Rindler metric

Written by Daniel Volinski at danielvolinski@yahoo.es

(%i2) `info:build_info()$info@version;`

(%o2)

5.38.1

(%i2) `reset()$kill(all)$`

(%i1) `derivabbrev:true$`

(%i2) `ratprint:false$`

(%i3) `fpprintprec:5$`

(%i4) `if get('draw,'version)=false then load(draw)$`

(%i5) `wxplot_size:[1024,768]$`

(%i6) `if get('optvar,'version)=false then load(optvar)$`

(%i7) `if get('rkf45,'version)=false then load(rkf45)$`

(%i8) `declare(trigsimp,evfun)$`

(%i9) `declare(s,mainvar)$`

1 Settings

(%i10) `ct_coords:[ξ,τ]$`

(%i11) `dim:length(ct_coords)$`

Line element

(%i12) `ldisplay(ds^2=line_element:ξ^2*del(τ)^2-del(ξ)^2)$`

$$ds^2 = \xi^2 \operatorname{del}(\tau)^2 - \operatorname{del}(\xi)^2 \qquad (\%t12)$$

Covariant Metric Tensor

(%i16)
```
lg:zeromatrix(dim,dim)$
for i thru dim do
lg[i,i]:coeff(expand(line_element),del(ct_coords[i])^2)$
for j thru dim do
for k thru dim do
if j≠k then lg[j,k]:1/2*coeff(coeff(line_element,del(ct_coords[j])),del(ct_coords[k]))$
ldisplay(lg)$
```

$$lg = \begin{pmatrix} -1 & 0 \\ 0 & \xi^2 \end{pmatrix} \qquad (\%t16)$$

2 Using optvar

(%i17) `depends(ct_coords,s)$`

Lagrangian

(%i18) `params:[m=1]$`

(%i19) `ldisplay(L:`$\frac{1}{2}$`*m*diff(ct_coords,s).lg.transpose(diff(ct_coords,s)))$`

$$L = \frac{m\left(\xi^2\left(\tau_s\right)^2 - \left(\xi_s\right)^2\right)}{2} \qquad (\%t19)$$

Momentum Conjugate

(%i20) `ldisplay(P_ξ:ev(diff(L,'diff(ξ,s))))$`

$$P_\xi = -m\,\left(\xi_s\right) \qquad (\%t20)$$

(%i21) `linsolve(p_ξ=P_ξ,diff(ξ,s)),factor;`

$$\left[\xi_s = -\frac{p_\xi}{m}\right] \qquad (\%o21)$$

(%i22) `ldisplay(P_τ:ev(diff(L,'diff(τ,s))))$`

$$P_\tau = m\,\xi^2\,\left(\tau_s\right) \qquad (\%t22)$$

(%i23) `linsolve(p_τ=P_τ,diff(τ,s)),factor;`

$$\left[\tau_s = \frac{p_\tau}{m\,\xi^2}\right] \qquad (\%o23)$$

Generalized Forces

(%i24) `ldisplay(F_ξ:diff(L,ξ))$`

$$F_\xi = m\xi\left(\tau_s\right)^2 \qquad (\%t24)$$

(%i25) `ldisplay(F_τ:diff(L,τ))$`

$$F_\tau = 0 \qquad (\%t25)$$

Euler-Lagrange Equations

```
(%i26) aa:el(L,ct_coords,s)$
(%i30) bb:ev(aa,eval,diff)$
(%i32) bb[1]:subst([k[0]=-E],-bb[1])$
       bb[4]:subst([k[2]=Λ],bb[4])$
(%i36) bb[1]:rhs(bb[1])=lhs(bb[1])$
       bb[2]:lhs(bb[2])-rhs(bb[2])=0$
       bb[3]:lhs(bb[3])-rhs(bb[3])=0$
       bb[4]:rhs(bb[4])=lhs(bb[4])$
```

Conservation Laws

```
(%i37) map(ldisp,part(bb,[1,4]))$
```

$$E=-\frac{m\left(\xi^2\,(\tau_s)^2-(\xi_s)^2\right)}{2}+m\,\xi^2\,(\tau_s)^2-m\,(\xi_s)^2 \qquad (\%t37)$$

$$\Lambda=m\,\xi^2\,(\tau_s) \qquad (\%t38)$$

Express the Energy in terms of the Angular Momentum

```
(%i39) linsolve(eliminate(part(bb,[1,4]),[diff(τ,s)]),E),expand;
```

$$\left[E=\frac{\Lambda^2}{2m\,\xi^2}-\frac{m\,(\xi_s)^2}{2}\right] \qquad (\%o39)$$

Equations of Motion

```
(%i40) map(ldisp,part(bb,[2,3]))$
```

$$-m\xi\,(\tau_s)^2-m\,(\xi_{ss})=0 \qquad (\%t40)$$

$$m\,\xi^2\,(\tau_{ss})+2m\xi\,(\xi_s)\,(\tau_s)=0 \qquad (\%t41)$$

Solve for second derivative of coordinates

```
(%i42) linsol:linsolve(part(bb,[2,3]),diff(ct_coords,s,2))$
(%i43) map(ldisp,linsol)$
```

$$\xi_{ss}=-\xi\,(\tau_s)^2 \qquad (\%t43)$$

$$\tau_{ss}=-\frac{2\,(\xi_s)\,(\tau_s)}{\xi} \qquad (\%t44)$$

Check Conservation of Energy

```
(%i45) subst(linsol,diff(rhs(bb[1]),s));
```

$$0 \qquad (\%o45)$$

Check Conservation of Angular momentum

```
(%i46) subst(linsol,diff(rhs(bb[4]),s));
```

$$0 \qquad (\%o46)$$

Legendre Transformation

```
(%i47) Legendre:linsolve([p_ξ=P_ξ,p_τ=P_τ],['diff(ξ,s),'diff(τ,s)])$
(%i48) map(ldisp,Legendre)$
```

$$\xi_s = -\frac{p_\xi}{m} \tag{\%t48}$$

$$\tau_s = \frac{p_\tau}{m\,\xi^2} \tag{\%t49}$$

Hamiltonian

```
(%i50) ldisplay(H:radcan(subst(Legendre,p_ξ*'diff(ξ,s)+p_τ*'diff(τ,s)-L)))$
```

$$H = -\frac{{p_\xi}^2\xi^2 - {p_\tau}^2}{2m\,\xi^2} \tag{\%t50}$$

```
(%i51) ldisplay(H:ev(p_ξ*'diff(ξ,s)+p_τ*'diff(τ,s)-L,Legendre,radcan))$
```

$$H = -\frac{{p_\xi}^2\xi^2 - {p_\tau}^2}{2m\,\xi^2} \tag{\%t51}$$

Equations of Motion

```
(%i52) Hq:makelist(Hq[i],i,1,2*dim)$
(%i56) Hq[1]:'diff(ξ,s)=diff(H,p_ξ)$
       Hq[2]:'diff(τ,s)=diff(H,p_τ)$
       Hq[3]:'diff(p_ξ,s)=-diff(H,ξ)$
       Hq[4]:'diff(p_τ,s)=-diff(H,τ)$
(%i57) map(ldisp,radcan(Hq))$
```

