This mathematica code solves the differential equation for the height of the interface between the almost colloid free liquid phase and the gel of particles.
The differential equation can be found in the paper by Alexis Darras et al. named: “Erythrocyte sedimentation: Fracture and collapse of a high-volume-fraction soft-colloid gel”.
A scaled time and height are used. The results from the two-dimensional model are included for comparison.

```
In[∘]:= (*Results from 2D model*)
height = Import["C:\\Users\\myrth\\Downloads\\Data.xlsx"];
(*change directory of excel file*)
times = Import["C:\\Users\\myrth\\Downloads\\time_Data.xlsx"];
time = {};
For[i = 1, i < Length[height[[1]][[1]]] + 1,
  i++, AppendTo[time, times[[1]][[i]][[1]]]];
```

```
In[∘]:= (*Plot 2D result*)
p2 = ListPlot[Thread[{time, height[[1]][[1]]}], AxesLabel → {t, h[t]},
  PlotLegends → {"h(t), 2D", hm}, LabelStyle → Directive[FontSize → 14],
  PlotRange → {0, 1}, PlotStyle → Purple, ImageSize → Large]
```

```
In[∘]:= (*1D model*)
Clear["Global`*"]
(*Initial conditions:*)
phi0 = 4/10; (*initial colloid volume fraction*)
h0 = 4; (*[cm] initial height of the interface*)
drho = 80 * 10^(-6);
(*[kg/cm^3] density difference between the colloids and the suspending medium*)
g = 981; (*[cm/s^2] gravitational acceleration*)
mu = 0.000012; (*[kg/(cm s)]suspending medium viscosity*)
phim = 86/100; (*maximum colloid volume fraction*)
hm = h0 * phi0/phim ;(*minimum height of the interface*)
r = 3.3 * 10^(-4); (*cm, radius of RBC*)
tau = 9 * mu * h0/(2 * r^2 * drho * g);(*sedimentation time of a single RBC*)
(*Solve the differential equation:*)
hsol = NDSolveValue[
  {h'[t] == -(drho * g) * ((phim - phi0 * h0/(h[t]))^3) * (phi0 * h0/(h[t]))/mu,
   h[0] == h0}, h, {t, 0, 10^12}]
```

```
In[∘]:= (*Results from the 1D model*)
tfinal = Last[time];
pa = Plot[{hsol[t * tau]/h0, hm/h0}, {t, 0, tfinal},
  PlotRange → {0, 1}, AxesLabel → {t, h[t]}, PlotPoints → 1000,
  PlotLegends → {"h(t)", FractionBox[Subscript[h, m], Subscript[h, 0]] // DisplayForm},
  LabelStyle → Directive[FontSize → 14], ImageSize → Large]
```

```
In[∘]:= (*Export picture:*)
Export[NotebookDirectory[] <> "1D_h(t)_plot_scaled.png", pa, ImageResolution → 1000]
```

```
In[∘]:= (*Log plot 1D model*)
tfinal = 2 * 10^(-4);
pb = LogLinearPlot[{hsol[t * tau]/h0, hm/h0}, {t, 0.1 * 10^(-9), tfinal},
  PlotRange → {0, 1}, AxesLabel → {t, h[t]}, PlotPoints → 1000,
  PlotLegends → {"h(t)", FractionBox[Subscript[h, m], Subscript[h, 0]] // DisplayForm},
  LabelStyle → Directive[FontSize → 14], ImageSize → Large]
```

In[ ]:=
```
(*Export picture:*)
Export[NotebookDirectory[] <> "1D_h(t)_Logplot_scaled.png", pb, ImageResolution → 1000]

(*Plot 1D to combine:*)
tfinal = Last[time];
p1 = Plot[{hsol[t * tau] / h0, hm / h0}, {t, 0, tfinal},
  PlotRange → {0, 1}, AxesLabel → {t, h[t]}, PlotPoints → 1000, PlotLegends →
   {"h(t), 1D", FractionBox[Subscript[h, m], Subscript[h, 0]] // DisplayForm},
  LabelStyle → Directive[FontSize → 14], ImageSize → Large]

(*1D and 2D combined plot*)
combine = Show[p1, p2]
```

In[ ]:=
```
(*Export picture:*)
Export[NotebookDirectory[] <> "1D_2D_h(t)_scaled.png", combine, ImageResolution → 1000]
```