1. > # 3 species with Variable Parms : C P Evolution

2. > restart : with(plots) : with(StringTools) : with(DEtools) : with(plots) : with(plottools) : with(FileTools) : with(Optimization) : with(VectorCalculus) : with(LinearAlgebra) :

3. > FormatTime("%I:%M-%p----%d-%b-%Y"); currentdir( );

"02:24-PM----30-Jul-2020"

"C:\Users\nn\Documents\2 research\2017 summer non const parms\current" **(1)**

14. > ##*Figure 4 Parms Depend on S with logistic curve*

15. > $a1 := s \rightarrow \frac{La1}{1 + e^{-ma1\,(s - ca1)}}$:

16. > $a2 := s \rightarrow La2 - \frac{La2}{1 + e^{-ma2\,(s - ca2)}}$:

17. > $u1 := s \rightarrow \frac{Lu1}{1 + e^{-mu1\,(s - cu1)}}$:

18. > $u2 := s \rightarrow Lu2 - \frac{Lu2}{1 + e^{-mu2\,(s - cu2)}}$:

19. > ## Parameter set

20. > ku := 10000 : kv := 30000 : v1 := 0.001 : v2 := 0.03 : v3 := 0.1 :

21. > La1 := 0.2 : ma1 := 0.4 : ca1 := 10 :

22. > La2 := 0.00003 : ma2 := 0.6 : ca2 := 8.845 :

23. > Lu1 := 0.2 : mu1 := 1 : cu1 := 12.944 :

24. > Lu2 := 0.02 : mu2 := 1 : cu2 := 10 :

25. > evalf(u1(10)); evalf(u2(10)); a1(10); a2(10);

26. > go to top;

0.01000417112

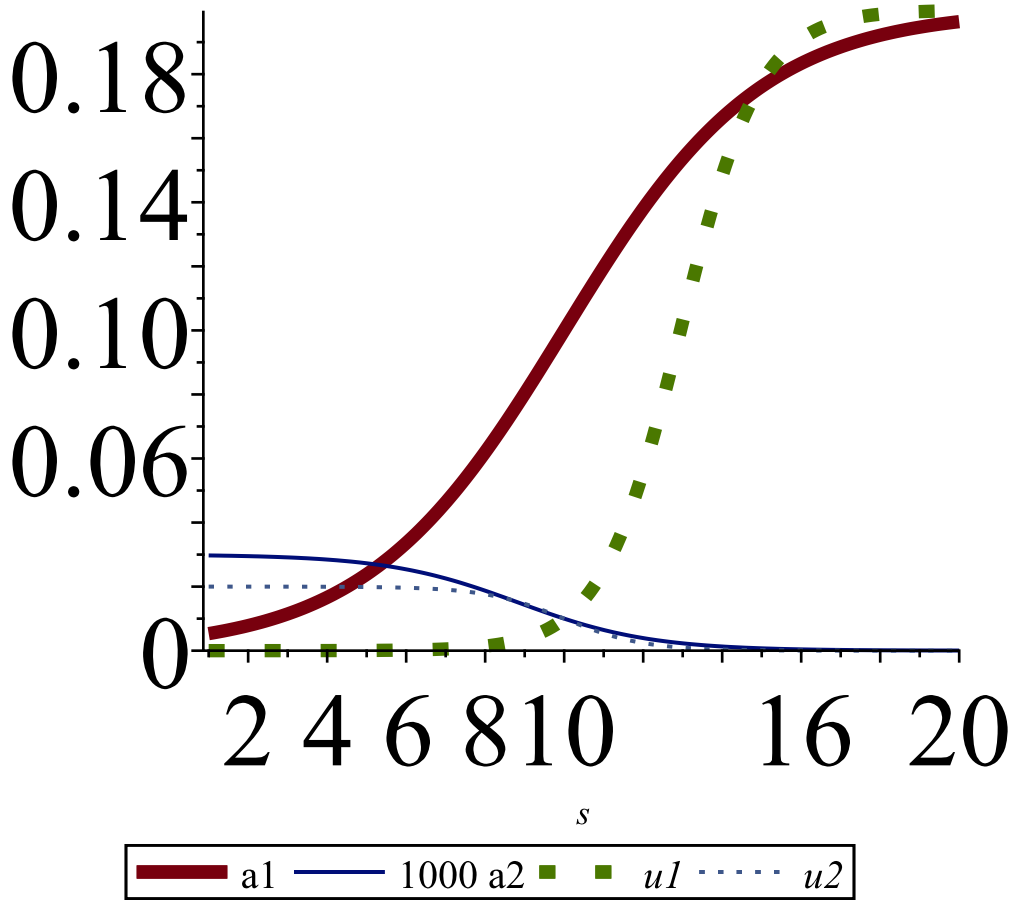
0.01000000000

0.1000000000

0.00001000098123 **(2)**

27. > unassign('s'); plot([a1(s), 1000 a2(s), u1(s), u2(s)], s = 1 ..20, thickness = [5, 1, 5, 1], linestyle = [solid, solid, dot, dot], legend = ["a1", "1000 a2",'u1','u2'], title = "Variable Parameters", font = [Roman, 30])

Variable Parameters




```
28. > ######################JITTA  parm set ########################:
29. #values from jitta case 1   s=10 corresponds to that paper;
30. > ###ku:=10000;kv:=30000;
31. > "# v1 = 0.001; v2:= 0.03; v3 = .1;";
32. > A1 := 0.1 :
33. > A2 := 0.00001 :
34. > U1 := .01 :
35. > U2 := .01 :
36. > ##### confirm consistent with JITTA for s=10
37. > a1(10) - A1 / A1 ; a2(10) - A2 / A2 ; evalf( u1(10) - U1 / U1 ); evalf( u2(10) - U2 / U2 )
```

0.