$$\xi_s = -\frac{p_\xi}{m} \tag{\%t57}$$

$$\tau_s = \frac{p_\tau}{m\,\xi^2} \tag{\%t58}$$

$$p_{\xi\,s} = \frac{{p_\tau}^2}{m\,\xi^3} \tag{\%t59}$$

$$p_{\tau\,s} = 0 \tag{\%t60}$$

Check Conservation of Energy

```
(%i61) depends([p_ξ,p_τ],s)$
(%i62) subst(Hq,diff(H,s)),fullratsimp;
```

$$0 \tag{\%o62}$$

Reduce Order

(%i64)
```
cv_coords:[Ξ,T]$
depends(cv_coords,s)$
```

(%i66)
```
gradef(ξ,s,Ξ)$
gradef(τ,s,T)$
```

Euler-Lagrange Equations

(%i67)
```
aa:el(L,ct_coords,s)$
```

(%i71)
```
bb:ev(aa,eval,diff)$
```

(%i73)
```
bb[1]:subst([k[0]=-E],-bb[1])$
bb[4]:subst([k[2]=Λ],bb[4])$
```

(%i77)
```
bb[1]:rhs(bb[1])=lhs(bb[1])$
bb[2]:lhs(bb[2])-rhs(bb[2])=0$
bb[3]:lhs(bb[3])-rhs(bb[3])=0$
bb[4]:rhs(bb[4])=lhs(bb[4])$
```

Conservation Laws

(%i78)
```
map(ldisp,part(bb,[1,4]))$
```

$$E=-\frac{m\,\left(T^2\,\xi^2-\Xi^2\right)}{2}+m\,T^2\,\xi^2-m\,\Xi^2 \tag{\%t78}$$

$$\Lambda=m\,T\,\xi^2 \tag{\%t79}$$

Equations of Motion

(%i80)
```
map(ldisp,part(bb,[2,3]))$
```

$$-m\,T^2\xi-m\,\left(\Xi_s\right)=0 \tag{\%t80}$$

$$m\,\left(T_s\right)\,\xi^2+2m\Xi T\xi=0 \tag{\%t81}$$

Solve for second derivative of coordinates

(%i82)
```
linsol:linsolve(part(bb,[2,3]),diff(ct_coords,s,2))$
```

(%i83)
```
map(ldisp,linsol)$
```

$$\Xi_s=-T^2\xi \tag{\%t83}$$

$$T_s=-\frac{2\Xi T}{\xi} \tag{\%t84}$$

Numerical solution (Lagrangian)

(%i85) `kill(labels)$`

(%i8)
```
funcs:append(ct_coords,cv_coords)$ldisplay(funcs)$
initial:[8,10,0.01,-0.02]$ldisplay(initial)$
odes:append(cv_coords,map('rhs,linsol))$ldisplay(odes)$
interval:[s,0,50]$ldisplay(interval)$
```

$$funcs = [\xi, \tau, \Xi, T] \tag{\%t2}$$

$$initial = [8, 10, 0.01, -0.02] \tag{\%t4}$$

$$odes = \left[\Xi, T, -T^2\xi, -\frac{2\Xi T}{\xi}\right] \tag{\%t6}$$

$$interval = [s, 0, 50] \tag{\%t8}$$

(%i9) `P:map("=",funcs,initial);`

$$[\xi = 8, \tau = 10, \Xi = 0.01, T = -0.02] \tag{P}$$

(%i10) `lgP:lg,P,params;`

$$\begin{pmatrix} -1 & 0 \\ 0 & 64 \end{pmatrix} \tag{lgP}$$

(%i11) `gVV:diff(ct_coords).lgP.transpose(diff(ct_coords));`

$$64\,T^2\,\mathrm{del}(s)^2 - \Xi^2\,\mathrm{del}(s)^2 \tag{gVV}$$

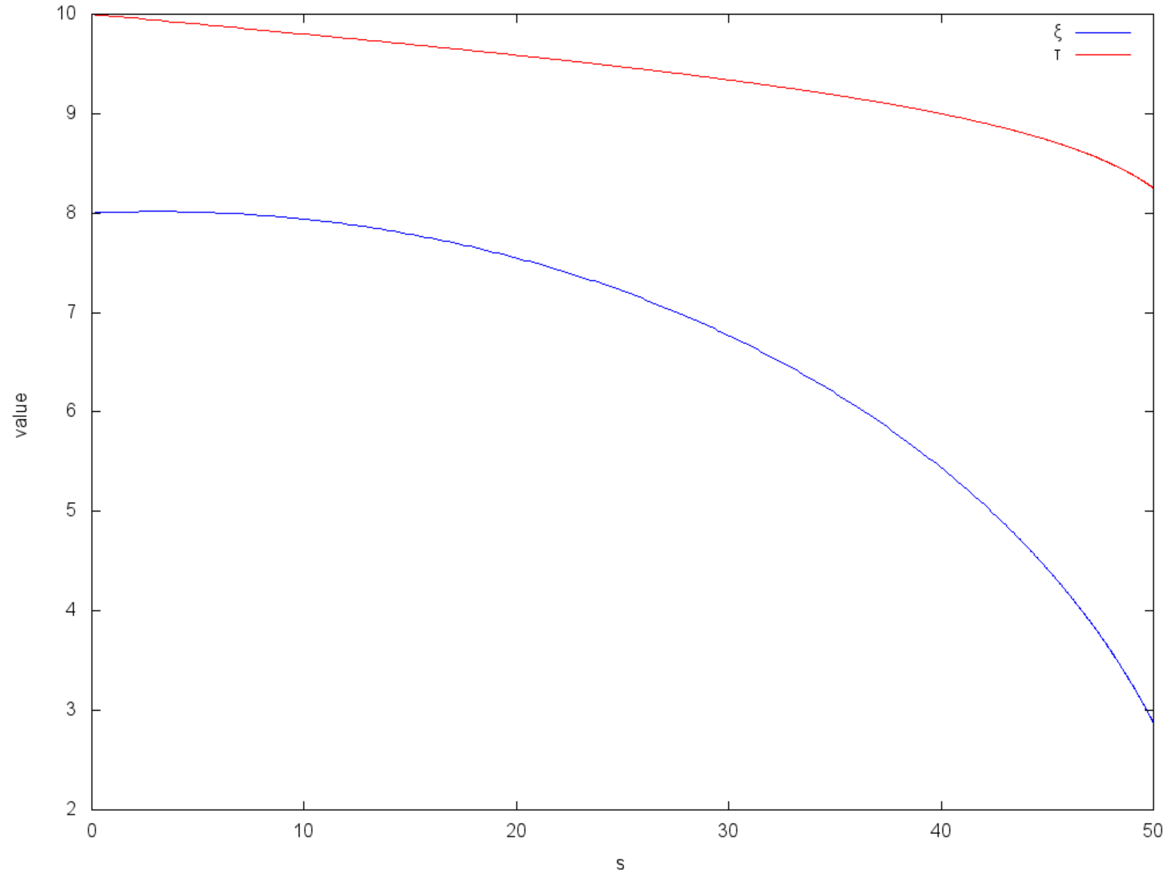
(%i12) `gVVP:gVV,P,params;`

$$0.0255\mathrm{del}(s)^2 \tag{gVVP}$$

(%i13) `rksol:rkf45(odes,funcs,initial,interval, absolute_tolerance=1E-12,report=true),params$`

