

17-30 January 1995 CHR, continuing from "Oseen", 2 November 1994

Use Zimm (1980) formulation to solve hydrodynamics of a given subunit array. Compare various aspects of the Kirkwood/Bloomfield et al. (which I will sometimes call "K/B") approximation.

CHR started 18 Feb 2020. Test of reproducibility for the ReScience 10-year reproducibility challenge (<https://rescience.github.io/ten-years/>). Chosen paper is Robert, C.H. (1995) Estimating Friction Coefficients of Mixed Globular/Chain Molecules, such as Protein/DNA Complexes. Biophys J. 69, 840-48.

All code in this file came from the original notebook "s6.0 Zimm (1980).nb" and was run using Mathematica versions 5.2 and 12.0.

Changes:

- added dependence on library s6.1.1 (Get[...])
- added comments
- copied in MatrixForm tables of original results for direct comparison
- added new final plot that matches Figure 3 in the published article for easier comparison

Blue and gray boxes indicate results to compare to originals as well as certain comments.

```
Get["ReScience/" <> "s6.1_repro.m"];
```

■ General functions to define interaction tensor, velocity perturbations, etc.

```
Off[General::spell];
Off[General::spell1];

avg[x_] := Apply[Plus, x] / Length[x];

l[r_] := Sqrt[r.r];
rr[r_] := Outer[Times, r, r];
Id[n_] := IdentityMatrix[n];
zero[n_] := Table[0, {n}, {n}];

T[r_] := 1 / (8 Pi nu l[r]) (Id[Length[r]] + rr[r] / l[r]^2);

MatrixForm[T[{x, y, z}]]
```

$$\begin{pmatrix} \frac{1 + \frac{x^2}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi \sqrt{x^2 + y^2 + z^2}} & \frac{\frac{x y}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} & \frac{\frac{x z}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} \\ \frac{\frac{x y}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} & \frac{1 + \frac{y^2}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi \sqrt{x^2 + y^2 + z^2}} & \frac{\frac{y z}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} \\ \frac{\frac{x z}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} & \frac{\frac{y z}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi (x^2 + y^2 + z^2)^{3/2}} & \frac{1 + \frac{z^2}{x^2 + y^2 + z^2}}{8 \text{ nu } \pi \sqrt{x^2 + y^2 + z^2}} \end{pmatrix}$$

Velocity and velocity perturbation functions.

```
V[i_, {p_, radius_}] := Id[3] / (6 Pi nu radius[[i]])
dV[i_, j_, {p_, radius_}] := T[p[[j]]] - p[[i]];
u[i_, {p_, radius_}] := {-p[[i, 2]], p[[i, 1]], 0};
z[i_, {p_, radius_}] := {p[[i, 2]], -p[[i, 1]], 0};
```

Demonstrate some velocity profiles

The perturbation of the velocity of the fluid at a given point $r\{x,y,z\}$ with respect to a frictional sphere is given by the following expression for dv for a force $f\{x,y,z\}$ acting on the center.

The viscosity nu has typically the units of poise, or $\text{g}/(\text{cm sec})$. If distances are in cm and force is $(\text{g cm}/\text{sec}^2)$, the

velocity given here is cm/sec.

