ConstantsOfMotion subpackage of KerrGeodesics

Define usage for public functions

```
BeginPackage["KerrGeodesics`ConstantsOfMotion`"];

KerrGeoEnergy::usage = "KerrGeoEnergy[a, p, e, x] returns the orbital energy."
KerrGeoAngularMomentum::usage =
 "KerrGeoAngularMomentum[a, p, e, x] returns the orbital
   angular momentum about the symmetry axis."
KerrGeoCarterConstant::usage = "KerrGeoCarterConstant[a, p,
   e, x] returns the Carter constant of the orbit."
KerrGeoConstantsOfMotion::usage =
 "KerrGeoConstantsOfMotion[a, p, e, x] returns the three constants of motion."

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

Schwarzschild (a=0)

Circular (e=0)

```
KerrGeoEnergy[0, p_, 0, x_] := (-2 + p) / Sqrt[(-3 + p) p]

KerrGeoAngularMomentum[0, p_, 0, x_] := (p x) / Sqrt[-3 + p]

KerrGeoCarterConstant[0, p_, 0, x_] := -((p^2 (-1 + x^2)) / (-3 + p))
```

Eccentric

```
KerrGeoEnergy[0, p_, e_, x_] := Sqrt[(-4 e^2 + (-2 + p)^2) / (p (-3 - e^2 + p))]

KerrGeoAngularMomentum[0, p_, e_, x_] := (p x) / Sqrt[-3 - e^2 + p]

KerrGeoCarterConstant[0, p_, e_, x_] := (p^2 (-1 + x^2)) / (3 + e^2 - p)
```

Convenience function to compute all three constants of motion

```
KerrGeoConstantsOfMotion[0, p_, e_, x_] :=
 <|"ℰ" -> KerrGeoEnergy[0, p, e, x],
  "ℒ" -> KerrGeoAngularMomentum[0, p, e, x],
  "𝒬" -> KerrGeoCarterConstant[0, p, e, x] |>
```

Kerr

Equatorial orbits (x^2 = 1)

The Carter constant is zero for all equatorial orbits

```
KerrGeoCarterConstant[a_, p_, e_, x_ /; x^2 == 1] := 0
```

Circular (e=0)

```
KerrGeoEnergy[a_, p_, 0, x_ /; x^2 == 1] :=
 ((-2 + p) Sqrt[p] + a / x) / Sqrt[2 a / x p^(3 / 2) + (-3 + p) p^2]
```

```
KerrGeoAngularMomentum[a_, p_, 0, x_ /; x^2 == 1] :=
 (a^2 - 2 a / x Sqrt[p] + p^2) / (Sqrt[2 a / x + (-3 + p) Sqrt[p]] p^(3 / 4))
```

Eccentric

Simplified from Glampedakis and Kennefick, Phys. Rev. D66 (2002) 044002, arXiv:gr-qc/0203086, Eq. 7 and appendix A

```
KerrGeoEnergy[a_, p_, e_, x_ /; x^2 == 1] :=
  Sqrt[1 - ((1 - e^2) (1 + ((-1 + e^2) (a^2 (1 + 3 e^2 + p) +
                 p (-3 - e^2 + p - 2 x Sqrt[(a^6 (-1 + e^2)^2 + a^2 (-4 e^2 + (-2 + p)^2)
                           p^2 + 2 a^4 p (-2 + p + e^2 (2 + p))) / (p^3 x^2)]))) /
            (-4 a^2 (-1 + e^2)^2 + (3 + e^2 - p)^2 p))) / p];
```

```
KerrGeoAngularMomentum[a_, p_, e_, x_ /; x^2 == 1] :=
  p x Sqrt[(a^2 (1 + 3 e^2 + p) + p (-3 - e^2 + p - 2 x Sqrt[(a^6 (-1 + e^2)^2 + a^2 (-4 e^2 +
                    (-2 + p)^2) p^2 + 2 a^4 p (-2 + p + e^2 (2 + p))) / (x^2 p^3)])) /
       ((-4 a^2 (-1 + e^2)^2 + (3 + e^2 - p)^2 p) x^2)] + a Sqrt[
     1 - ((1 - e^2) (1 + ((-1 + e^2) (a^2 (1 + 3 e^2 + p) +
                 p (-3 - e^2 + p - 2 x Sqrt[(a^6 (-1 + e^2)^2 + a^2 (-4 e^2 + (-2 + p)^2)
                           p^2 + 2 a^4 p (-2 + p + e^2 (2 + p))) / (p^3 x^2)]))) /
            (-4 a^2 (-1 + e^2)^2 + (3 + e^2 - p)^2 p))) / p];
```

Convenience function to compute all three constants of motion