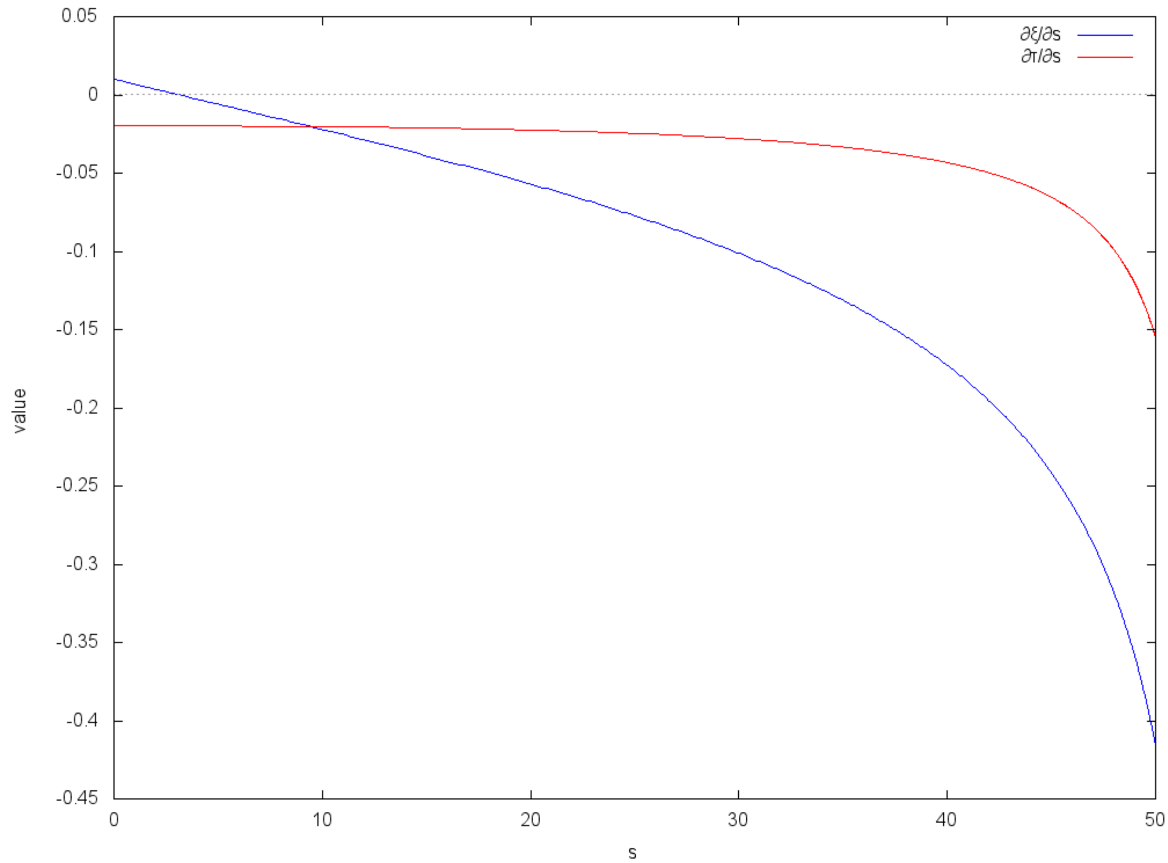
Info: rkf45:
Integration points selected:563
Total number of iterations:563
Bad steps corrected:1
Minimum estimated error:5.047210^{-14}
Maximum estimated error:5.884210^{-13}
Minimum integration step taken:0.0077667
Maximum integration step taken:0.36342

(%i14)
```
wxplot2d([[discrete,map(lambda([u],part(u,[1,2])),rksol)], [discrete,map(lambda([u],part(u,[1,3]
[style,[lines,1]],[xlabel,"s"],[ylabel,"value"], [legend,"ξ","τ"], [gnuplot_preamble,"set
key top right"])$
```



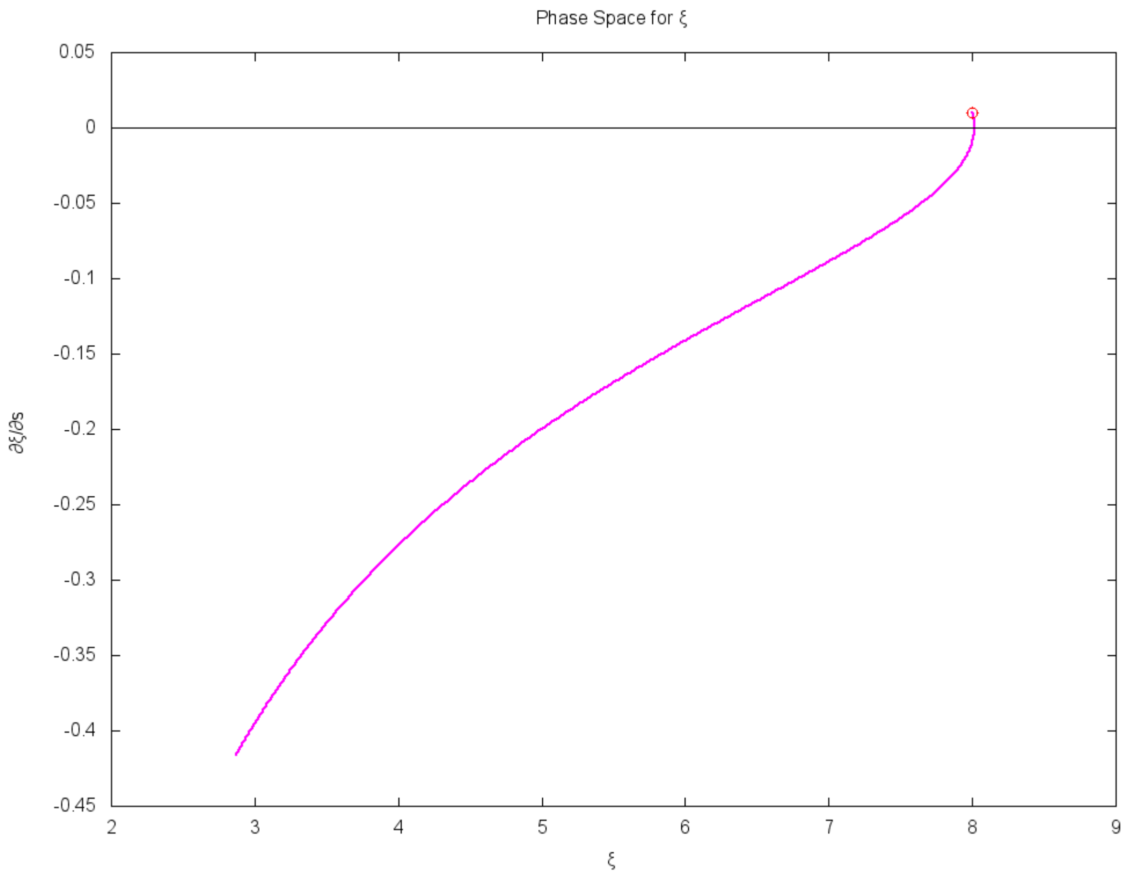
(%t14)