The relative velocity $dv0$ is the relative solvent velocity perturbation about a particle travelling with whatever velocity results from a unit force acting on it, expressed in terms of the distance relative to the particle radius. The factor of the particle radius cancels. A positive value says that the sphere is dragging fluid with it.

```
ClearAll[dv,dv0];
dv[r_,f_]:=T[r].f;
dv0[r0_,f0_]:=dv[r0,f0] 6 Pi nu;

dv[{1,0,0} cm,{1,0,0} g cm/s^2]/.nu->g/(cm s)

{ $\frac{\sqrt{cm^2}}{4 \pi s}$ , 0, 0}

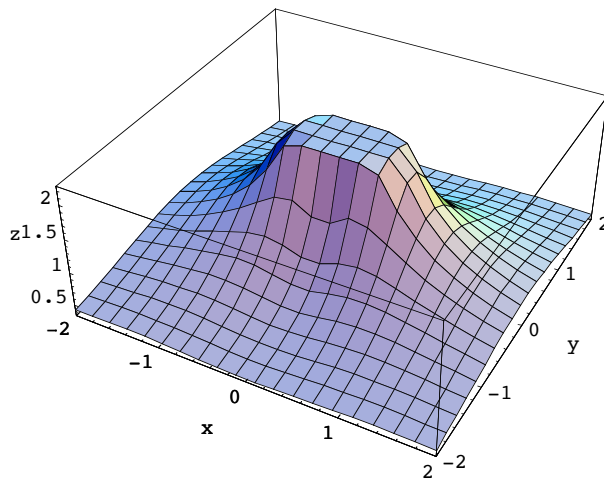
dv0[{1,0,0},{1,0,0}]

{ $\frac{3}{2}$ , 0, 0}

dv0[{1.1,0,0},{fx,fy,fz}]

{1.36364 fx, 0.681818 fy, 0.681818 fz}

Plot3D[r={x,y,0};1[N[dv0[r,{1,0,0}]]],
{x,-2,2},{y,-2,2},PlotPoints->20,
AxesLabel->{"x","y","z"}]
```



- SurfaceGraphics -

This 3D plot was not displayed in Mathematica 12.0. (It was calculated here as an exploration, and is not part of the original article).

■ **Define matrix flatteners to set up the linear hydrodynamics equations:**

```
flatmat[M_]:= Module[{Mt,Mf},
  Mt=Transpose[M,{1,3,2,4}];
  Mf=Table[Flatten[Mt[[i,j]]],{i,Length[Mt]},{j,Length[Mt[[1]]]}];
  N[Flatten[Mf,1]]
];

flatvec[x_]:=Flatten[x,1];
```

```

complete[M_,geo_]:=Module[{base,n,npoints,Mc},
  (* Ugly but necessary, since the additional equations required
    to solve the tableau destroy the "tensorness" of the matrix
    of matrices. *)
  dim=Length[M];
  n=dim/3-1;
  Mc=Table[0,{dim+1},{dim+1}];
  Do[ Mc[[i,j]]=M[[i,j]],{i,dim},{j,dim}];
  Do[ base=(i-1)*3;
      Mc[[base+1,dim+1]]=Evaluate[u[i,geo]][[1]];
      Mc[[base+2,dim+1]]=Evaluate[u[i,geo]][[2]];
      Mc[[base+3,dim+1]]=Evaluate[u[i,geo]][[3]];
      {i,n}];
  bottom=Flatten[Table[z[i,geo],{i,n}]];
  Do[ Mc[[dim+1,j]]=bottom[[j]],{j,3*n}];
  Return[Mc]
];

```

Show tensor->matrix flattening in action:

```

M={ {{a1111,a1112},{a1121,a1122}},{a1211,a1212},{a1221,a1222}},
    {{a2111,a2112},{a2121,a2122}},{a2211,a2212},{a2221,a2222}} };

```

MatrixForm[M]

$$\begin{pmatrix} \begin{pmatrix} a_{1111} & a_{1112} \\ a_{1121} & a_{1122} \end{pmatrix} & \begin{pmatrix} a_{1211} & a_{1212} \\ a_{1221} & a_{1222} \end{pmatrix} \\ \begin{pmatrix} a_{2111} & a_{2112} \\ a_{2121} & a_{2122} \end{pmatrix} & \begin{pmatrix} a_{2211} & a_{2212} \\ a_{2221} & a_{2222} \end{pmatrix} \end{pmatrix}$$

MatrixForm[Transpose[M,{1,3,2,4}]]

$$\begin{pmatrix} \begin{pmatrix} a_{1111} & a_{1112} \\ a_{1211} & a_{1212} \end{pmatrix} & \begin{pmatrix} a_{1121} & a_{1122} \\ a_{1221} & a_{1222} \end{pmatrix} \\ \begin{pmatrix} a_{2111} & a_{2112} \\ a_{2211} & a_{2212} \end{pmatrix} & \begin{pmatrix} a_{2121} & a_{2122} \\ a_{2221} & a_{2222} \end{pmatrix} \end{pmatrix}$$