0.000098123

0.000417112

0. **(3)**

```
38. > ## The Differential Equations ############################
39. > ############ 2D subproblem  ####################
40. > #unassign('ku','kv','v1','v2','v3','u1','u2','a1','a2','s')
```

41. > $Vdot2D := s \rightarrow -v1 + v2 \cdot U \cdot \left(1 - \frac{V}{kv}\right)$:

42. > $Udot2D := s \rightarrow -u1(s) + u2(s) \cdot V \left(1 - \frac{U}{ku}\right)$:

43. > ### ########################2D equations ##########

44. > $eqV2D := s \rightarrow Vdot2D(s) = 0$:

45. > $eqU2D := s \rightarrow Udot2D(s) = 0$:

46. > $UVsol := s \rightarrow solve([eqV2D(s), eqU2D(s)], [U, V])$:

47. > ##########3 D problem variable parms

48. > $Vdot3D := s \rightarrow -v1 + v2 \cdot U \cdot \left(1 - \frac{V}{kv}\right) - v3\,A$:

49. > $Udot3D := s \rightarrow -u1(s) + u2(s) \cdot V \cdot \left(1 - \frac{U}{ku}\right)$:

50. > $Adot3D := s \rightarrow -a1(s) + a2(s) \cdot V$:

51. > $Vs3D := s \rightarrow solve(Adot3D(s) = 0, V)$:

52. > $Us3D := s \rightarrow (solve(subs(V = Vs3D(s), Udot3D(s)), U))$:

53. > $As1 := s \rightarrow solve(Vdot3D(s), A)$:

54. > $As2 := s \rightarrow subs(U = Us3D(s), As1(s))$:

55. > $As3D := s \rightarrow subs(V = Vs3D(s), As2(s))$:

56. > ########## find 3D CP near optimal S ##########

57. > $s := 10; Astar := As3D(s); Vstar := Vs3D(s); Ustar := Us3D(s)$

$$s := 10$$
$$Astar := 1999.888001$$
$$Vstar := 9999.018866$$
$$Ustar := 9998.999485 \quad \textbf{(4)}$$

58. > ##Are the values here consistent with case 1 of the JITTA paper?

59. > ##Exerpts from Table 7 NP2017

60. Case 1

61. V* 10000

62. U* 9900

63. A* 1969

64. > ####################

65. > $Vj := 10000 : Uj := 9900 : Aj := 1969$

66. > $\frac{(Vj - Vstar)}{Vj}$

67. > $\frac{(Uj - Ustar)}{Uj}$

68. > $\frac{(Aj - Astar)}{Aj}$

69. > ## Values are within 1.5%

70. > ####### at CP marginal impact of S should be 0 #############

71. > $s := 10; adot := evalf(subs(V = Vs3D(s), Adot3D(s)))$

72. > $udot := evalf(subs(V = Vs3D(s), subs(U = Us3D(s), Udot3D(s))))$

73. > $vdot := evalf(subs(V = Vs3D(s), subs(U = Us3D(s), subs(A = As3D(s), Vdot3D(s)))))$

74. > ## Pretty close !

$$Aj := 1969$$

$$0.00009811340000$$
$$-0.009999947980$$
$$-0.01568715135$$
$$s := 10$$
$$adot := 0.$$
$$udot := -2.76\ 10^{-9}$$
$$vdot := 2.\ 10^{-8} \quad \textbf{(5)}$$

```
75. > go to top;
76. > ##june 2018  movement of V· wrt S
77. > s := 11; Astar := As3D(s); Vstar := Vs3D(s); Ustar := Us3D(s); unassign('s');
78. > # note that as S gets bigger, Vstar increases, A star decreases, and Ustar decreases
```

$$s := 11$$
$$Astar := 1146.284219$$
$$Vstar := 18534.17770$$
$$Ustar := 9997.488086 \quad \textbf{(6)}$$

```
79. > s := 12; Astar := As3D(s); Vstar := Vs3D(s); Ustar := Us3D(s); unassign('s')
```

$$s := 12$$
$$Astar := -513.594406$$
$$Vstar := 35139.28063$$
$$Ustar := 9993.313134 \quad \textbf{(7)}$$

80. #Figure 6 combined CP values vs s;

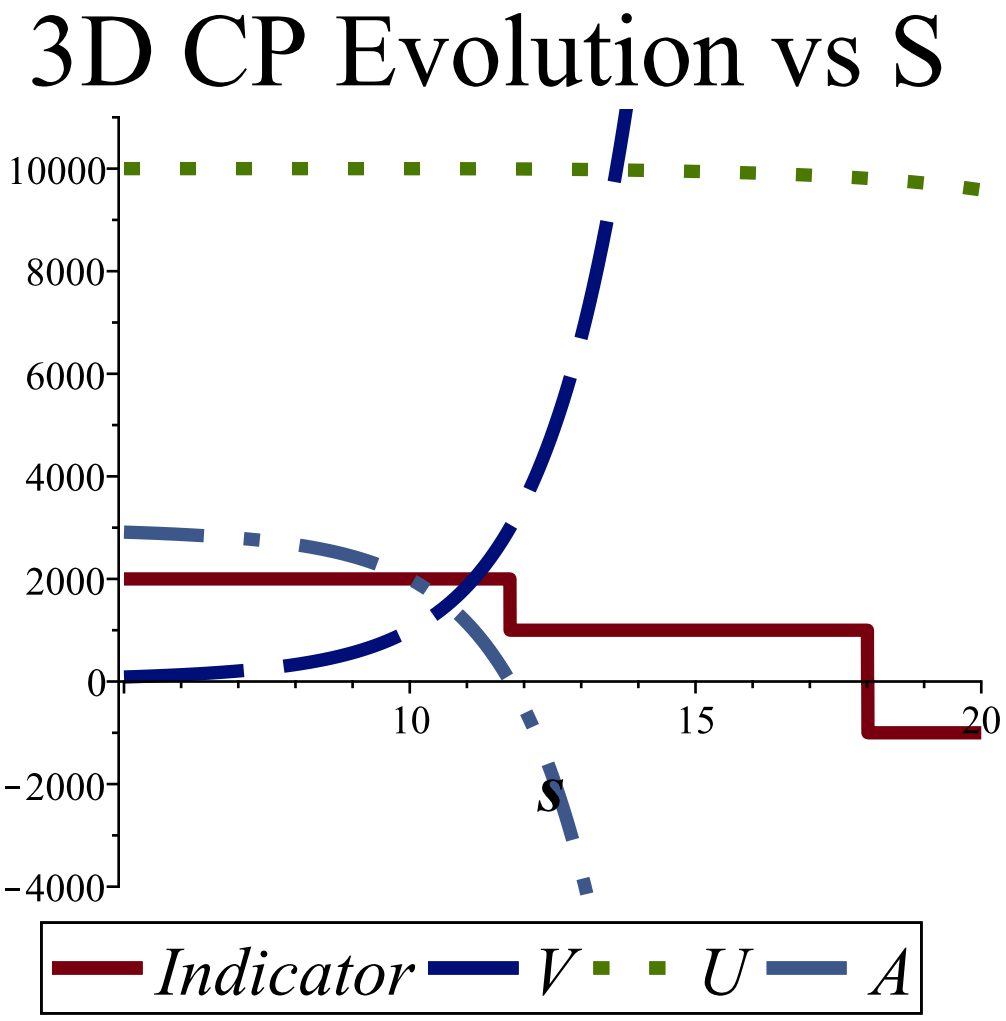
81. > #Indicator for extinct / 2 D/ 3 D

```
82. > UVsol(s) :
83. > iS := t→`if`((rhs(UVsol(t)[1,1])·rhs(UVsol(t)[1,2])) > 0, 1, -1);
          # soln of 2d problem is non zero
84. > iU := t→`if`((Vs3D(t) > 0 and Us3D(t) > 0), 1, 0); # insures a real solution
85. > iR := s→`if`(type(rhs(UVsol(s)[1,1]), nonreal), 0, 1); # U is real
86. > iA := t→`if`(As3D(t) > 0 , 2, 1);                    # A >0 (2,1)
87. > iV := s→iU(s)·iR(s)·iA(s)·iS(s);
88. > #for t from 0 to 9 do  UVsol(t)[1,2] end do;
89. > #iV= <0, complex;  0 null;  2 A>0; 1 A=0
90. > #when ind=-1, solution is not stable
91. > # when ind = 0, solution is the origin
92. > # when ind =1, there is a 2d solution
93. > #when ind =2, there is a 3d solution;
```