```
KerrGeoConstantsOfMotion[a_, p_, e_, x : (1 | -1)] :=
 <|"ℰ" -> KerrGeoEnergy[a, p, e, x],
  "ℒ" -> KerrGeoAngularMomentum[a, p, e, x],
  "𝒬" -> KerrGeoCarterConstant[a, p, e, x] |>
```

Polar orbits (x=0)

The angular momentum is zero for all polar orbits

```
KerrGeoAngularMomentum[a_, p_, e_, (0 | 0.)] := 0
```

Spherical (e=0)

Simplified formula starting from Stoghianidis & Tsoubelis, Gen. Rel, Grav., vol. 19, No. 12, p. 1235 (1987), Eqs. (17)-(19)

```
KerrGeoEnergy[a_, p_, (0 | 0.), (0 | 0.)] :=
 Sqrt[(p (a^2 - 2 p + p^2)^2) / ((a^2 + p^2) (a^2 + a^2 p - 3 p^2 + p^3))]
```

```
KerrGeoCarterConstant[a_, p_, (0 | 0.), (0 | 0.)] :=
 (p^2 (a^4 + 2 a^2 (-2 + p) p + p^4)) / ((a^2 + p^2) ((-3 + p) p^2 + a^2 (1 + p)))
```

Eccentric

These equations were worked out by N. Warburton starting with Schmidt's formula

```
KerrGeoEnergy[a_, p_, e_, (0 | 0.)] :=
 Sqrt[-((p (a^4 (-1 + e^2)^2 + (-4 e^2 + (-2 + p)^2) p^2 + 2 a^2 p (-2 + p + e^2 (2 + p)))) /
     (a^4 (-1 + e^2)^2 (-1 + e^2 - p) + (3 + e^2 - p) p^4 -
       2 a^2 p^2 (-1 - e^4 + p + e^2 (2 + p))))]
```

```
KerrGeoCarterConstant[a_, p_, e_, (0 | 0.)] :=
 -((p^2 (a^4 (-1 + e^2)^2 + p^4 + 2 a^2 p (-2 + p + e^2 (2 + p)))) / (a^4 (-1 + e^2)^2
       (-1 + e^2 - p) + (3 + e^2 - p) p^4 - 2 a^2 p^2 (-1 - e^4 + p + e^2 (2 + p))))
```

Convenience function to compute all three constants of motion

```
KerrGeoConstantsOfMotion[a_, p_, e_, (0 | 0.)] :=
 <|"ℰ" -> KerrGeoEnergy[a, p, e, 0],
  "ℒ" -> KerrGeoAngularMomentum[a, p, e, 0],
  "𝒬" -> KerrGeoCarterConstant[a, p, e, 0] |>
```

Spherical orbits (e=0)

```
KerrGeoEnergy[a_, p_, 0, x_] :=
 √(((-3 + p) (-2 + p)^2 p^5 - 2 a^5 x (-1 + x^2) Sqrt[p^3 + a^2 p (-1 + x^2)] +
       a^4 p^2 (-1 + x^2) (4 - 5 p (-1 + x^2) + 3 p^2 (-1 + x^2)) -
       a^6 (-1 + x^2)^2 (x^2 + p^2 (-1 + x^2) - p (1 + 2 x^2)) +
       a^2 p^3 (4 - 4 x^2 + p (12 - 7 x^2) - 3 p^3 (-1 + x^2) + p^2 (-13 + 10 x^2)) +
       a (-2 p^(9/2) x Sqrt[p^2 + a^2 (-1 + x^2)] + 4 p^3 x Sqrt[p^3 + a^2 p (-1 + x^2)]) +
       2 a^3 (2 p x (-1 + x^2) Sqrt[p^3 + a^2 p (-1 + x^2)] -
          x^3 Sqrt[p^7 + a^2 p^5 (-1 + x^2)])) /
     ((p^2 - a^2 (-1 + x^2)) ((-3 + p)^2 p^4 - 2 a^2 p^2 (3 + 2 p - 3 x^2 + p^2 (-1 + x^2)) +
         a^4 (-1 + x^2) (-1 + x^2 + p^2 (-1 + x^2) - 2 p (1 + x^2)))))