(%i15) wxplot2d([[discrete,map(lambda([u],part(u,[1,4])),rksol)], [discrete,map(lambda([u],part(u,[1,5]
[style,[lines,1]],[xlabel,"s"],[ylabel,"value"], [legend,"∂ξ/∂s","∂τ/∂s"],
[gnuplot_preamble,"set key top right"])$



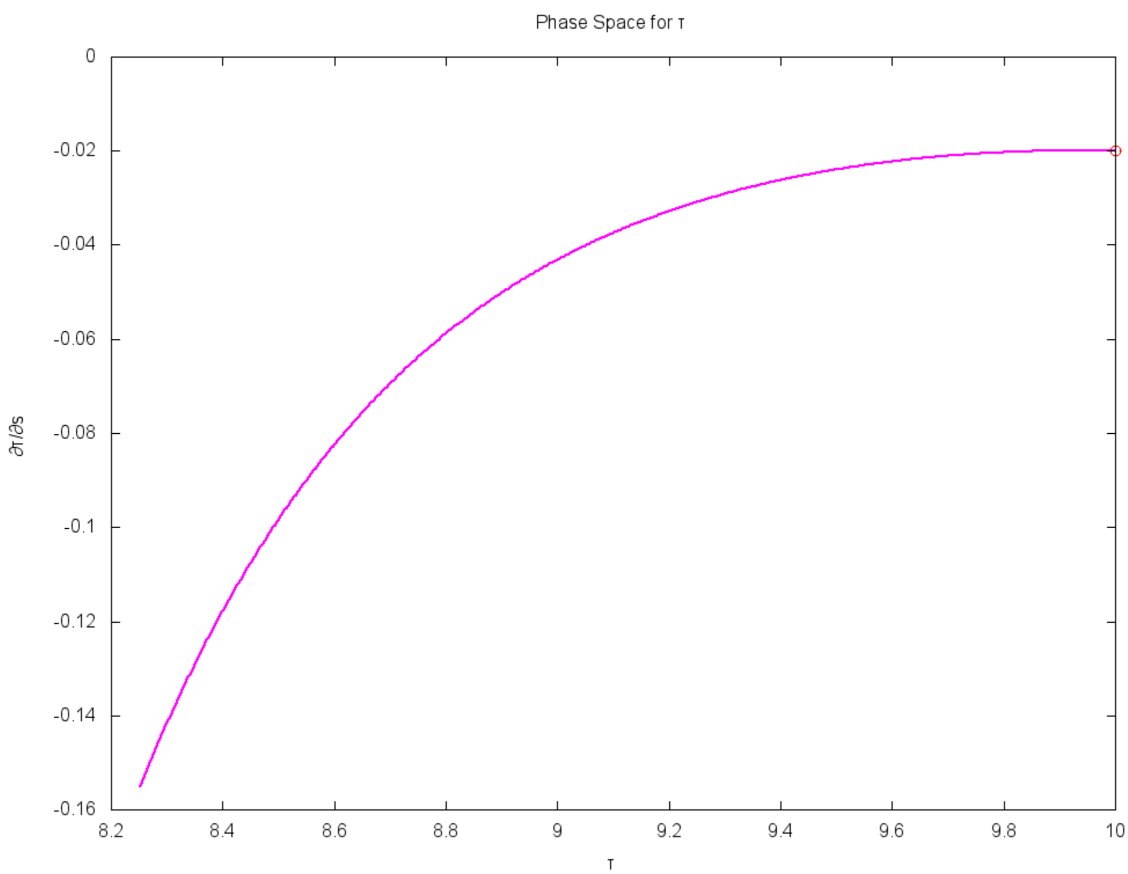
(%t15)

(%i16) `wxplot2d([[discrete,map(lambda([u],part(u,[2,4])),rksol)], [discrete,[part(initial,[1,3])]]],[ax`
`[title,"Phase Space for ξ"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,red]`
`[xlabel,"ξ"],[ylabel,"∂ξ/∂s"],[legend,false])$`



(%t16)

```
(%i17) wxplot2d([[discrete,map(lambda([u],part(u,[3,5])),rksol)], [discrete,[part(initial,[2,4])]]],[axe
[title,"Phase Space for τ"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,red]
[xlabel,"τ"],[ylabel,"∂τ/∂s"],[legend,false])$
```



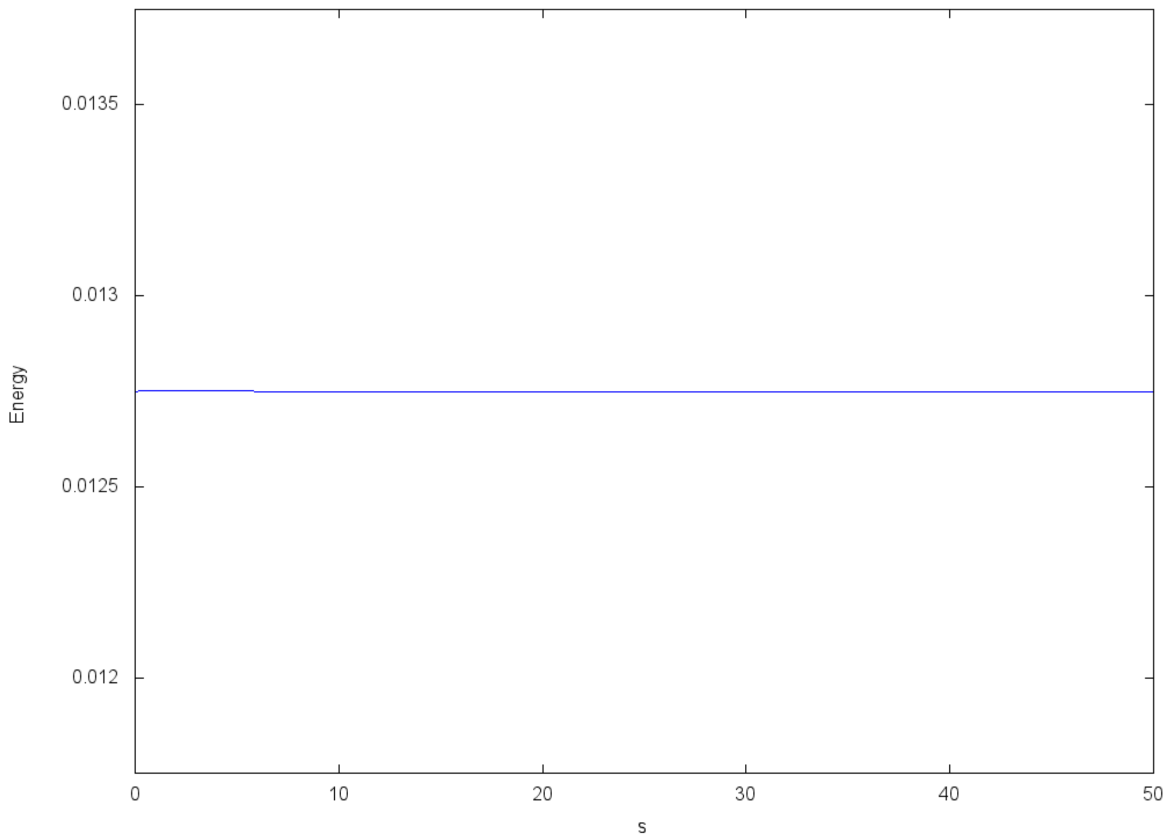
(%t17)

