

```
In [1]: #import necessary packages
import matplotlib.pyplot as plt
import numpy as np
import sympy
from IPython.display import Image
from tqdm import trange

#Set up sympy to display outputs after a latex conversion.
from sympy.interactive import printing
printing.init_printing(use_latex=True)

# Import custom module
import SMB_Code_models_equilibria as models_equil
```

1 Where does an increase in ϵ cause a discontinuous increase in disease burden and what is the increase in disease burden.

```
In [2]: param values = {'N':1,'epsilon':0.4,'beta W': 0.04 , 'beta Z': 0.015,'gamma': 0.01,'gamma
```

```
In [3]: epsilon_vals = np. arange (0, 1.001, 0.001)
S_endem = []
I_W_endem = []
I_Z_endem = []

for i in trange(len(epsilon_vals)):
    param_values['epsilon']=epsilon_vals[i]
    prevs = models_equil.replace_inf_end_equil(param_values)
    S_endem.append(prevs[0]['S'])
    I_W_endem.append(prevs[0]['I_W'])
    I_Z_endem.append(prevs[0]['I_Z'])

fig = plt.figure(figsize=(15,5))

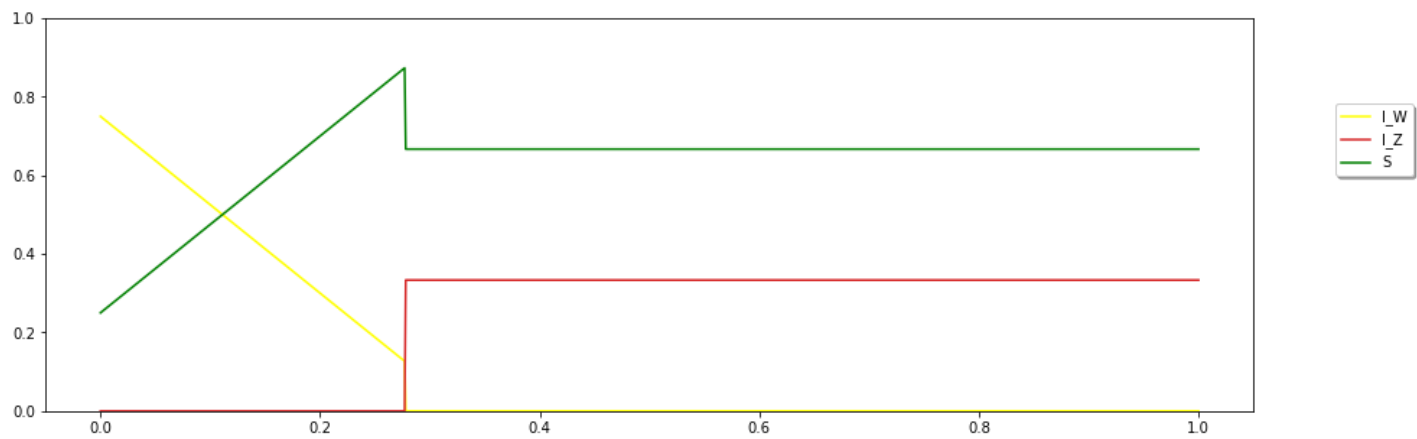
fig.suptitle('Proportion over Time (Days)',
             fontsize=20)

ax = plt.subplot(111)
axes = plt.gca()
axes.set_ylim([0,1])

ax.plot(epsilon_vals,I_W_endem,color='yellow',label='I_W')
ax.plot(epsilon_vals,I_Z_endem,color='tab:red',label='I_Z')
ax.plot(epsilon_vals,S_endem,color='green',label='S')
chartBox = ax.get_position()
ax.legend(loc='upper center', bbox_to_anchor=(1.1, 0.8), shadow=True, ncol=1)
plt.show()
```

```
100%|███████████████████████████████████████████████████████| 1001/  
1001 [01:43<00:00,   9.66it/s]
```

Proportion over Time (Days)



