SpecialOrbits subpackage of KerrGeodesics

Define usage for public functions

```
BeginPackage["KerrGeodesics`SpecialOrbits`",
    {"KerrGeodesics`ConstantsOfMotion`"}];

KerrGeoPhotonSphereRadius::usage = "KerrGeoPhotonSphereRadius[a,x] returns the rad

KerrGeoISCO::usage = "KerrGeoISCO[a,x] returns the location of the innermost stabl
KerrGeoISSO::usage = "KerrGeoISSO[a,x] returns the location of the innermost stabl
KerrGeoIBSO::usage = "KerrGeoISBO[a,x] returns the location of the innermost bound

KerrGeoSeparatrix::usage = "KerrGeoSeparatrix[a,e,x] returns the value of p at the
KerrGeoBoundOrbitQ::usage = "KerrGeoBoundOrbitQ[a,p,e,x] tests if the orbital para

Begin["`Private`"];
```

Special orbits (separatrix, ISCO, ISSO etc...)

Innermost stable circular orbit (ISCO)

Schwarzschild ISCO is at r=6M

```
KerrGeoISCO[(0|0.),x_]:=6
```

Kerr inner-most circular orbit ISCO from Bardeen, Press, Teukolsky ApJ, 178, p347 (1972), Eq. 2.21

```
KerrGeoISCO[a_,x_/;x^2==1]:=Module[{M=1,Z1,Z2},
    Z1=1+(1-a^2/M^2)^(1/3) ((1+a/M)^(1/3)+(1-a/M)^(1/3));
    Z2=(3a^2/M^2 + Z1^2)^(1/2);
    M(3+Z2-x ((3-Z1)(3+Z1+2Z2)/x^2)^(1/2))
];
```

Photon Sphere

The photon sphere is at 3M for all radii in Schwarzschild

```
KerrGeoPhotonSphereRadius[(0|0.),x_]:=3
```

Radius of photon sphere for equatorial orbits from Bardeen, Press, Teukolsky ApJ, 178, p347 (1972), Eq. 2.18

```
KerrGeoPhotonSphereRadius[a_,1]:=2(1+Cos[2/3 ArcCos[-a]])
KerrGeoPhotonSphereRadius[a_,-1]:=2(1+Cos[2/3 ArcCos[a]])
```

For polar orbits the radius was given by E. Teo, General Relativity and Gravitation, v. 35, Issue 11, p. 1909-1926 (2003), Eq. (14)

```
KerrGeoPhotonSphereRadius[a_,(0|0.)]:=1+2Sqrt[1-1/3 a^2]Cos[1/3 ArcCos[(1-a^2)/(1-
```

In the extremal limit we can find the photon sphere radius exactly

```
KerrGeoPhotonSphereRadius[1,x_]:=If[x < Sqrt[3]-1, 1+Sqrt[2] Sqrt[1-x]-x, 1];
```

For all other inclinations we have to numerically find the photon sphere radius

```
KerrGeoPhotonSphereRadius[a1_?NumericQ,x0_?NumericQ/;Abs[x0]<=1]/;Precision[{a1,x0
prec=Precision[{a1,x0}];
req=KerrGeoPhotonSphereRadius[a,Sign[x0]];
rpolar=KerrGeoPhotonSphereRadius[a,0];

Φ=-((r^3-3M r^2+a^2 r+a^2 M)/(a(r-M)));
Q=-((r^3 (r^3-6M r^2+9M^2 r-4a^2 M))/(a^2 (r-M)^2));

u0Sq=((a^2-Q-Φ^2)+Sqrt[(a^2-Q-Φ^2)^2+4a^2  Q])/(2a^2);

r/.FindRoot[1-u0Sq-x0^2,Flatten[{r,(req+rpolar)/2,Sort[{req,rpolar}]}],WorkingPrec
(*The final Quiet[] is there to stop FindRoot complaining about the precision of t
This seems to be fine near the equatorial plane but might not be ideal for inclin

]
```

Innermost bound spherical orbits (IBSO)

```
KerrGeoIBSO[0,x_]:= 4
```

Equatorial IBSO results from Bardeen, Press, Teukolsky 1972

```
KerrGeoIBSO[a_,1]:= 2-a+2(1-a)^(1/2)
KerrGeoIBSO[a_,-1]:= 2+a+2(1+a)^(1/2)
```