Check Conservation of Energy using the Numerical Data

(%i18) `W:rhs(bb[1]),P,params,numer,eval;`

$$0.01275 \quad \text{(W)}$$

(%i19) `wxplot2d([discrete,makelist([first(rkline), ev(rhs(bb[1]),map("=",funcs,rest(rkline)))],rkline,r`
`[xlabel,"s"],[ylabel,"Energy"],[y,W-0.001,W+0.001]),params$`



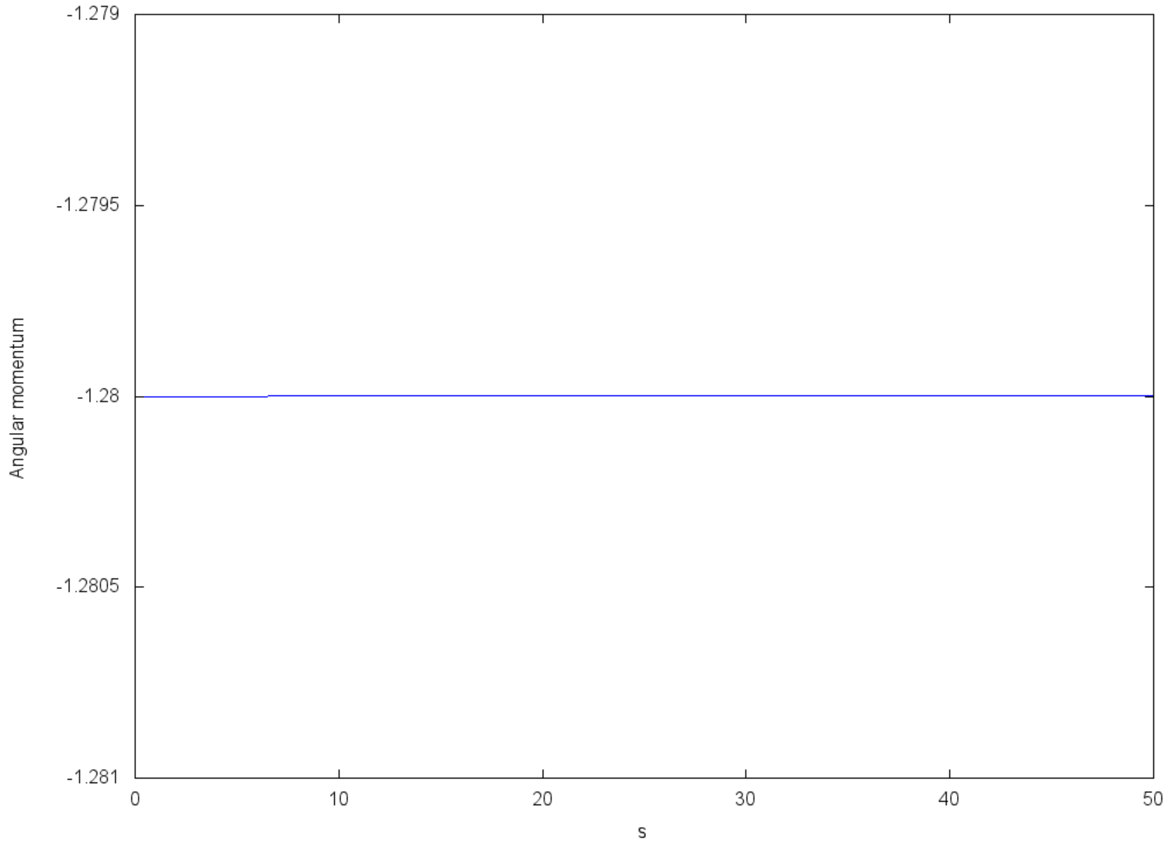
(%t19)

Check Conservation of Angular momentum using the Numerical Data

(%i20) W:rhs(bb[4]),P,params,numer,eval;

$$-1.28 \tag{W}$$

(%i21) wxplot2d([discrete,makelist([first(rkline), ev(rhs(bb[4]),map("=",funcs,rest(rkline)))],rkline,rk
[xlabel,"s"],[ylabel,"Angular momentum"],[y,W-0.001,W+0.001]),params$



(%t21)

Numerical solution (Hamiltonian)

(%i22) `kill(labels)$`

Calculate the initial values

(%i1) `initialH:initial;`

$$[8, 10, 0.01, -0.02] \qquad \text{(initialH)}$$

(%i3)
```
initialH[3]:P_ξ,['diff(ξ,s)=diff(ξ,s)],P,params,numer$
initialH[4]:P_τ,['diff(τ,s)=diff(τ,s)],P,params,numer$
```

(%i9)
```
funcs:[ξ,τ,p_ξ,p_τ]$ldisplay(funcs)$
ldisplay(initialH)$
odes:map('rhs,radcan(Hq))$ldisplay(odes)$
ldisplay(interval)$
```

