% drawing curves without explicitly knowing target coordinates.
%
%\begin{PkTikzpicture}[scale=1.0]
%   \draw
%      (0,0) -- (0:4)
%            -- ([turn]90:2)
%            -- ([turn]90:4);
%   \draw[yshift=-3cm,pkdocred]
%      (0,0) -- (0:4) to[bend right=90,looseness=2]
%               ([turn]90:2)
%            -- ([turn]0:4);
%   \draw[yshift=-6cm,pkdocpurple,looseness=10]
%      (0,0) -- (0:4) to[out=90,out=-90,in=-90,relative]
%               ([turn]90:2)
%            -- ([turn]0:4);
%\end{PkTikzpicture}
%
%\begin{PkTikzpicture}[scale=1.0]
%   \draw
%      (0,0) -- (20:4)
%             -- ([turn]90:2)
%             -- ([turn]90:4);
%   \draw[yshift=-3cm,pkdocred]
%      (0,0) -- (20:4) to[bend right=90,looseness=1]
%               ([turn]90:2)
%            -- ([turn]0:4);
%   \draw[yshift=-6cm,pkdocpurple,looseness=10]
%      (0,0) -- (20:4) to[out=90,out=-90,in=-90,relative]
%               ([turn]90:2)
%            -- ([turn]0:4);
%\end{PkTikzpicture}
```

The following Gnuplot script was used to compute Figure 4.

```
reset
#
#set terminal pdfcairo linewidth 2 size 16cm,18cm
#set output "helix-surface-path.pdf"
#
set terminal cairolatex linewidth 2 size 12cm,12cm
set output "helix-surface-path.tex"
#
unset border
##set border 0
unset tics
```

```
unset key
unset colorbox
set xyplane at 0
set view equal xyz
set view 70,20,1.5,
#
#set xlabel "x"
#set ylabel "y"
#set zlabel "z"
#set style function dots
#set autoscale z
unset pm3d
unset hidden
unset isosamples
set samples 2000
#
set parametric
set dummy theta,psi
#
# Define my own line type.
#
set linetype 1 linecolor rgb "black"
#
spins=17
revs=4
set vrange [0:spins*2*pi]
#
radius=1.0
rho=0.3
delta=0.2
helixx(theta,psi) = cos(theta)*(radius + rho*cos(psi))
helixy(theta,psi) = sin(theta)*(radius + rho*cos(psi))
helixz(theta,psi) = ((1+delta)*theta/pi + sin(psi))*rho
#
myTheta(psi) = revs/(spins*1.0)*psi
psiMax = spins*2*pi
set label ’$\pktikzVector {h}(0)$’ at helixx(myTheta(0),0),\
                                              helixy(myTheta(0),0),\
                                              helixz(myTheta(0),0) offset 0.7
set label ’$\pktikzVector {h}(2m\pi )$’ at helixx(myTheta(psiMax),psiMax),\
                                                           helixy(myTheta(psiMax),psiMax),\
                                                           helixz(myTheta(psiMax),psiMax) offse
#splot helixx(myTheta(psi),psi),\
       helixy(myTheta(psi),psi),\
       helixz(myTheta(psi),psi) title ’$h$’ linecolor "orange"
splot helixx(myTheta(psi),psi),\
      helixy(myTheta(psi),psi),\
      helixz(myTheta(psi),psi) title ’$h$’ linetype 1 linewidth 2
```