In[60]:=
```
T[string_] := Style[string, FontFamily -> "TeX Gyre Pagella", FontSize → 20]
customStyle = {
   PlotStyle → Thickness[0.006],
   Frame → True,
   FrameStyle → Black,
   FrameTicksStyle → {Directive[18, FontFamily -> "TeX Gyre Pagella"],
     Directive[18, FontFamily -> "TeX Gyre Pagella"]},
   ImageSize → {550, 350}
  };
```

In[62]:=
```
(*Dynamo effect for Kerr Black Hole*)
V0[r_] := - M / r

a := J / M (* it must always hold: a < M, J < M^2*)
Σ[r_] := r ^ 2 + a ^ 2 Cos[θ] ^ 2
Δ[r_] := r ^ 2 - r 2 M + a ^ 2
dt[r_] := 1 / (1 - 2 M r / Σ[r]) (*inverse of gtt*)
j [r_] := ((r ^ 2 + a ^ 2) ^ 2 - Δ[r] a ^ 2 Sin[θ] ^ 2 ) ω Sin[θ] ^ 2 / Σ[r] - (dt[r]) a 2 M / Σ[r] r Sin[θ] ^ 2
ϵ[r_] := a 2 M r ω Sin[θ] ^ 2 / Σ[r] + (1 - 2 M r / Σ[r]) (dt[r])

Vefftot[r_] := V0[r] +
   j[r] ^ 2 / 2 / r ^ 2 +
   1 / 2 (1 - ϵ[r] ^ 2) (1 + a ^ 2 / r ^ 2) -
  2 M / 2 / r ^ 3 (j[r] - a ϵ[r])(* only gtϕ element *)

Veff[r_, ω_, θ_] := V0[r] - M / r ω Sin[θ] ^ 2

(*ω[r_]:=Sqrt[M/r^ 3]*)
v[r_] := Sqrt[M / r]
rh[r_] := M + M Sqrt[1 - a ^ 2 Cos[θ] ^ 2]

Expand[Vefftot[r] /. J :→ 0, {r, 0, 1}]

subs := {M :→ 0.5, J :→ 0.05}
```

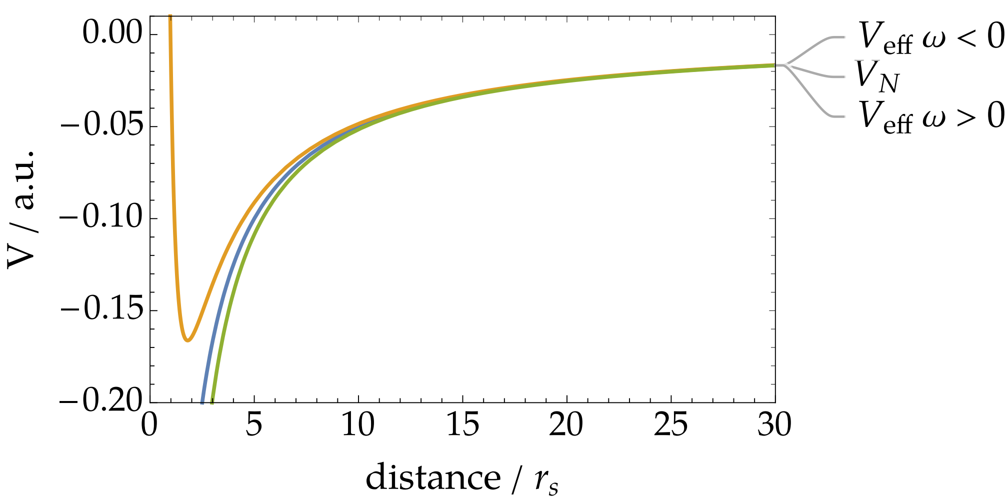
Out[73]= $-\frac{M}{r}-\frac{M\,\omega\,\mathrm{Sin}[\theta]^2}{r}+\frac{1}{2}\,r^2\,\omega^2\,\mathrm{Sin}[\theta]^4$

In[75]:=

In[102]:=

```
Plot[
 {V0[r] /. subs,
  Veff[r, -r^(-3/2), π/2 + 10 Degree] /. subs,
  Veff[r, +r^(-3/2), π/2 + 10 Degree] /. subs},
 {r, 0.5, 30},
 Evaluate@customStyle,
 FrameLabel → {T["distance / r_s"], T["V / a.u."]},
 PlotLabels → {T["V_N"], T["V_eff ω < 0"], T["V_eff ω > 0"]},
 PlotRange → {{0, 30}, {-0.2, 0.01}}
]
```

