```
(*Mathematica does not have a PolarStreamPlot
 available.  In order to produce the streamline
 figurs in the paper in this post PolarStreamPlot
 needed to be created.

  The basic form of the Mathematica program used
 for the stream plots comes from Mathematica
 StackExchange and was posted by “s.pledger” on
 10 February 2014. The program is:*)

SetAttributes[PolarStreamPlot, HoldAll];
PolarStreamPlot[fns_, {r_Symbol, rMin_, rMax_},
  {t_Symbol, tMin_, tMax_}, opts___] :=
 Module[{Fns, x, y, RF, TMin, TMax, mArcTan, RMax},
  If[rMin ≥ rMax, Throw["Invalid range for r!"]];
  If[tMin ≥ tMax || tMax - tMin > 2 Pi,
   Throw["Invalid range for θ!"]];
  TMin = Mod[tMin, 2 Pi, tMin];
  TMax = Mod[tMax, 2 Pi, tMin];
  If[TMax == TMin, TMax += 2 Pi];
  mArcTan[vars__] = Mod[ArcTan[vars], 2 Pi, TMin];
  Fns = TransformedField["Polar" → "Cartesian", fns,
     {r, t} → {x, y}] /. ArcTan → mArcTan;
  RMax = rMax;
  StreamPlot[Fns, {x, -RMax, RMax}, {y, -RMax, RMax},
   RegionFunction →
    Function[{x, y},
     Evaluate[mArcTan[x, y] ≤ TMax &&
       rMin ≤ Sqrt[x^2 + y^2] ≤ rMax]], opts]]



(*Here is an example of how to use the program:*)


n = 3;
v0 = -0.2;
λ = 0.5;
b = a = 1

dblsol =
  {1/r
    D[((BesselJ[n, r] Sin[n θ]) + (λ) (1 - BesselJ[0, r]) +
       ((2 v0 a)/(b (BesselJ[0, b]))) (BesselJ[1, b r / a] Sin[θ])),
     θ],
   -D[((BesselJ[n, r] Sin[n θ]) + (λ) (1 - BesselJ[0, r]) +
       ((2 v0 a)/(b (BesselJ[0, b]))) (BesselJ[1, b r / a] Sin[θ])),
     r]};

PolarStreamPlot[dblsol, {r, 0, 10.0}, {θ, 0, 2 Pi},
 StreamPoints → Fine, StreamStyle → Black]
```