$$iS := t→if(0 < rhs(UVsol(t)_{1,1})\ rhs(UVsol(t)_{1,2}), 1, -1)$$
$$iU := t→if(0 < Vs3D(t) \text{ and } 0 < Us3D(t), 1, 0)$$
$$iR := s→if(type(rhs(UVsol(s)_{1,1}), nonreal), 0, 1)$$
$$iA := t→if(0 < As3D(t), 2, 1)$$
$$iV := s→iU(s)\ iR(s)\ iA(s)\ iS(s) \quad \textbf{(8)}$$

```
94. > unassign('s')
```

```
95. > plot([1000·'iV(s)', 'Vs3D(s)'/10, 'Us3D(s)', 'As3D(s)'], s = 5..20, thickness = [5, 5, 5, 5], linestyle = [solid, dash, dot, dashdot], legend = ['Indicator', 'V', 'U', 'A'], title = "3D CP Evolution vs S", titlefont = [roman, 30], legendstyle = [font = [roman, 20]], labelfont = [roman, bold, 20])
```




```
96. > go to top;
```

97. > # Figure 7 Find CP for 2 D piece ;

```
98. > unassign('s');
99. > Vs2D1 := s→rhs((UVsol(s)[1])[2]):
100. > Vs2D2 := s→rhs((UVsol(s)[2])[2]):
101. > #unassign('s');for s from 1 to 16 do s; UVsol(s); end do;: unassign('s');
102. > Vs2Dmax := s→max(Vs2D1(s), Vs2D2(s)) :
103. > Us2D1 := s→rhs(allvalues(UVsol(s)[1])[1]):
104. > Us2D2 := s→rhs(allvalues(UVsol(s)[2])[1]):
105. > Us2Dmax := s→max(Us2D1(s), Us2D2(s)):
106. > s := 1; (s); Us2Dmax(s); Vs2Dmax(s); Us2D1(s)
```

$s := 1$

1
9999.999978
29999.90000
0.03333333341 **(9)**

107. > #Combine Pieces

```
108. > cp3dV := (i)→piecewise( i ≤ 0, 0, i = 1,'Vs2Dmax'(s), Vs3D(s) )
109. > cp3dU := (i)→piecewise( i ≤ 0, 0, i ≤ 1,'Us2Dmax'(s), Us3D(s) )
110. > cp3dA := (i)→piecewise(i ≤ 0, 0, i ≤ 1, 0, As3D(s) )
```

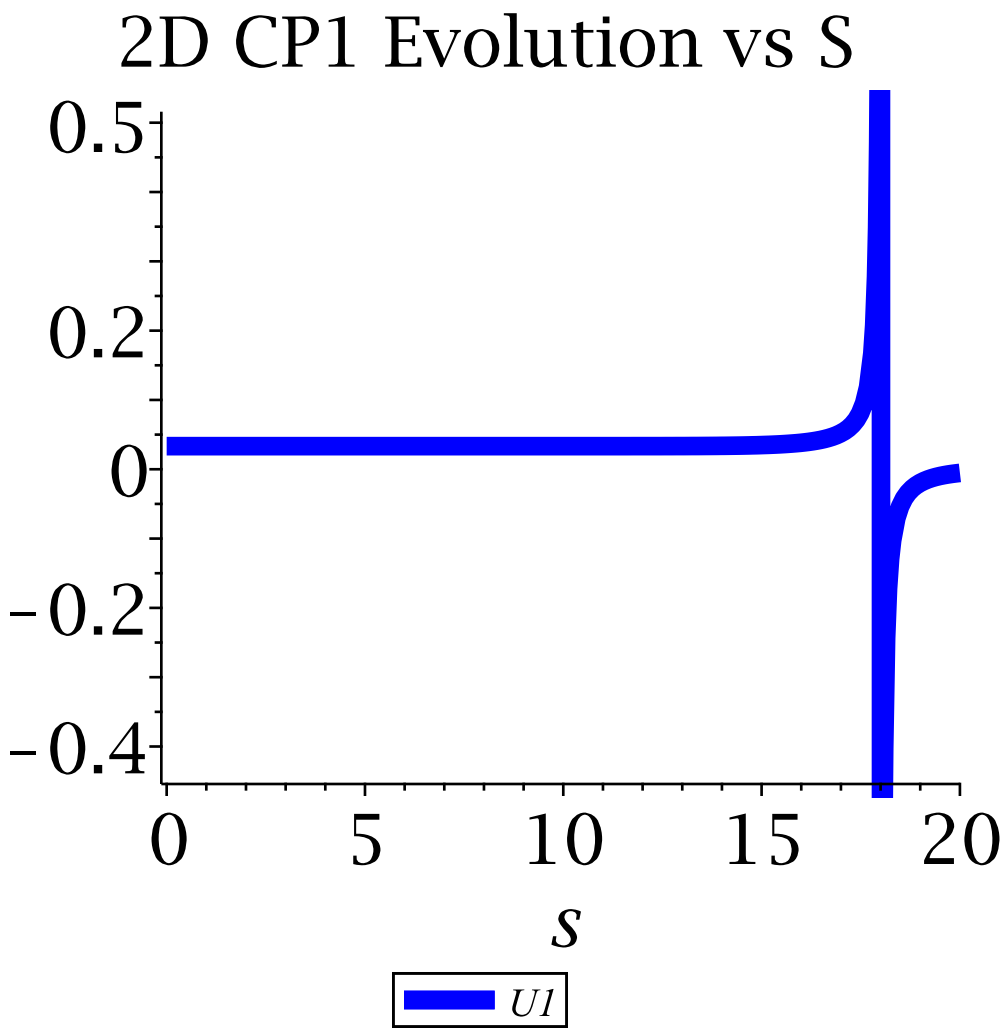
$$cp3dV := i \to piecewise(i \le 0, 0, i = 1, 'Vs2Dmax'(s), Vs3D(s))$$