MatrixForm[flatmat[M]]

$$\begin{pmatrix} a_{1111} & a_{1112} & a_{1211} & a_{1212} \\ a_{1121} & a_{1122} & a_{1221} & a_{1222} \\ a_{2111} & a_{2112} & a_{2211} & a_{2212} \\ a_{2121} & a_{2122} & a_{2221} & a_{2222} \end{pmatrix}$$

■ Random-orientation matrix definition

These are from my Nov'94 hydrodynamics program for the random orientation of a chain

```

Ax[w_]:={{1,0,0},
          {0,Cos[w],Sin[w]},
          {0,-Sin[w],Cos[w]}};
Ay[w_]:={{Cos[w],0,-Sin[w]},
          {0,1,0},
          {Sin[w],0,Cos[w]}};
Az[w_]:={{Cos[w],Sin[w],0},
          {-Sin[w],Cos[w],0},
          {0,0,1}};

AO:=Az[Random[Real,N[{0,2 Pi}]]].Ax[Random[Real,N[{0,2 Pi}]]].
    Az[Random[Real,N[{0,2 Pi}]]];
(* Random overall orientation matrix *)

rotate[A_,o_]:=Module[{i,j,depth},
  (* Routine to rotate vectors even when they are assembled

```

```

        into groups and lists of those groups. Rotation matrix
        is A, vector object is o *)
    depth=Depth[N[o]];
    Which[ depth==1,Print["Not done: ",depth];Return[o],
           depth==2,Return[N[A].o],
           depth==3,Return[Transpose[N[A].Transpose[o]]],
           depth==4,Return[Map[
               Transpose[N[A].Transpose[#]] &,o ]
           ],
           depth>=5,Print["Not done: ",depth];Return[o]
    ]
];

```

■ Higher level (automating) routines

```

element[i_,j_,geo_]:=If[And[i<n+1,j<n+1],
                        If[i==j,V[i,geo],dV[i,j,geo]],
                        If[i<j,-Id[3],If[i==j,zero[3],Id[3]]]
];

ClearAll[hydro];
hydro[geo_]:=Module[{},
  (* Set up and solve tableau AC.xc = yc for xc for a given
     rigid set of particles at positions p with radii s and
     for a given viscosity nu (here nu->1). 'geo' is a
     structure containing a list of coordinates and an
     associated list of hydrodynamic radii *)
  n=Length[geo[[1]]];
  If[ test,
    Print["Hydrodynamics of an assembly of ",n," spheres"];
    Print[" {x,y,z} radius"];
    Print[MatrixForm[Transpose[geo]]]
  ];
  A=Table[element[i,j,geo]/.nu->1,{i,n+1},{j,n+1}];
  AA=flatmat[A];
  AC=complete[AA,geo];
  xx=Join[Table[Map[#<>ToString[i] &,{ "fx", "fy", "fz"}],{i,n}],
    {{ "ux", "uy", "uz", "omega z"}]];
  xc=flatvec[xx];
  yy=Join[Table[{0,0,0},{n}],{0,0,1,0}];
  yc=flatvec[yy];
  If[test,Print["Matrix is size: ",Dimensions[AC]]];
  G=Inverse[AC];
  data=Append[data,G.yc];
  If[test,Return[MatrixForm[Transpose[{xc,G.yc}]]]]
];

```

■ Show relationships between equivalent sphere radius and distance in "lollipop" geometry

Copy over the constants used for the equivalent sphere calculations.

```

el0=3.4; (* bp length, A *)
persistence=150 el0; (* persistence length, A *)
repeat0=10.4; (* helical repeat of free chain bp/turn *)

NAvo=6.02 10^23;
ro=1.003; (* density, g/cm^3 *)
visc=0.01016; (* viscosity, g/(cm s) or "Poise" *)