$$funcs = [\xi, \tau, p_\xi, p_\tau] \qquad (\%t5)$$

$$initialH = [8, 10, -0.01, -1.28] \qquad (\%t6)$$

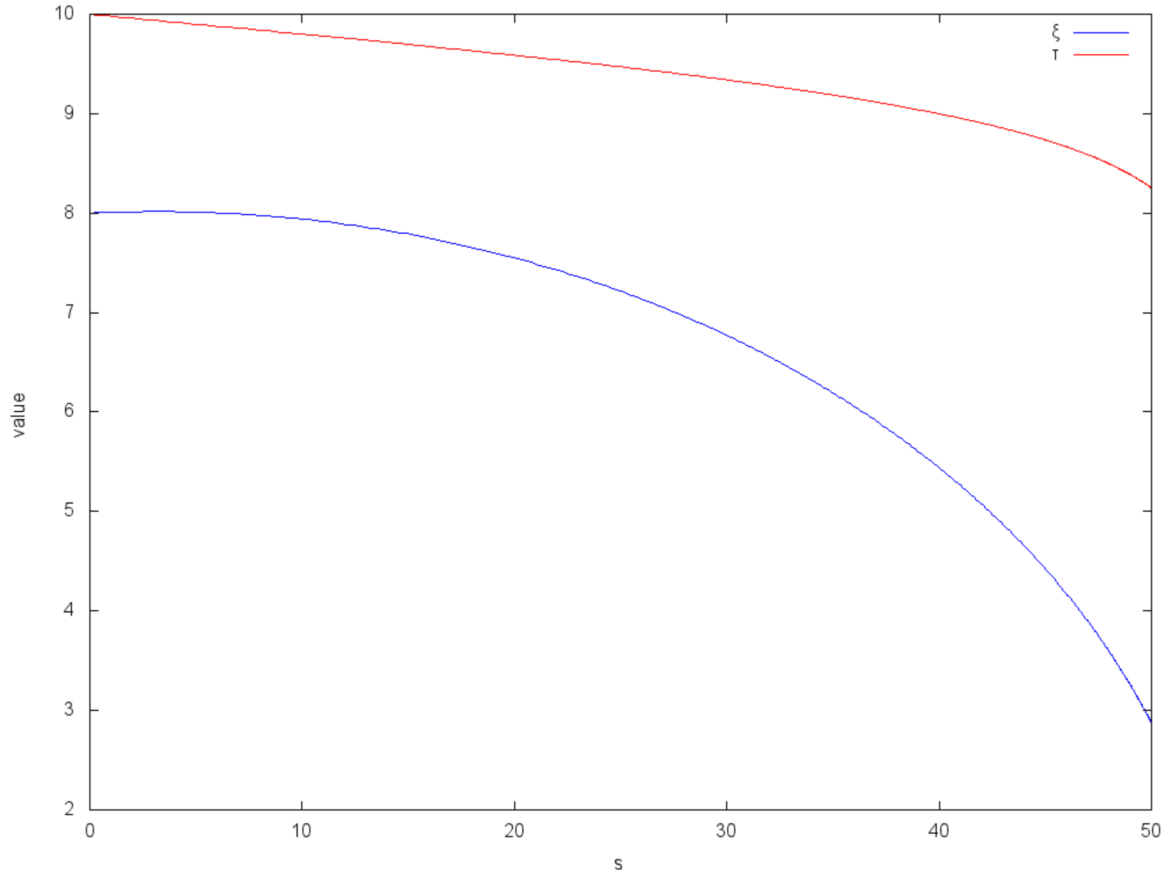
$$odes = \left[-\frac{p_\xi}{m}, \frac{p_\tau}{m\,\xi^2}, \frac{{p_\tau}^2}{m\,\xi^3}, 0\right] \qquad (\%t8)$$

$$interval = [s, 0, 50] \qquad (\%t9)$$

(%i10) `rksol:rkf45(odes,funcs,initialH,interval, absolute_tolerance=1E-12,report=true),params$`

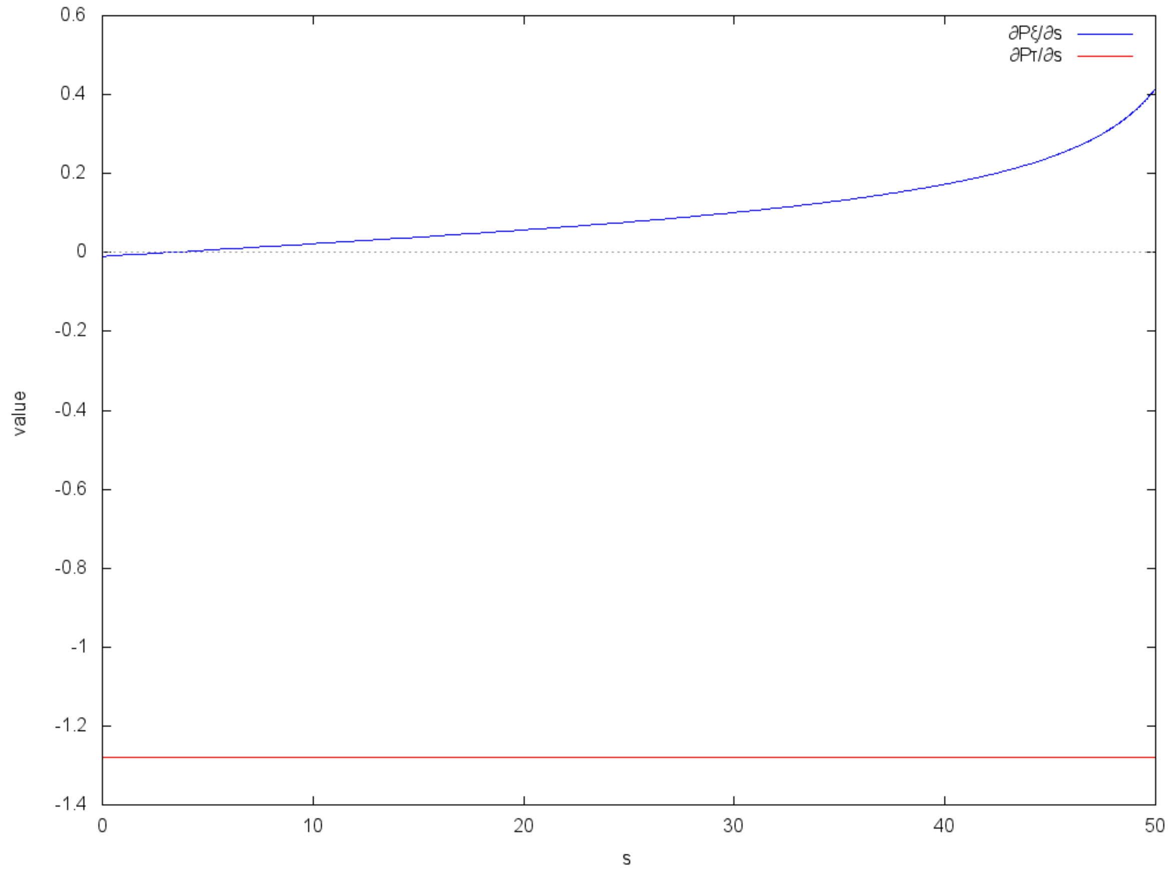
Info: rkf45:
Integration points selected:443
Total number of iterations:443
Bad steps corrected:1
Minimum estimated error:$3.7287\,10^{-18}$
Maximum estimated error:$5.6115\,10^{-13}$
Minimum integration step taken:$5.6791\,10^{-5}$
Maximum integration step taken:0.38395

(%i11) `wxplot2d([[discrete,map(lambda([u],part(u,[1,2])),rksol)], [discrete,map(lambda([u],part(u,[1,3]`
`[style,[lines,1]],[xlabel,"s"],[ylabel,"value"], [legend,"ξ","τ"], [gnuplot_preamble,"set`
`key top right"])$`