The jump in disease burden occurs when resistant strain endemic equilibrium becomes stable. So what is the first element of `I_Z_endem` to be `>0`.

```
In [4]: I_Z_endem = np.array(I_Z_endem)
I_Z_endem

np.where(I_Z_endem>0)
```

```
Out[4]: (array([ 278,  279,  280,  281,  282,  283,  284,  285,  286,  287,  288,
 289,  290,  291,  292,  293,  294,  295,  296,  297,  298,  299,
300,  301,  302,  303,  304,  305,  306,  307,  308,  309,  310,
311,  312,  313,  314,  315,  316,  317,  318,  319,  320,  321,
322,  323,  324,  325,  326,  327,  328,  329,  330,  331,  332,
333,  334,  335,  336,  337,  338,  339,  340,  341,  342,  343,
344,  345,  346,  347,  348,  349,  350,  351,  352,  353,  354,
355,  356,  357,  358,  359,  360,  361,  362,  363,  364,  365,
366,  367,  368,  369,  370,  371,  372,  373,  374,  375,  376,
377,  378,  379,  380,  381,  382,  383,  384,  385,  386,  387,
388,  389,  390,  391,  392,  393,  394,  395,  396,  397,  398,
399,  400,  401,  402,  403,  404,  405,  406,  407,  408,  409,
410,  411,  412,  413,  414,  415,  416,  417,  418,  419,  420,
421,  422,  423,  424,  425,  426,  427,  428,  429,  430,  431,
432,  433,  434,  435,  436,  437,  438,  439,  440,  441,  442,
443,  444,  445,  446,  447,  448,  449,  450,  451,  452,  453,
454,  455,  456,  457,  458,  459,  460,  461,  462,  463,  464,
465,  466,  467,  468,  469,  470,  471,  472,  473,  474,  475,
476,  477,  478,  479,  480,  481,  482,  483,  484,  485,  486,
487,  488,  489,  490,  491,  492,  493,  494,  495,  496,  497,
498,  499,  500,  501,  502,  503,  504,  505,  506,  507,  508,
509,  510,  511,  512,  513,  514,  515,  516,  517,  518,  519,
520,  521,  522,  523,  524,  525,  526,  527,  528,  529,  530,
531,  532,  533,  534,  535,  536,  537,  538,  539,  540,  541,
542,  543,  544,  545,  546,  547,  548,  549,  550,  551,  552,
553,  554,  555,  556,  557,  558,  559,  560,  561,  562,  563,
564,  565,  566,  567,  568,  569,  570,  571,  572,  573,  574,
575,  576,  577,  578,  579,  580,  581,  582,  583,  584,  585,
586,  587,  588,  589,  590,  591,  592,  593,  594,  595,  596,
597,  598,  599,  600,  601,  602,  603,  604,  605,  606,  607,
608,  609,  610,  611,  612,  613,  614,  615,  616,  617,  618,
619,  620,  621,  622,  623,  624,  625,  626,  627,  628,  629,
630,  631,  632,  633,  634,  635,  636,  637,  638,  639,  640,
641,  642,  643,  644,  645,  646,  647,  648,  649,  650,  651,
652,  653,  654,  655,  656,  657,  658,  659,  660,  661,  662,
663,  664,  665,  666,  667,  668,  669,  670,  671,  672,  673,
674,  675,  676,  677,  678,  679,  680,  681,  682,  683,  684,
685,  686,  687,  688,  689,  690,  691,  692,  693,  694,  695,
696,  697,  698,  699,  700,  701,  702,  703,  704,  705,  706,
707,  708,  709,  710,  711,  712,  713,  714,  715,  716,  717,
```

```

718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728,
729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739,
740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750,
751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761,
762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772,
773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783,
784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794,
795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805,
806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816,
817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827,
828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838,
839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849,
850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860,
861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871,
872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882,
883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893,
894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904,
905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915,
916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926,
927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937,
938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948,
949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959,
960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970,
971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981,
982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992,
993, 994, 995, 996, 997, 998, 999, 1000], dtype=int64),)

```

The values of ϵ , S , I_W and I_Z at that point are equal to:

```
In [5]: display(epsilon_vals[278],S_endem[278],I_W_endem[278],I_Z_endem[278])
```

0.278

0.6666666666666667

0

0.3333333333333333

Just before that point the values of ϵ , S , I_W and I_Z are equal to:

```
In [6]: display(epsilon_vals[277],S_endem[277],I_W_endem[277],I_Z_endem[277])
```

0.277

0.87325

0.12675

0

Meaning the jump is disease burden is:

```
In [7]: S_endem[277]-S_endem[278]
```

```
Out[7]: 0.2065833333333333
```