```

```

mbpa=660;          (* basepair mol. weight, g/mole *)
rbpa=1.8294;       (* basepair Stokes radius, A *)
vDNA=0.55;         (* specific volume of DNA, cm^3/g *)
dDNA=27;           (* Yamagawa-Fujii (YF) diam, A *)
switchbp=50;       (* point at which we switch from YF
                    to ellipsoid model, bp *)

mbpa=660;          (* basepair mol. weight, g/mole *)
rbpa=1.8294;       (* basepair Stokes radius, A *)
vDNA=0.55;         (* specific volume of DNA, cm^3/g *)
dDNA=27;           (* Yamagawa-Fujii (YF) diam, A *)
switchbp=50;       (* point at which we switch from YF
                    to ellipsoid model, bp *)

```

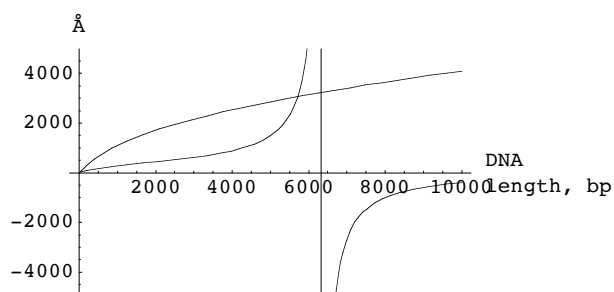
findoffset

Stokes offset 3.00764 A at 50 bp

```

Plot[{rootr2[i/2],rbpb[i]},{i,0.1,10000},
     PlotRange->{-5000,5000},
     AxesLabel->{"DNA\nlength, bp","Å"}}

```

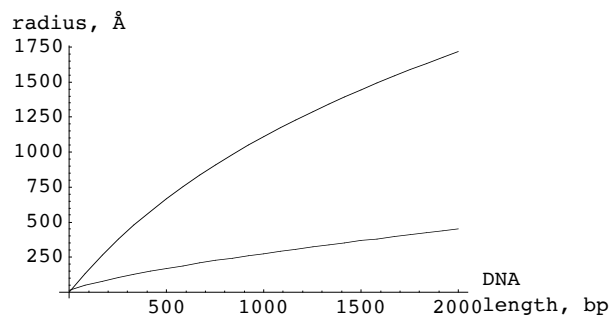


- Graphics -

```

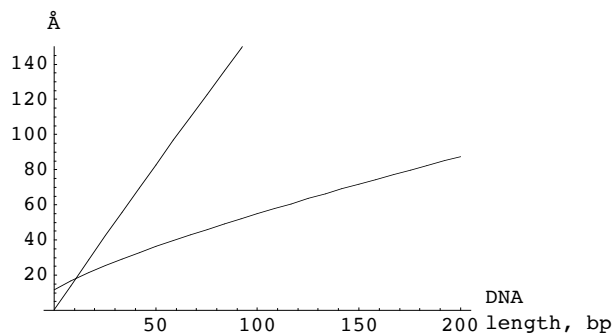
Plot[{rootr2[i/2],rbpb[i]},{i,0.1,2000},
     AxesLabel->{"DNA\nlength, bp","radius, Å"}}

```



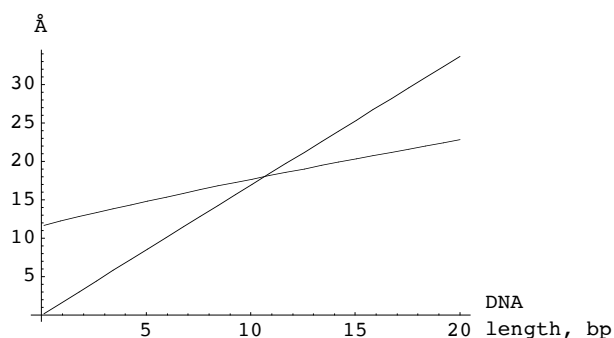
- Graphics -

```
Plot[{rootr2[i/2],rbpb[i]},{i,0.1,200},
      PlotRange->{0,150},
      AxesLabel->{"DNA\nlength, bp","Å"}]
```