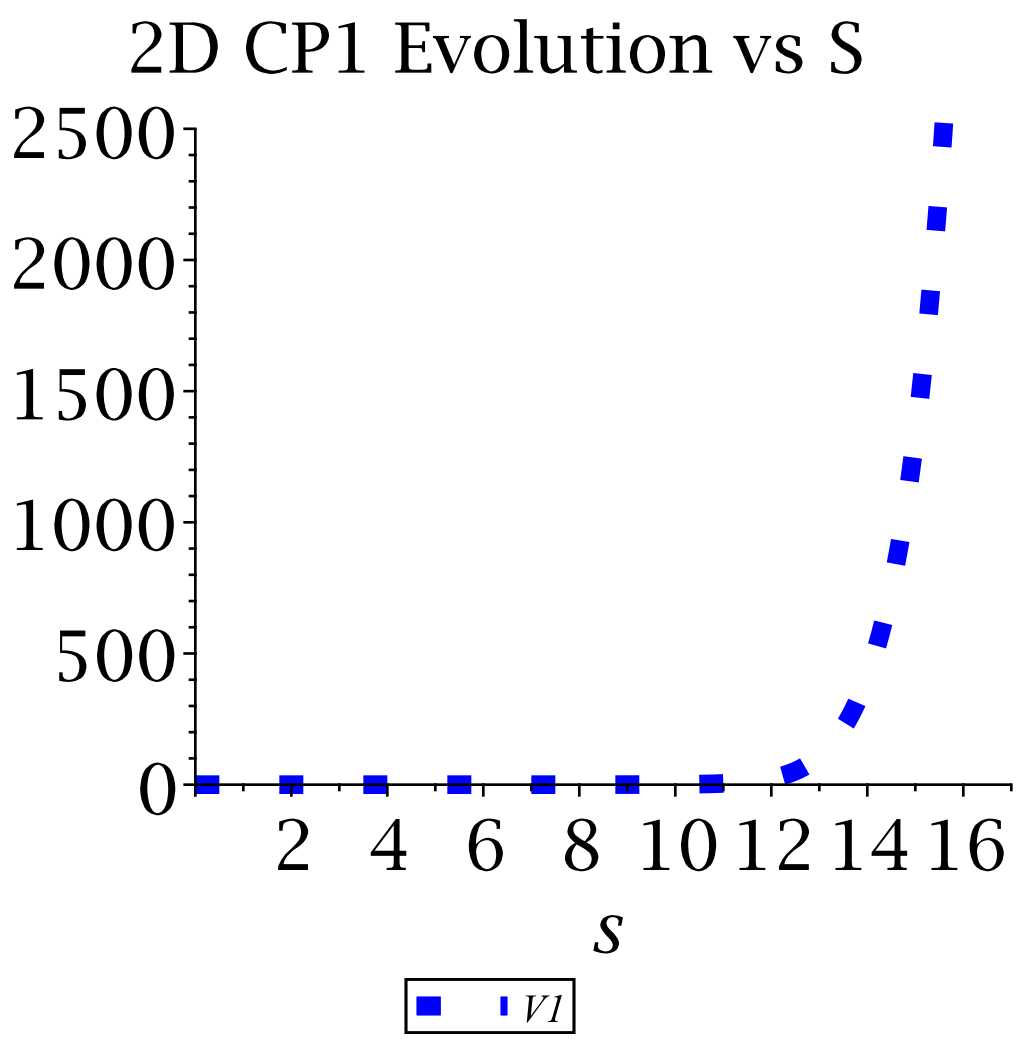
$$cp3dU := i \to piecewise(i \le 0, 0, i \le 1, 'Us2Dmax'(s), Us3D(s))$$

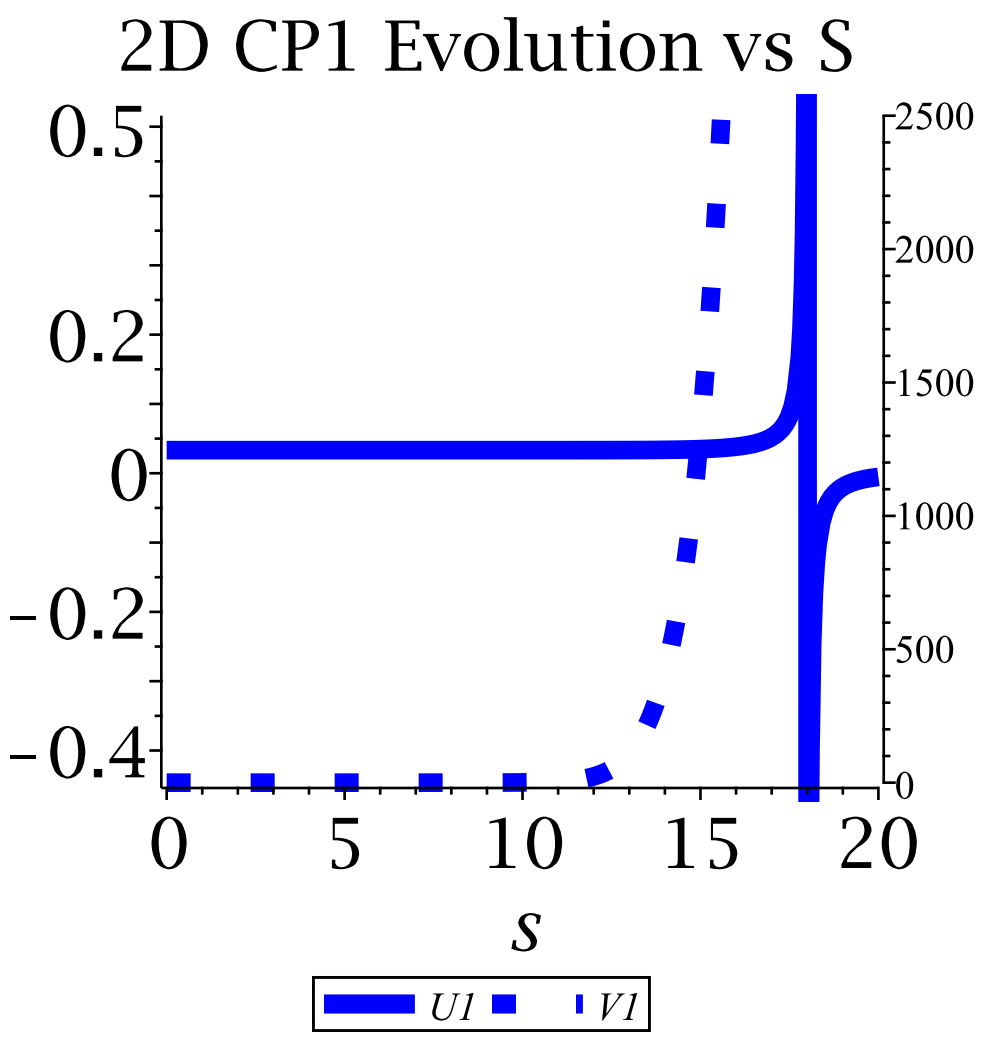
$$cp3dA := i \to piecewise(i \le 0, 0, i \le 1, 0, As3D(s)) \quad \textbf{(10)}$$