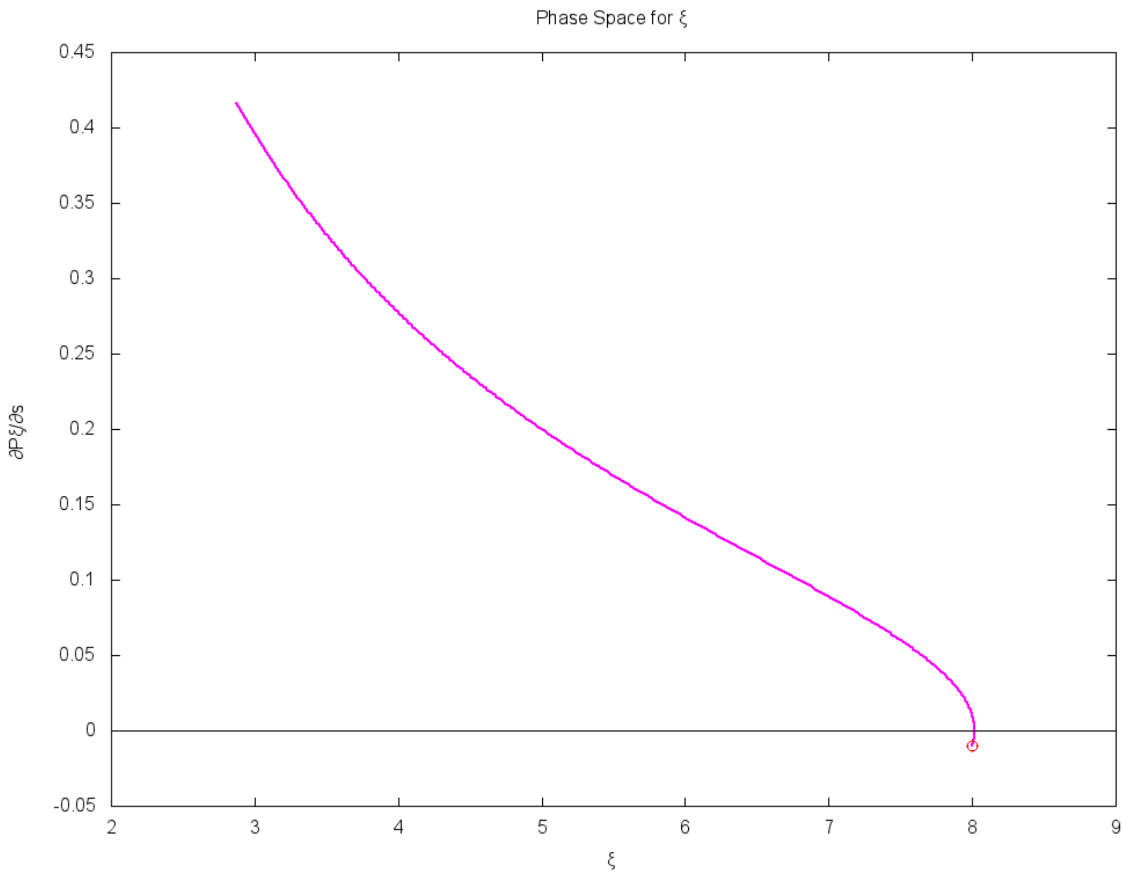
(%t11)

(%i12) `wxplot2d([[discrete,map(lambda([u],part(u,[1,4])),rksol)], [discrete,map(lambda([u],part(u,[1,5]`
`[style,[lines,1]],[xlabel,"s"],[ylabel,"value"], [legend,"∂Pξ/∂s","∂Pτ/∂s"],`
`[gnuplot_preamble,"set key top right"])$`



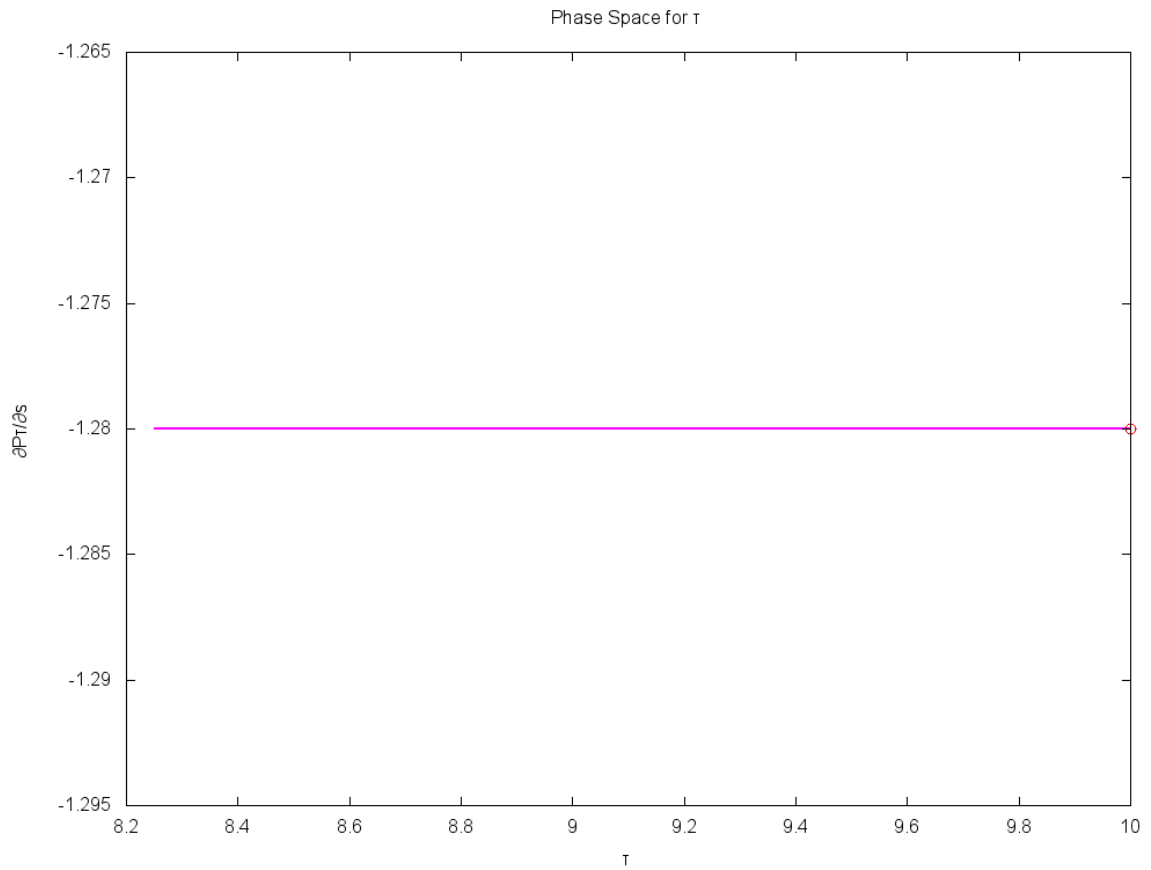
(%t12)

(%i13) `wxplot2d([[discrete,map(lambda([u],part(u,[2,4])),rksol)], [discrete,[part(initial,[1,3])]]],[axe`
`[title,"Phase Space for ξ"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,red]`
`[xlabel,"ξ"],[ylabel,"∂Pξ/∂s"],[legend,false])$`



(%t13)

(%i14)
```
wxplot2d([[discrete,map(lambda([u],part(u,[3,5])),rksol)], [discrete,[part(initial,[2,4])]]],[ax
[title,"Phase Space for τ"],[point_type,circle], [style,[lines,2],[points,3]],[color,magenta,red]
[xlabel,"τ"],[ylabel,"∂Pτ/∂s"],[legend,false])$
```



(%t14)