KerrGeoAngularMomentum[a_, p_, 0, x_, En1_: Null] := Block[{En = En1, g, d, h, f},
If[En == Null, En = KerrGeoEnergy[a, p, 0, x]];

g = 2 a p;
d = (a^2 + (-2 + p) p) (p^2 - a^2 (-1 + x^2));
h = ((-2 + p) p - a^2 (-1 + x^2)) / x^2;
f = p^4 + a^2 (p (2 + p) - (a^2 + (-2 + p) p) (-1 + x^2));

(-En g + x Sqrt[(-d h + En^2 (g^2 + f h)) / x^2]) / h

]
```

CarterConstant and ConstantsOfMotion calculations are covered by the generic case

Generic orbits

```
KerrGeoEnergy[a_, p_, e_, x_] := Module[{r1, r2, zm, Δ, f, g, h, d, κ, ρ, ϵ, σ, η, r},

    r1 = p / (1 - e);
    r2 = p / (1 + e);

    zm = Sqrt[1 - x^2];

  Δ[r_] = r^2 - 2 r + a^2;

  f[r_] = r^4 + a^2 (r (r + 2) + zm^2 Δ[r]);
  g[r_] = 2 a r;
  h[r_] = r (r - 2) + zm^2 / (1 - zm^2) Δ[r];
  d[r_] = (r^2 + a^2 zm^2) Δ[r];

  κ = d[r1] × h[r2] - h[r1] × d[r2];
  ϵ = d[r1] × g[r2] - g[r1] × d[r2];
  ρ = f[r1] × h[r2] - h[r1] × f[r2];
  η = f[r1] × g[r2] - g[r1] × f[r2];
  σ = g[r1] × h[r2] - h[r1] × g[r2];

    Sqrt[(κ ρ + 2 ϵ σ - x 2 Sqrt[σ (σ ϵ^2 + ρ ϵ κ - η κ^2) / x^2]) / (ρ^2 + 4 η σ)]
]

KerrGeoAngularMomentum[a_, p_, e_, x_, En1_: Null] :=
 Module[{En = En1, r1, zm, Δ, f, g, h, d, r},
    If[En == Null, En = KerrGeoEnergy[a, p, e, x]];

    r1 = p / (1 - e);

    zm = Sqrt[1 - x^2];

  Δ[r_] = r^2 - 2 r + a^2;

  f[r_] = r^4 + a^2 (r (r + 2) + zm^2 Δ[r]);
  g[r_] = 2 a r;
  h[r_] = r (r - 2) + zm^2 / (1 - zm^2) Δ[r];
  d[r_] = (r^2 + a^2 zm^2) Δ[r];

  (-En g[r1] + x Sqrt[(-d[r1] × h[r1] + En^2 (g[r1]^2 + f[r1] × h[r1])) / x^2]) / h[r1]
]
```

```
KerrGeoCarterConstant[a_, p_, e_, x_, En1_: Null, L1_: Null] :=
 Module[{En = En1, L = L1, zm},
    If[En == Null, En = KerrGeoEnergy[a, p, e, x]];
    If[L == Null, L = KerrGeoAngularMomentum[a, p, e, x, En]];
  zm = Sqrt[1 - x^2];

    zm^2 (a^2 (1 - En^2) + L^2 / (1 - zm^2))
]
```

```
KerrGeoConstantsOfMotion[a_, p_, e_, x_] := Module[{En, L},
 En = KerrGeoEnergy[a, p, e, x];
 L = KerrGeoAngularMomentum[a, p, e, x];

 <|"ℰ" -> En, "ℒ" -> L, "𝒬" -> KerrGeoCarterConstant[a, p, e, x, En, L] |>
 ]
```

Close the package

```
End[];

EndPackage[];
```