Out[102]=



In[77]:=

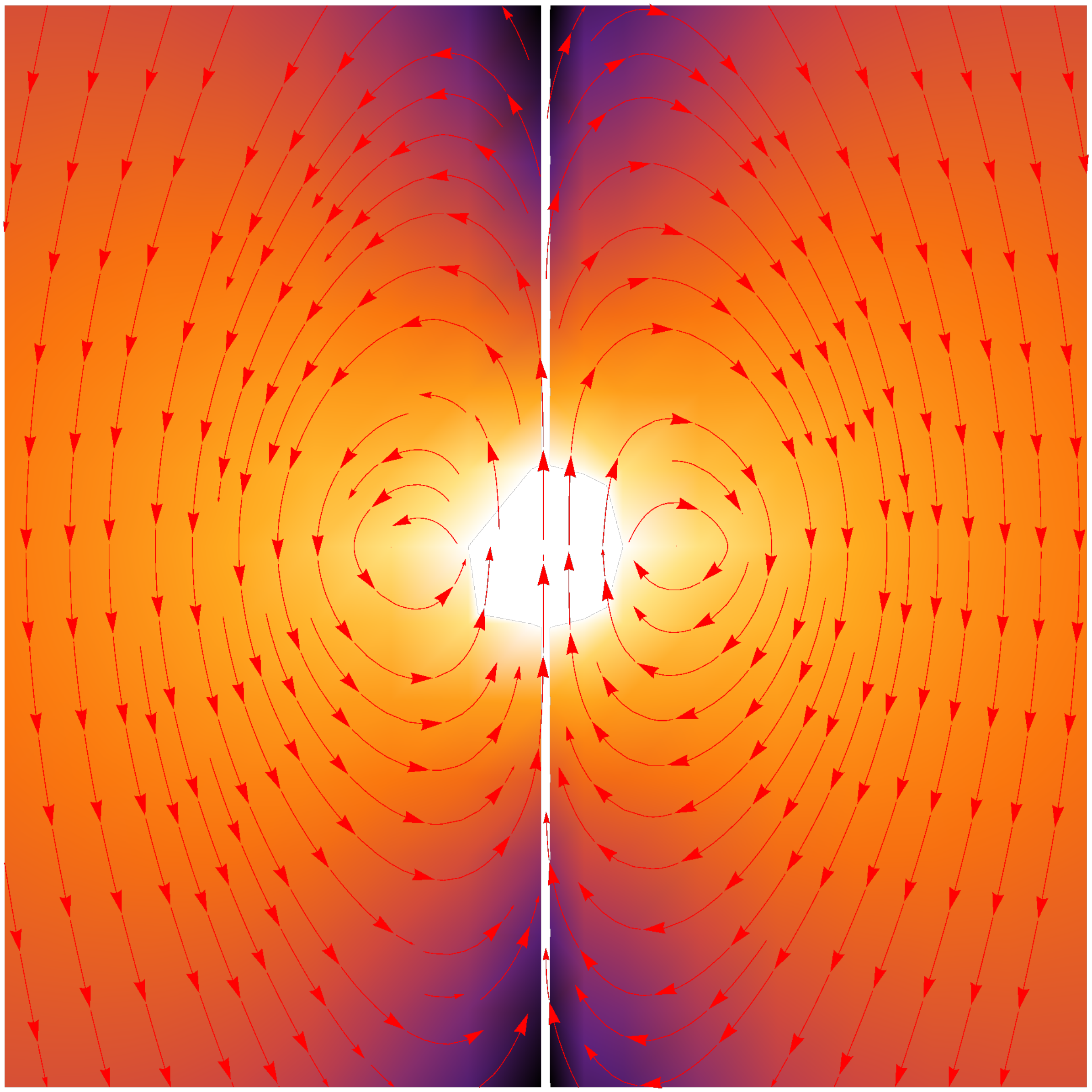
In[78]:=

```
mmin[R_, l_, r_, z_] := 4 R r / ((r + R)^2 + (z - l / 2)^2)
mplus[R_, l_, r_, z_] := 4 R r / ((r + R)^2 + (z + l / 2)^2)
n[R_, r_] := 4 R r / (r + R)^2
Br[R_, l_, r_, z_] := 1 / l / r (
   (Sqrt[(r + R)^2 + (z - l / 2)^2]
      ((2 - mmin[R, l, r, z]) EllipticK[mmin[R, l, r, z]] - 2 EllipticE[mmin[R, l, r, z]])) -
    (Sqrt[(r + R)^2 + (z + l / 2)^2] ((2 - mplus[R, l, r, z]) EllipticK[mplus[R, l, r, z]] -
        2 EllipticE[mplus[R, l, r, z]])))
Bz[R_, l_, r_, z_] := 2 / l (
    (z - l / 2) / Sqrt[(r + R)^2 + (z - l / 2)^2]
      ((r - R) / (r + R) (EllipticPi[n[R, r], mmin[R, l, r, z]] - EllipticK[mmin[R, l, r, z]])) -
     (z + l / 2) / Sqrt[(r + R)^2 + (z + l / 2)^2] ((r - R) / (r + R)
        (EllipticPi[n[R, r], mplus[R, l, r, z]] - EllipticK[mplus[R, l, r, z]]))
   ) +
  (*last term is to yield solution for (0,0,0) with surpression of power -
   (5/2) wrt distance, as should be. inserting constant δ=
   0.01 to regularize the expression for singularity in origin*)
  π / Sqrt[(2 R^2 + l^2)] / Sqrt[(r / R)^2 + (z / R)^2 + 0.01]^5

Bx[R_, l_, x_, y_, z_] := Abs[x] / x Br[R, l, Sqrt[x^2 + y ^2], z] Cos[ArcSin[y / Sqrt[x^2 + y^2]]]
By[R_, l_, x_, y_, z_] :=
 Abs[y] / y Br[R, l, Sqrt[x^2 + y ^2], z] Sin[Abs[ArcSin[y / Sqrt[x^2 + y^2]]]]
Bz[R_, l_, x_, y_, z_] := Bz[R, l, Sqrt[x^2 + y ^2], z]

(*test for numeric stability*)
Bz[5, 5, 0, 0, 0]

v[r_] := r^(-1 / 2)

size := 1
l := 0.5 * size
dr := 20 * size
rmin := 1 * size
rmax := 1000 * size

size2d := 3000 * size
size3d := 1000 * size
```