111. > #Plot CP vs S

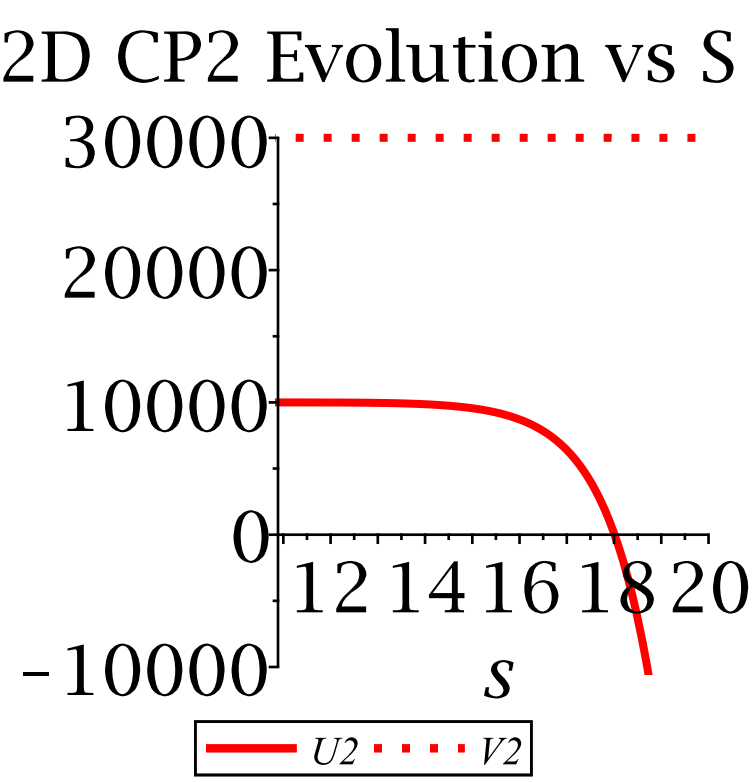
```
112. > unassign('s') : smin := 0 : smax := 20 :
113. > p2D2 := plot( ['Us2D2(s)','Vs2D2(s)'], s = smin..smax, linestyle = [solid, dot],
           size = [300, 300], color = [red, red], legend = ['U2','V2'], font = [Times,
           roman, 20], labelfont = [Times, roman, 20], title = "2D CP2 Evolution vs S",
           thickness = [3, 3]) :
114. > p2Du1 := plot( ['Us2D1(s)'], s = smin..smax, linestyle = [solid], color = [blue],
           legend = ['U1'], font = [Times, roman, 20], labelfont = [Times, roman, 20],
           title = "2D CP1 Evolution vs S", thickness = [7]) :
115. > p2Dv1 := plot( ['Vs2D1(s)'], s = smin..smax, linestyle = [dot], color = [blue],
           legend = ['V1'], font = [Times, roman, 20], labelfont = [Times, roman, 20],
           title = "2D CP1 Evolution vs S", thickness = [7, 7, 1])
116. > dualaxisplot(p2Du1, p2Dv1)
```


2D CP1 Evolution vs S
0.5
0.2
0
-0.2
-0.4
0
5
10
15
20
s
U1


2D CP1 Evolution vs S
2500
2000
1500
1000
500
0
2
4
6
8
10
12
14
16
s
V1



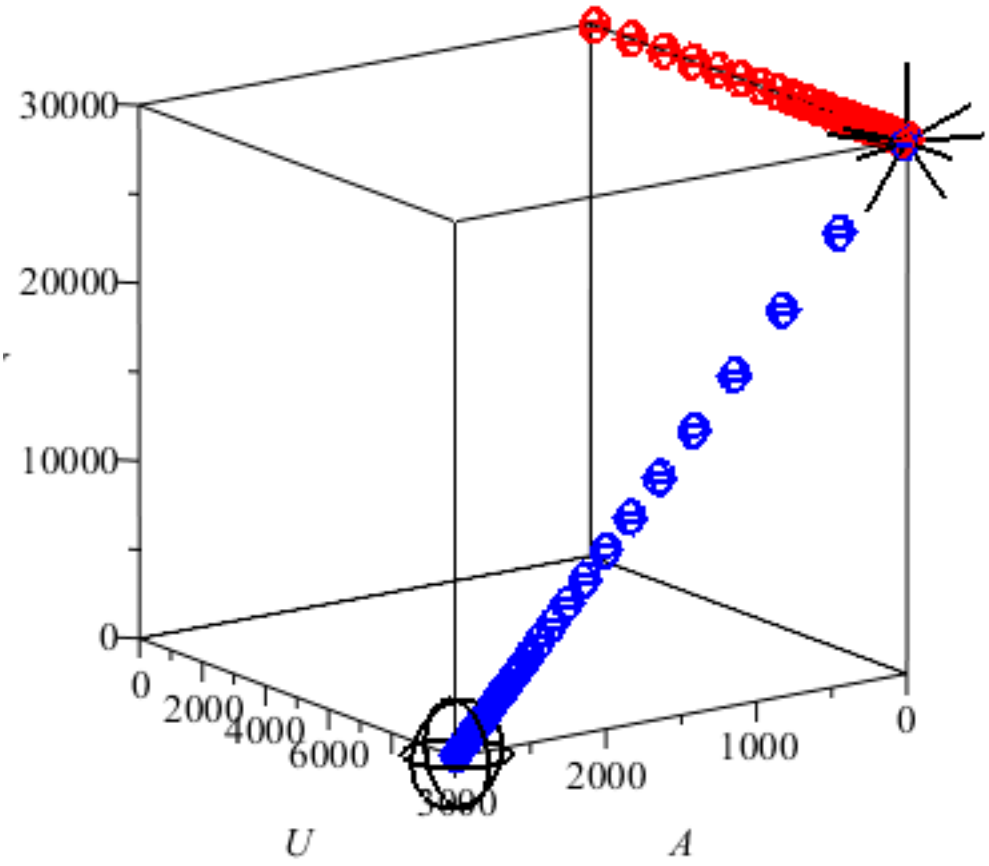

```
> #for i from 5 to 19 do
> #s:=i;  s,Us2D1(s), Vs2D1(s), Us2D2(s), Vs2D2(s), iV(s)
> #end do; unassign('s');
> display(p2D2, view=[11..20,-10000..30000]);
```