- Graphics -

```
Plot[{rootr2[i/2],rbpb[i]},{i,0.1,20},
      AxesLabel->{"DNA\nlength, bp","Å"}]
```



- Graphics -

■ Simulate lollipops to see applicability of Kirkwood/BDvH approximation

I make a two-particle system here that is again like one in my paper. The separation is taken as the center of the array of linker spheres (smaller spheres) from the previous example. First orient perpendicular to sedimentation direction, then nearly parallel (I used a 5 degree angle with the z axis. When the linear assembly is exactly parallel, the set of equations becomes singular).

```
test=True;
data={};
nall={};
radiall={};
uall={};
fzall={};
fBzall={};
AFzall={};
AFBzall={};
```

```

Do[ nbp=N[Exp[i/2.2]];
    points={{0,0,0},{57+rootr2[nbp/2],0,0}};
    radii={57,rbpb[nbp]};

    hydro[{points,radii}];
    ftrue=G.yc;

    ftruez=ftrue[[Range[3,3 n,3]]];
    uttruez=ftrue[[3 n+3]];
    AAz=AA[[Range[3,3 n,3],Range[3,3 n,3]]];
    AFz=N[-6 Pi radii AAz.DiagonalMatrix[ftruez]];
    Do[AFz[[i,i]]=N[6 Pi radii[[i]] uttruez],{i,Length[AFz]}];

    fbloomz=N[Map[#/Apply[Plus,radii] &,radii]];
    AAz=AA[[Range[3,3 n,3],Range[3,3 n,3]]];
    AFBz=N[-6 Pi radii AAz.DiagonalMatrix[fbloomz]];
    Do[AFBz[[i,i]]=N[6 Pi radii[[i]] uttruez],{i,Length[AFBz]}];

    AppendTo[nall,nbp];
    AppendTo[radiiall,radii];
    AppendTo[uall,uttruez];
    AppendTo[fzall,N[Apply[Plus,Apply[Plus,Transpose[AFz]]]]];
    AppendTo[fBzall,N[Apply[Plus,Apply[Plus,Transpose[AFBz]]]]];
    AppendTo[AFzall,AFz];
    AppendTo[AFBzall,AFBz],
{i,-2,9}];

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius

( {0, 0, 0}          57
  {57.6848, 0, 0}   11.8775 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius

( {0, 0, 0}          57
  {58.0787, 0, 0}   12.0343 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius

( {0, 0, 0}          57
  {58.6991, 0, 0}   12.2788 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius

( {0, 0, 0}          57
  {59.6759, 0, 0}   12.6582 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius

( {0, 0, 0}          57
  {61.2137, 0, 0}   13.2427 )

```

```

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {63.6332, 0, 0}   14.1351 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {67.4374, 0, 0}   15.4827 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {73.4114, 0, 0}   17.4935 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {82.776, 0, 0}    20.46 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {97.413, 0, 0}    24.7961 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {120.189, 0, 0}   31.0949 )

Matrix is size: {10, 10}

Hydrodynamics of an assembly of 2 spheres

{x,y,z}    radius
( {0, 0, 0}          57
  {155.383, 0, 0}   40.0377 )

Matrix is size: {10, 10}

```

Results for orientation 1 (aligned in x direction)