Out[86]= 36 276.

In[95]:=
```
stream = StreamPlot[
    (*accretion disk by superposing lines*)
    {Sum[v[i] × Bx[i, l, x, 0, z], {i, rmin, rmax, dr}],
     Sum[v[i] × Bz[i, l, x, 0, z], {i, rmin, rmax, dr}]}
    ,
    {x, -size2d, size2d},
    {z, -size2d, size2d},
    ImageSize → Full,
    StreamStyle → Red,
    Frame → False];
density = DensityPlot[
    Log10[
     Sqrt[
      Sum[v[i] × Bx[i, l, x, 0, z], {i, rmin, rmax, dr}]^2 +
       Sum[v[i] × Bz[i, l, x, 0, z], {i, rmin, rmax, dr}]^2]
    ]
    ,
    {x, -size2d, size2d},
    {z, -size2d, size2d}, ImageSize → Full,
    ColorFunction → "SunsetColors",
    Frame → False];

Show[density, stream]
```

Out[97]=



In[98]:=
```
vecs = VectorPlot3D[
   {Sum[v[i] × Bx[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}],
    Sum[v[i] × By[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}],
    Sum[v[i] × Bz[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}]},
   {x, -size3d, size3d},
   {y, -size3d, size3d},
   {z, -size3d * 2.1, size3d * 2.1},
   VectorPoints → 10,
   ImageSize → Full, VectorScale → {Scaled[0.1], 0.2, Automatic},
   VectorStyle → Red, Boxed → False, Axes → False];
density3d = DensityPlot3D[
   Log10[
    Sqrt[
     Sum[v[i] × Bx[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}]^2 +
      Sum[v[i] × By[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}]^2 +
      Sum[v[i] × Bz[i, l, x, y, z], {i, rmin, rmax / 4, dr * 4}]^2]]
   ,
   {x, -size3d, size3d},
   {y, -size3d, size3d},
   {z, -size3d * 2, size3d * 2}, ImageSize → Full,
   ColorFunction → "SunsetColors", Boxed → False, Axes → False];
Show[vecs, density3d]
```

Out[100]=