At the IBSO E=1. Solve[KerrGeo[a,p,0,0]==1,p] to get the formula for the IBSO for polar orbits

```
KerrGeoIBSO[a_,0]:=Module[{δ},
    δ=27 a^4-8 a^6+3 Sqrt[3] Sqrt[27 a^8-16 a^10];
    1+Sqrt[12-4 a^2-(6 Sqrt[6] (-2+a^2))/Sqrt[6-2 a^2+(4 a^4)/δ^(1/3)+δ^(1/3)]-(4
]
```

```
KerrGeoIBSO[1,(0|0.)]:=1/3 (3+(54-6 Sqrt[33])^(1/3)+(6 (9+Sqrt[33]))^(1/3))
```

```
KerrGeoIBSO[a1_?NumericQ,x1_?NumericQ/;Abs[x1]<=1]/;Precision[{a1,x1}]!=∞:=Block[
prec=Precision[{a1,x}];
rph=KerrGeoPhotonSphereRadius[a,x];

E0=KerrGeoEnergy[a,ru,0,x];

While[(E0/.ru->rph+10^-n)<1,n++];
ru/.FindRoot[E0==1,{ru,rph+10^-n,10},WorkingPrecision->Max[MachinePrecision,prec-
]
```

Separatrix

Schwarzschild

```
KerrGeoSeparatrix[0,e_,x_]:= 6+2e;
```

From Glampedakis and Kennefick arXiv:gr-qc/0203086, for a=M we have Subscript[p, s]=1+e

```
KerrGeoSeparatrix[1,e_,1]:= 1+e
```

Separatrix for equatorial Kerr from Levin and Periz-Giz arXiv:1108.1819

```
KerrGeoSeparatrix[a1_,e_,x_/;Abs[x]==1]:= Module[{ru,a=a1},
    If[x==-1, a = -a];
    ru=ru/.Solve[e==(-ru^2+6 ru-8a ru^(1/2)+3a^2)/(ru^2-2ru+a^2),ru][[-1]];
    (4 ru (ru^(1/2)-a)^2)/(ru^2-2ru+a^2)
]
```

Polar ISSO in extremal case found from playing around with the equations

```
KerrGeoSeparatrix[1,0,0]:=1+Sqrt[3]+Sqrt[3+2 Sqrt[3]]
```

For e=1 the Subscript[p, s] is at 2 Subscript[r, ibso]

```
KerrGeoSeparatrix[a_,1,x_]:=2KerrGeoIBSO[a,x]
```

This method is an extension of the method in arXiv:1108.1819. See N. Warburton's notes for details.
The results of the KerrGeoSeparatrix function have also been tested against the recent analytic results in arXiv:1901.02730 (which also extends the method in arXiv:1108.1819) -- see Eqs. (26a-d) in that paper.

```
KerrGeoSeparatrix[a1_?NumericQ,e1_?NumericQ,x1_?NumericQ/;Abs[x1]<=1]/;Precision[

{E0,L0,Q0}=Values[KerrGeoConstantsOfMotion[a,ru,0,x]];
β=(-1+E0^2);

ra2=2 (-a E0+L0)^2+2 Q0+ru^2 (-1-ru β+Sqrt[1+β (L0^2+Q0-a^2 β-2 ru (1+ru β))]);

β2=ru (2+ru β-2 Sqrt[1+L0^2 β+Q0 β-2 ru β-a^2 β^2-2 ru^2 β^2]);

e2=(ra2-ru β2)/(ra2+ru β2);


prec=Precision[{a,e1,x}];
r1=KerrGeoIBSO[a,x];
ru0=ru/.FindRoot[e2==e1,{ru,(r1+10)/2,r1,10},WorkingPrecision->Max[MachinePrecisi

p=(2ra2 ru)/(ra2+ru β2)/.ru->ru0
]
```

```
KerrGeoBoundOrbitQ[a_?NumericQ,p_?NumericQ,e_?NumericQ,x_?NumericQ]:=Module[{ps},
    ps = KerrGeoSeparatrix[a,e,x];
    If[p >= ps, True, False]
]
```

Innermost stable spherical orbit (ISSO)

```
KerrGeoISSO[a_,x_/;Abs[x]==1]:=KerrGeoISCO[a,x]
```

```
KerrGeoISSO[a_,x_]:=KerrGeoSeparatrix[a,0,x]
```

Close the package

```
End[];

EndPackage[];
```