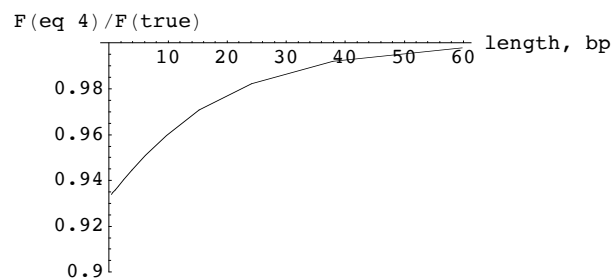

```
MatrixForm[N[Transpose[{nall, radiiall, uall, fzall, fBzall}], 4]]
```

```
{ 0.40289 {57.00, 11.8775} 0.000916278 1. 0.934018 }
{ 0.634736 {57.00, 12.0343} 0.000915527 1. 0.934715 }
{ 1. {57.00, 12.2788} 0.000914314 1. 0.935819 }
{ 1.57546 {57.00, 12.6582} 0.000912335 1. 0.937565 }
{ 2.48207 {57.00, 13.2427} 0.00090906 1. 0.94031 }
{ 3.91039 {57.00, 14.1351} 0.000903556 1. 0.944556 }
{ 6.16065 {57.00, 15.4827} 0.000894192 1. 0.950887 }
{ 9.70584 {57.00, 17.4935} 0.000878249 1. 0.959709 }
{ 15.2911 {57.00, 20.46} 0.000851683 1. 0.970702 }
{ 24.0905 {57.00, 24.7961} 0.000809678 1. 0.982305 }
{ 37.9536 {57.00, 31.0949} 0.000748624 1. 0.992031 }
{ 59.7942 {57.00, 40.0377} 0.000669685 1. 0.997898 }
```

The values in this table match exactly those in the original Mathematica notebook (but with a different number of significant figures provided), copied below:

0.4029	{57., 11.88}	0.0009163	1.	0.934
0.6347	{57., 12.03}	0.0009155	1.	0.9347
1.	{57., 12.28}	0.0009143	1.	0.9358
1.575	{57., 12.66}	0.0009123	1.	0.9376
2.482	{57., 13.24}	0.0009091	1.	0.9403
3.91	{57., 14.14}	0.0009036	1.	0.9446
6.161	{57., 15.48}	0.0008942	1.	0.9509
9.706	{57., 17.49}	0.0008782	1.	0.9597
15.29	{57., 20.46}	0.0008517	1.	0.9707
24.09	{57., 24.8}	0.0008097	1.	0.9823
37.95	{57., 31.09}	0.0007486	1.	0.992
59.79	{57., 40.04}	0.0006697	1.	0.9979

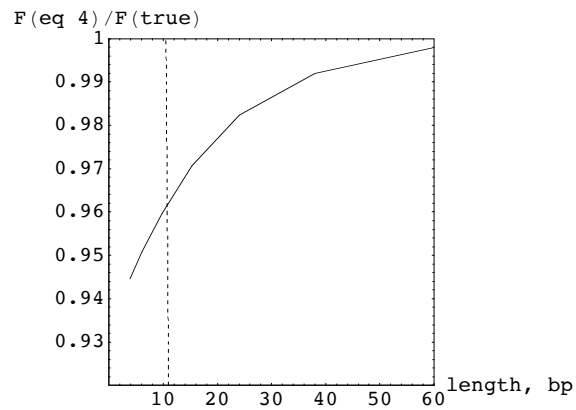
```
ListPlot[Transpose[{nall, fBzall}],
  PlotJoined->True,
  AxesLabel->{"length, bp", "F(eq 4)/F(true)"},
  PlotRange->{0.9, 1}]
```



- Graphics -

In the article, the data were plotted in Figure 3 using CricketGraph, but the relevant files are no longer readable. Below I mocked up the essential parts of this figure using the recalculated data. (Note that only the last 7 data points were used in making the published plot.)

```
Show[
  Graphics[{Dashing[{0.01,0.015}],Line[{{11,0.92},{10.5,1}}]}],
  ListPlot[Transpose[{Take[nall,-7],Take[fBzall,-7]}],
    PlotJoined->True,
    DisplayFunction->Identity
  ],
  AxesLabel->{"length, bp","F(eq 4)/F(true)"},
  PlotRange->{{0,60},{0.92,1}},AspectRatio->8.0/7.5,Axes->True, Frame->True,
  DisplayFunction->$DisplayFunction
]
```



- Graphics -