2 Deriving equations for ϵ at the discontinuous increase of disease burdan and the increase in Disease burden.

Lets look at the disease equalibria, assocaited Jacobian Matrix and Eigen values.

```
In [8]: #Setup Symbols for Sympy
N, S, I_W, I_Z, beta_W, beta_Z, gamma, gamma_T, epsilon = sympy.symbols('N S I_W I_Z beta_W beta_Z gamma gamma_T epsilon')

Eqn_S = -beta_W/N*I_W*S-beta_Z/N*I_Z*S+gamma*(1-epsilon)*I_W+gamma_T*epsilon*I_W+gamma*I_Z
Eqn_I_W = beta_W/N*I_W*S-gamma*(1-epsilon)*I_W-gamma_T*epsilon*I_W+(beta_W/N-beta_Z/N)*I_W*I_Z
Eqn_I_Z = beta_Z/N*I_Z*S-gamma*I_Z+(beta_Z/N-beta_W/N)*I_W*I_Z
```

```
In [9]: ODEs = sympy.Matrix([Eqn_S, Eqn_I_W, Eqn_I_Z])
ODEs
```

```
Out[9]:
```

$$\begin{bmatrix} I_W \epsilon \gamma_T + I_W \gamma (1 - \epsilon) - \frac{I_W S \beta_W}{N} + I_Z \gamma - \frac{I_Z S \beta_Z}{N} \\ I_W I_Z \left(\frac{\beta_W}{N} - \frac{\beta_Z}{N} \right) - I_W \epsilon \gamma_T - I_W \gamma (1 - \epsilon) + \frac{I_W S \beta_W}{N} \\ I_W I_Z \left(-\frac{\beta_W}{N} + \frac{\beta_Z}{N} \right) - I_Z \gamma + \frac{I_Z S \beta_Z}{N} \end{bmatrix}$$

```
In [10]: equilibria = sympy.solve(ODEs, [S, I_W, I_Z])
equilibria
```

```
Out[10]:
```

$$\begin{aligned} & (S, 0, 0), \\ & \left(\frac{I_W \beta_W - I_W \beta_Z + N \gamma}{\beta_Z}, I_W, \frac{-I_W \beta_W^2 + I_W \beta_W \beta_Z - N \beta_W \gamma - N \beta_Z \epsilon \gamma + N \beta_Z \epsilon \gamma_T + N \beta_Z \gamma}{\beta_Z (\beta_W - \beta_Z)} \right), \\ & \left(\frac{N (-\epsilon \gamma + \epsilon \gamma_T + \gamma)}{\beta_W}, I_W, 0 \right), \left(\frac{N \gamma}{\beta_Z}, 0, I_Z \right) \end{aligned}$$

```
In [11]: endem_equil = equilibria[-2:]
endem_equil
```

```
Out[11]:
```

$$\left[\left(\frac{N (-\epsilon \gamma + \epsilon \gamma_T + \gamma)}{\beta_W}, I_W, 0 \right), \left(\frac{N \gamma}{\beta_Z}, 0, I_Z \right) \right]$$

```
In [12]: I_W_equil = N-endem_equil[0][0]
I_Z_equil = N-endem_equil[1][0]
display(I_W_equil, I_Z_equil)
```

$$N - \frac{N (-\epsilon \gamma + \epsilon \gamma_T + \gamma)}{\beta_W}$$

$$N - \frac{N \gamma}{\beta_Z}$$

```
In [13]: J_of_ODEs = ODEs.jacobian(X=[S, I_W, I_Z])
J_of_ODEs
```

```
Out[13]:
```

$$\begin{bmatrix} -\frac{I_W\beta_W}{N} - \frac{I_Z\beta_Z}{N} & \epsilon\gamma_T + \gamma(1-\epsilon) - \frac{S\beta_W}{N} & \gamma - \frac{S\beta_Z}{N} \\ \frac{I_W\beta_W}{N} & I_Z\left(\frac{\beta_W}{N} - \frac{\beta_Z}{N}\right) - \epsilon\gamma_T - \gamma(1-\epsilon) + \frac{S\beta_W}{N} & I_W\left(\frac{\beta_W}{N} - \frac{\beta_Z}{N}\right) \\ \frac{I_Z\beta_Z}{N} & I_Z\left(-\frac{\beta_W}{N} + \frac{\beta_Z}{N}\right) & I_W\left(-\frac{\beta_W}{N} + \frac{\beta_Z}{N}\right) - \gamma + \frac{S\beta_Z}{N} \end{bmatrix}$$

Substitute the 2nd non-disease free equilibrium (non-DFE) into the Jacobian matrix and determine Eigen values.

```
In [14]: non_DFE_2 = {'S':equilibria[3][0], 'I_W':equilibria[3][1], 'I_Z':equilibria[3][2]}

J_of_ODEs_non_DFE_2 = J_of_ODEs.subs(non_DFE_2)
J_of_ODEs_non_DFE_2

eigs_J_non_DFE_2 = J_of_ODEs_non_DFE_2.eigenvals()
# The above code line enters the Eigen values as keys to a dictionary.
# It would be more useful to have them as elements in a vector.
eigs_J_non_DFE_2 = [k for k in eigs_J_non_DFE_2.keys()]
eigs_J_non_DFE_2
```

```
Out[14]:
```

$$\left[\frac{I_Z\beta_W\beta_Z - I_Z\beta_Z^2 + N\beta_W\gamma + N\beta_Z\epsilon\gamma - N\beta_Z\epsilon\gamma_T - N\beta_Z\gamma}{N\beta_Z}, -\frac{I_Z\beta_Z}{N}, 0 \right]$$