121. > # Figure 5 CP trajectory in 3 D

```
122. > unassign('s'); p := s→[evalf(As3D(s)), evalf(Us3D(s)), evalf(Vs3D(s))]:
123. > midS := 11.75; pMid := s→[evalf(As3D(midS)), evalf(Us3D(midS)),
           evalf(Vs3D(midS))]:
124. > plotStart := plot3d([p(0)], s=1..1, symbol=sphere, symbolsize=100,  style
           =point, color=black):
125. > plotMid := plot3d([pMid(s)], s=midS..midS, symbol=asterisk, symbolsize
           =300,  style=point, color=black):
126. > unassign('s'):
127. > p3d := plot3d([p(s)], s=0..12, labels=['A', 'U', 'V'], symbol=[sphere],
           symbolsize=[30], style=point, color=blue, title="3 D CP trajectory",
           titlefont=[Times, roman, 20], labelfont=[Times, roman, 20]):
                                  midS := 11.75                          (11)
128. > plot2d2 := plot3d([0,'Us2D2(s)', Vs2D2(s)'], s=12..18, symbol=sphere,
           symbolsize=30, style=point, labels=['A','U','V'], color=red, view=[0
           ..3000, -30000..30000, -30000..35000]):
129. > evalf(As3D(midS)), evalf(Us3D(midS)), evalf(Vs3D(midS))
                  8.971400, 9994.748455, 29910.13880
130. > display(plot2d2, p3d, plotMid, plotStart, view=[0..3000, 0..10000, 0..30000])
```

3 D CP trajectory



131. > *##########KEY##################*

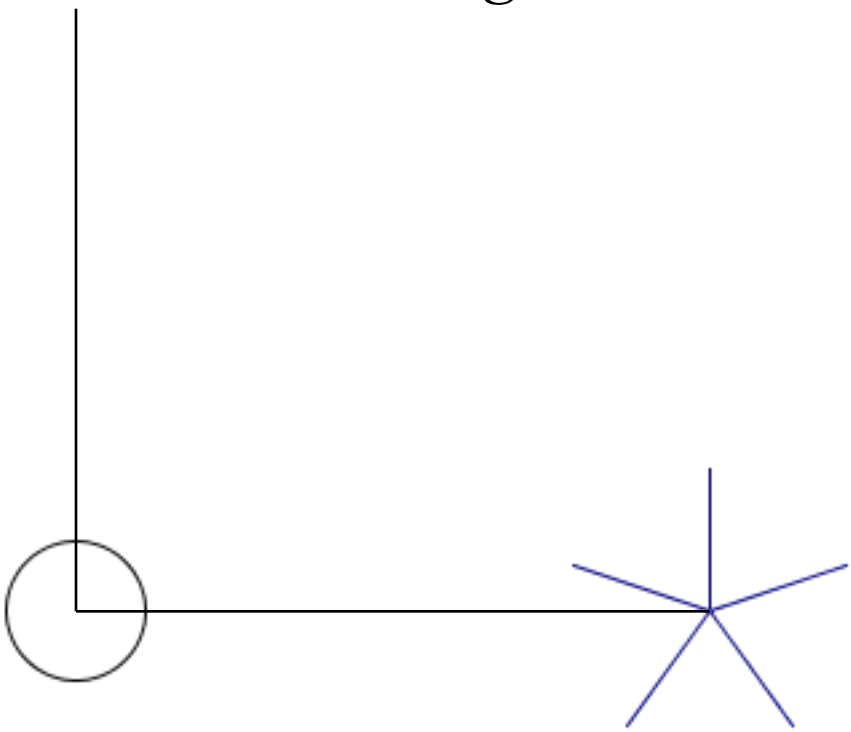
132. > *unassign*('*color*') :

133. > *p0* := *pointplot*([(0, 0)], *color* = *black*, *symbolsize* = 100, *symbol* = *circle*, *view* = [0..3, 0..0.1]) :

134. > *p2* := *pointplot*([(2, 0)], *color* = "DarkBlue", *symbolsize* = 200, *symbol* = *asterisk*) :

135. > *display*(*p0*, *p2*, *labelfont* = [*Roman*, *italic*, 20], *font* = [*Roman*, *italic*, 20], *title* = *typeset*("KEY Figure 5"), *caption* = *typeset*("Start Phase Change"), *view* = [0..2, 0.. 1], *tickmarks* = [0, 0])