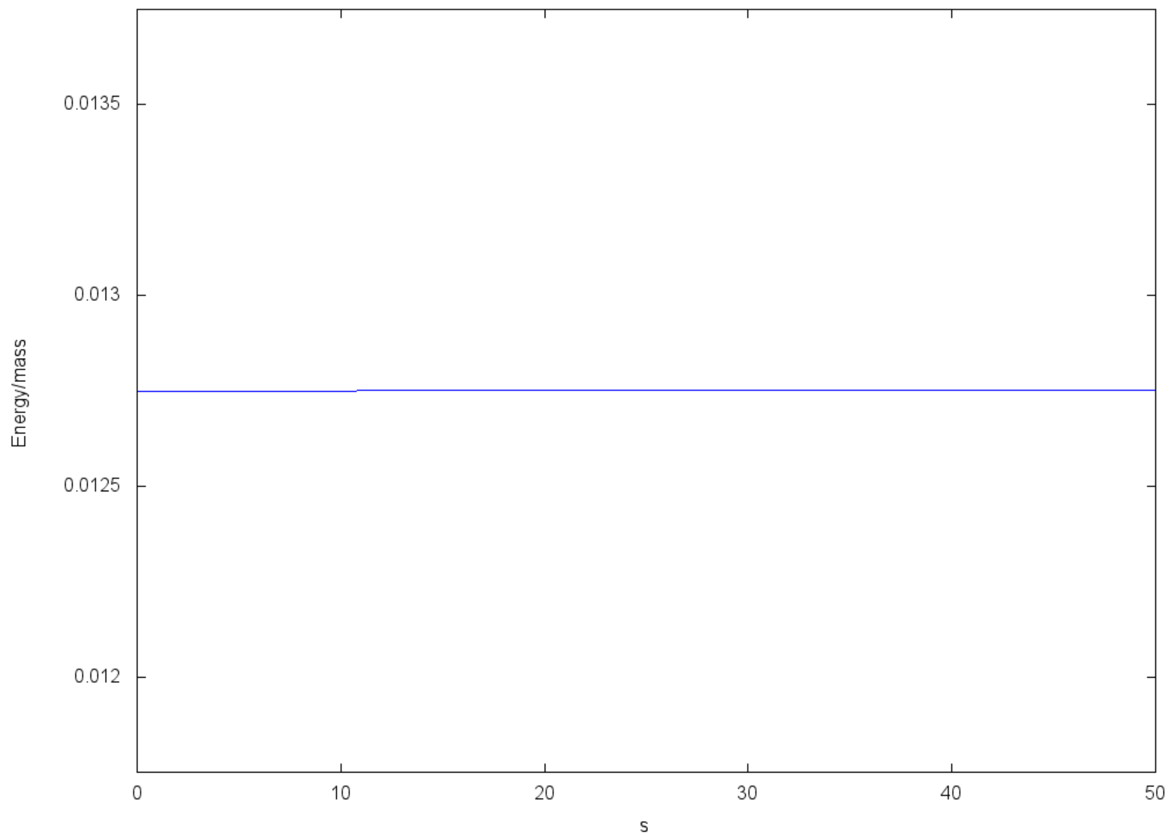
Check Conservation of Energy using the Numerical Data

(%i15)
```
P:map("=",funcs,initialH)$
```

(%i16)
```
Y:H,P,params,numer;
```

$$0.01275 \quad \text{(Y)}$$

(%i17)
```
wxplot2d([discrete,makelist([first(rkline), ev(H,map("=",funcs,rest(rkline)))],rkline,rksol)],
[xlabel,"s"],[ylabel,"Energy/mass"],[y,Y-0.001,Y+0.001]),params$
```



(%t17)

3 Using ctensor

(%i18) `kill(labels)$`

(%i1) `if get('itensor,'version)=false then load(itensor)$`

(%i2) `imetric(g)$`

(%i3) `if get('ctensor,'version)=false then load(ctensor)$`

(%i4) `dim:length(ct_coords)$`

(%i10)
```
ctrgsimp:true$
ratchristof:true$
ratriemann:true$
rateinstein:true$
ratweyl:true$
ratfac:true$
```

(%i11) `cmetric()$`

Covariant Metric tensor

(%i12) `ishow(g([μ,ν],[])=lg)$`

$$g_{\mu\nu} = \begin{pmatrix} -1 & 0 \\ 0 & \xi^2 \end{pmatrix} \tag{\%t12}$$

Contravariant Metric tensor

(%i13) `ishow(g([],[μ,ν])=ug)$`

$$g^{\mu\nu} = \begin{pmatrix} -1 & 0 \\ 0 & \frac{1}{\xi^2} \end{pmatrix} \tag{\%t13}$$

Christoffel Symbol of the first kind

(%i15)
```
christof(false)$
for i thru dim do for j:i thru dim do for k thru dim do
if lcs[i,j,k]≠0 then
ishow(Γ([ct_coords[i],ct_coords[j],ct_coords[k]],[])=lcs[i,j,k])$
```

$$\Gamma_{\xi\tau\tau} = \xi \tag{\%t15}$$

$$\Gamma_{\tau\tau\xi} = -\xi \tag{\%t15}$$

Christoffel Symbol of the second kind

(%i17)
```
christof(false)$
for i thru dim do for j:i thru dim do for k thru dim do
if mcs[i,j,k]≠0 then
ishow(Γ([ct_coords[i],ct_coords[j]],[ct_coords[k]])=mcs[i,j,k])$
```

$$\Gamma_{\xi\tau}^{\tau} = \frac{1}{\xi} \tag{\%t17}$$

$$\Gamma^{\xi}_{\tau\tau} = \xi \tag{\%t17}$$

Riemann Tensor

(%i20)
```
riemann(true)$
lriemann(false)$
uriemann(false)$
```

This spacetime is flat

Ricci Tensor

(%i22)
```
ricci(true)$
uricci(false)$
```

THIS SPACETIME IS EMPTY AND/OR FLAT

Scalar curvature

(%i23)
```
scurvature();
```

$$0 \tag{\%o23}$$

Kretschmann invariant

(%i24)
```
rinvariant();
```

$$0 \tag{\%o24}$$

Einstein Tensor

(%i26)
```
einstein(true)$
leinstein(false)$
```

THIS SPACETIME IS EMPTY AND/OR FLAT

Weyl Conformal tensor

(%i27)
```
weyl(true)$
```

ALL 2 DIMENSIONAL SPACETIMES ARE CONFORMALLY FLAT

Geodesics

(%i28)
```
cgeodesic(true)$
```

$$geod_1 = T^2\xi + \Xi_s \tag{\%t28}$$

$$geod_2 = \frac{(T_s)\,\xi + 2\Xi T}{\xi} \tag{\%t29}$$

Solve for second derivative of coordinates

(%i30)
```
linsol:linsolve(listarray(geod),diff(ct_coords,s,2))$
```

(%i31)
```
map(ldisp,linsol)$
```

$$\Xi_s = -T^2\xi \tag{\%t31}$$

$$T_s = -\frac{2\Xi T}{\xi} \tag{\%t32}$$