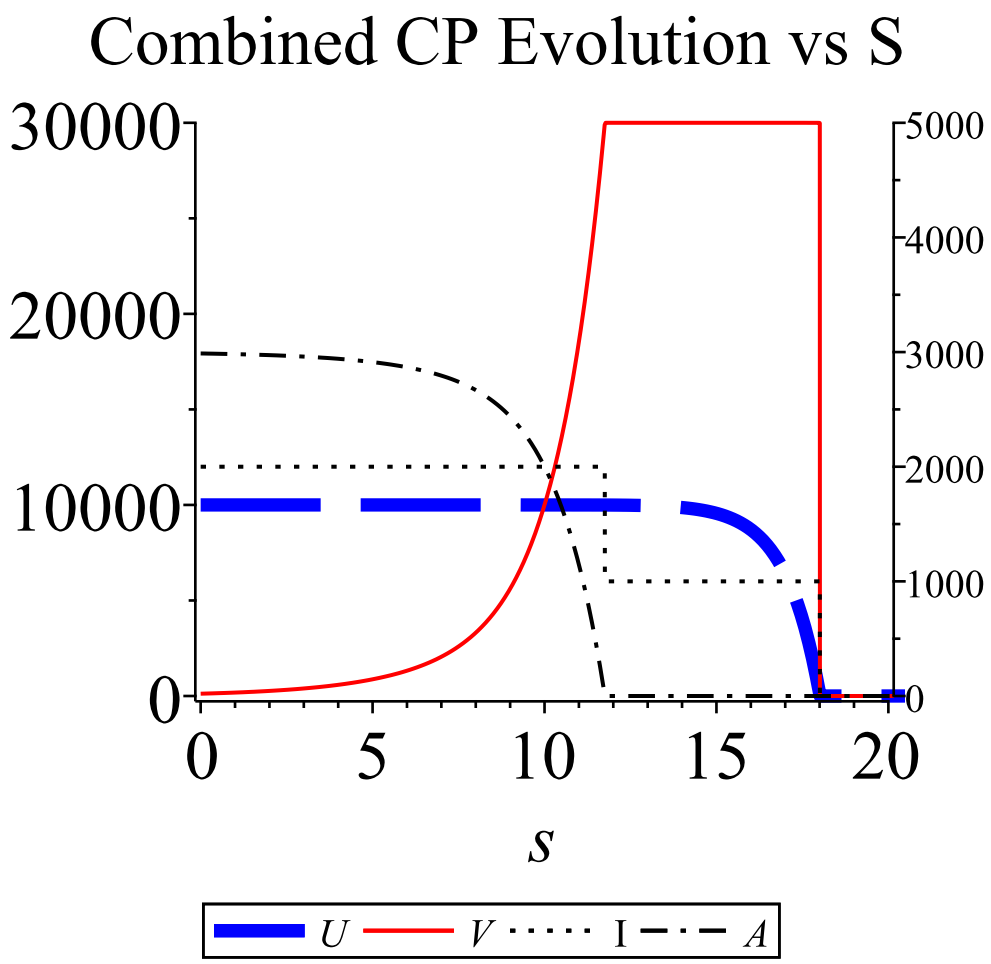
KEY Figure 5



Start Phase Change

136. # Figure 8 CP Evolution vs S;

```
137. > smin := 0 :; smax := 22 :; unassign('s');
138. > pcpV := plot([cp3dV('iV'(s))], s = smin .. smax, thickness = [1], color = red,
           linestyle = [solid], legend = 'V') :
139. > pcpA := plot([cp3dA('iV'(s))], s = smin .. smax, thickness = [1], color = [black],
           linestyle = [dashdot], legend = 'A') :
140. > pcpU := plot([cp3dU('iV(s)')], s = smin .. smax, thickness = [5, 1], color
           = [blue], linestyle = [dash], legend = 'U') :
141. > pcpInd := plot(1000·'iV'(s), s = smin ..smax, linestyle = dot, thickness = 1, color
           = black, legend = 'I') :
142. > # combine 2D and 3D plots
143. > dp1 := display((pcpInd, pcpA), view = [0 ..20, 0 ..5000]) : dp2 :=
           display((pcpU, pcpV), view = [0 ..20, 0 ..30000], font = [roman, 20], title
           = "Combined CP Evolution vs S") : dualaxisplot(dp2, dp1, caption
           = typeset("v1 = ", v1, " v2=", v2, U, V left, A right));
```



144. > go to top;

145. > ## Figure 9 CP Evolution vs S; v1 v2 altered

146. > *v1* := 0.3; *v2* := 0.0001;

147. > *smin* := 0 :; *smax* := 20 :; *unassign*('s');

148. > *pcpV* := *plot*([*cp3dV*('iV'(*s*))], *s* = *smin* .. *smax*, *thickness* = [1], *color* = *red*, *linestyle* = [*solid*], *legend* = 'V') :

149. > *pcpA* := *plot*([*cp3dA*('iV'(*s*))], *s* = *smin* .. *smax*, *thickness* = [1], *color* = [*black*], *linestyle* = [*dashdot*], *legend* = 'A') :

150. > *pcpU* := *plot*([*cp3dU*('iV(*s*)')], *s* = *smin* .. *smax*, *thickness* = [5, 1], *color* = [*blue*], *linestyle* = [*dash*], *legend* = 'U') :

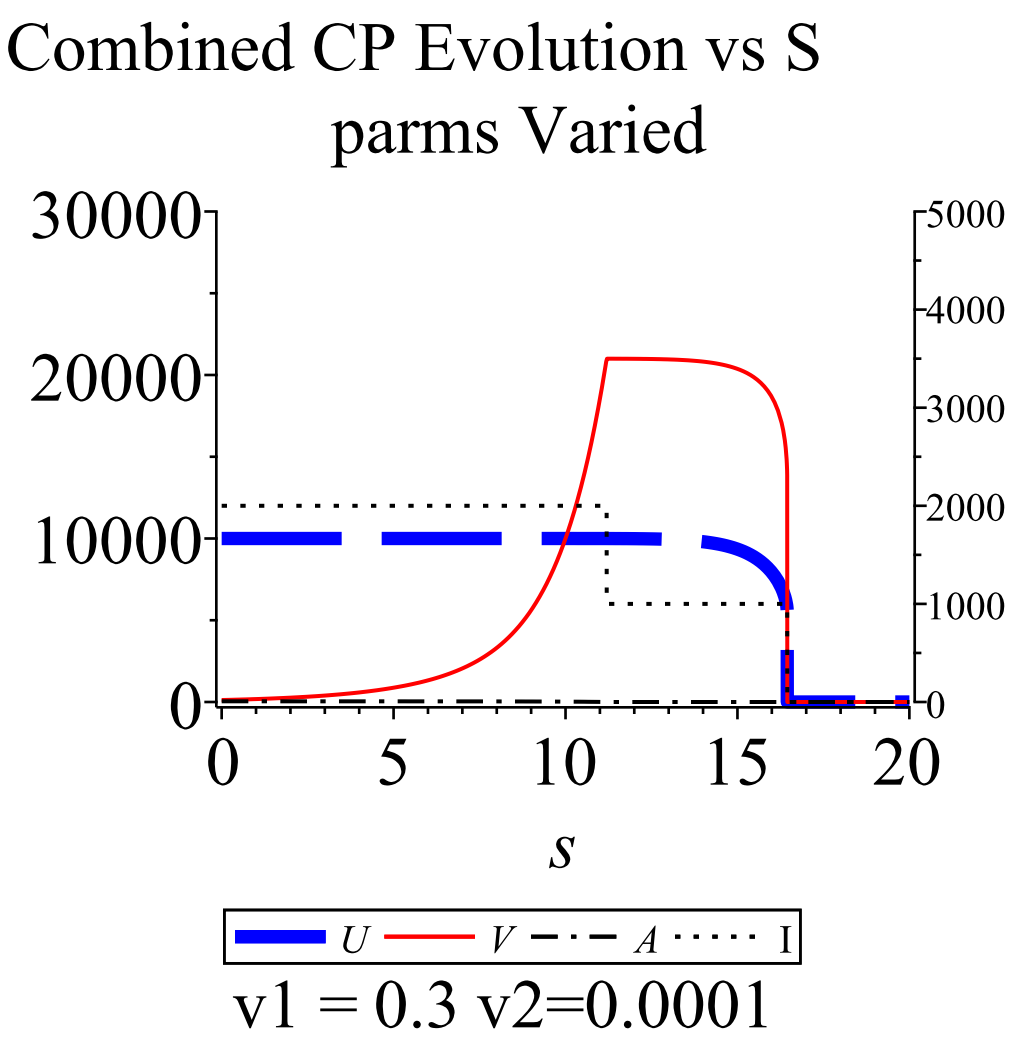
151. > *pcpInd* := *plot*(1000·'iV'(*s*), *s* = *smin* ..*smax*, *linestyle* = *dot*, *thickness* = 1, *color* = *black*, *legend* = 'I') :

152. > # *combine 2D and 3D plots*

153. > *dp1* := *display*((*pcpA*, *pcpInd*), *view* = [0 ..20, 0 ..5000]) : *dp2* := *display*((*pcpU*, *pcpV*), *view* = [0 ..20, 0 ..30000], *font* = [*roman*, 20], *title* = "Combined CP Evolution vs S parms Varied") : *dualaxisplot*(*dp2*, *dp1*, *caption* = *typeset*("v1 = ", *v1*, " v2=", *v2*));

$$v1 := 0.3$$

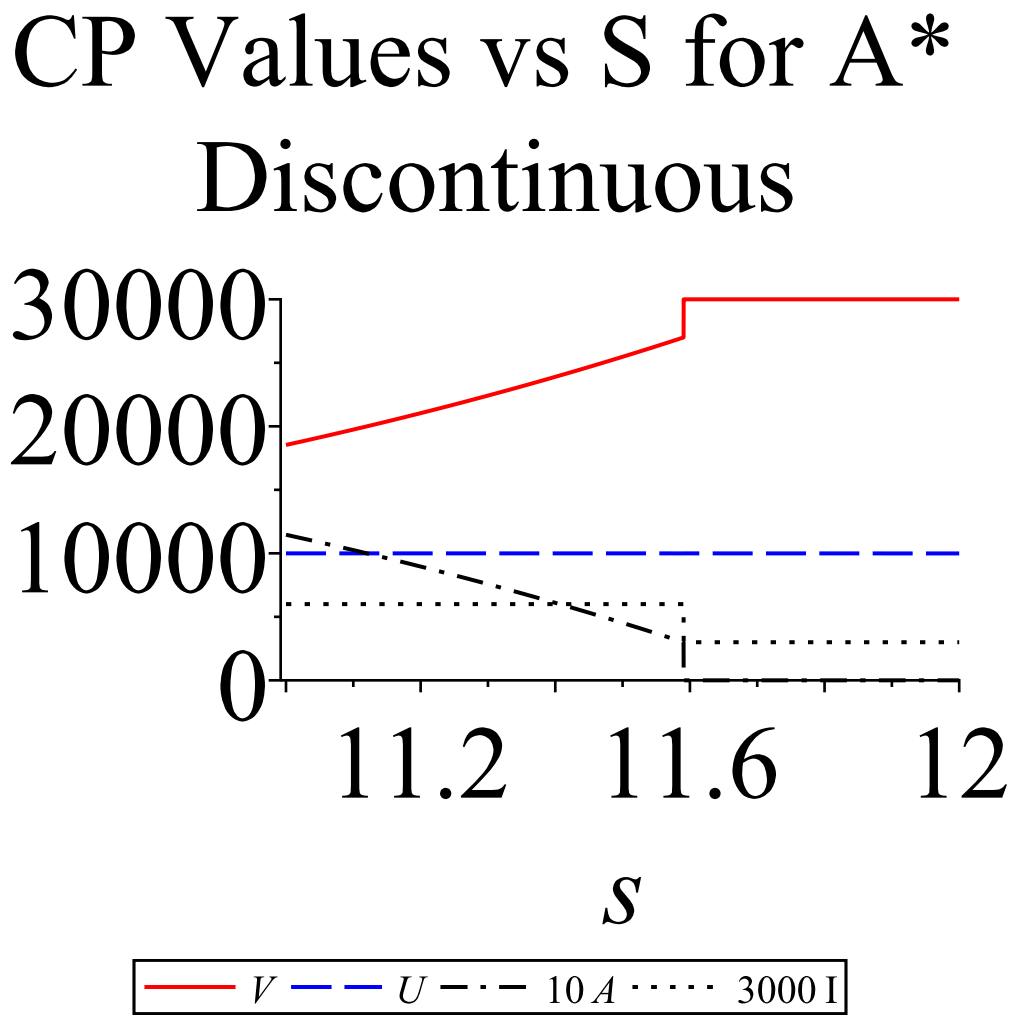
$$v2 := 0.0001$$



154. > go to top;

155. #Figure 10 3 D problem step function for A*;

```
156. > v1 := 0.001 : v2 := 0.03 : ##reset v1 and v2 to original values
157. > As3Dreg := s→subs(V = Vs3D(s), As2(s)) :
158. > Amin := 300 : As3D := s→ `if`(As3Dreg(s) < Amin, 0, As3Dreg(s)) :
159. > smin := 11 :; smax := 12 :; unassign('s');
160. > pcpV := plot([cp3dV('iV'(s))], s = smin .. smax, thickness = [1], color = red,
        linestyle = [solid], legend = 'V') :
161. > pcpA := plot([10· cp3dA('iV'(s))], s = smin .. smax, thickness = [1], color
        = [black], linestyle = [dashdot], legend = '10 A') :
162. > pcpU := plot([ cp3dU('iV'(s))], s = smin .. smax, thickness = [1], color = [blue],
        linestyle = [dash], legend = 'U') :
163. > pcpInd := plot(3000·'iV'(s), s = smin ..smax, linestyle = dot, thickness = 1, color
        = black, legend = '3000 I') :
164. > unassign('s');
165. > display(pcpV, pcpU, pcpA, pcpInd, title
        = typeset("CP Values vs S for A· Discontinuous"), font = [Roman, 30])
```



166. > ##########end of CP EVOLUTION##########