2.1 Deriving equation for ϵ at the discontinuous increase of disease burden

Model switches over between equilibrium when the 2nd equation in the set of eigen values = 0 so..

```
In [15]: change_point_v2 = sympy.Eq(0, eigs_J_non_DFE_2[0])
display(change_point_v2)
```

$$0 = \frac{I_Z\beta_W\beta_Z - I_Z\beta_Z^2 + N\beta_W\gamma + N\beta_Z\epsilon\gamma - N\beta_Z\epsilon\gamma_T - N\beta_Z\gamma}{N\beta_Z}$$

```
In [16]: epsilon_at_change_v2 = sympy.solve(change_point_v2, epsilon)
epsilon_at_change_v2
```

```
Out[16]:
```

$$\left[\frac{-I_Z\beta_W\beta_Z + I_Z\beta_Z^2 - N\beta_W\gamma + N\beta_Z\gamma}{N\beta_Z(\gamma - \gamma_T)} \right]$$

```
In [17]: epsilon_at_change_v2=epsilon_at_change_v2[0]
epsilon_at_change_v2 = epsilon_at_change_v2.subs(I_Z, I_Z_equil)
epsilon_at_change_v2
```

```
Out[17]:
```

$$\frac{-N\beta_W\gamma + N\beta_Z\gamma - \beta_W\beta_Z\left(N - \frac{N\gamma}{\beta_Z}\right) + \beta_Z^2\left(N - \frac{N\gamma}{\beta_Z}\right)}{N\beta_Z(\gamma - \gamma_T)}$$

```
In [18]: epsilon_at_change_v2.subs(param_values)
```

```
Out[18]: 0.2777777777777778
```

Lets check this is the correct value

```
In [19]: epsilon_vals[278]
```

Out[19]: 0.278

```
In [20]: param_values['epsilon']=epsilon_at_change_v2.subs(param_values)
         param_values
```

```
Out[20]: {'N': 1,
          'epsilon': 0.2777777777777778,
          'beta_W': 0.04,
          'beta_Z': 0.015,
          'gamma': 0.01,
          'gamma_T': 0.1}
```

Increase in disease burden in replacement model is:

```
In [21]: models_equil.replace_inf_end_equil(param_values)
```

```
Out[21]: [{'S': 0.8750000000000000, 'I_W': 0.1250000000000000, 'I_Z': 0}]
```

This is indeed the correct value of ϵ .

2.2 Deriving Equation for the Increase in Disease burden.

```
In [22]: inc_burd = I_Z_equil - I_W_equil.subs(epsilon,epsilon_at_change_v2)
         inc_burd
```

```
Out[22]:
```

$$-\frac{N\gamma}{\beta_Z} + \frac{N \left(\gamma - \frac{\gamma(-N\beta_W\gamma + N\beta_Z\gamma - \beta_W\beta_Z(N - \frac{N\gamma}{\beta_Z}) + \beta_Z^2(N - \frac{N\gamma}{\beta_Z}))}{N\beta_Z(\gamma - \gamma_T)} + \frac{\gamma_T(-N\beta_W\gamma + N\beta_Z\gamma - \beta_W\beta_Z(N - \frac{N\gamma}{\beta_Z}) + \beta_Z^2(N - \frac{N\gamma}{\beta_Z}))}{N\beta_Z(\gamma - \gamma_T)} \right)}{\beta_W}$$

```
In [23]: inc_burd=inc_burd.simplify()
         inc_burd
```

```
Out[23]:
```

$$N - \frac{N\gamma}{\beta_Z} - \frac{N\beta_Z}{\beta_W} + \frac{N\gamma}{\beta_W}$$

```
In [24]: inc_burd.subs(param_values)
```

```
Out[24]: 0.2083333333333333
```

Lets double check.

```
In [25]: S_endem[277]-S_endem[278]
```

```
Out[25]: 0.2065833333333333
```

3 Checking working with different set of parameters.

```
In [26]: param_values = {'N':1,'epsilon':0.4,'beta_W': 0.04 , 'beta_Z': 0.03,'gamma': 0.01,'gamma_T': 0.1}
         epsilon_vals = np. arange (0, 1.001, 0.001)
         S_endem = []
```


244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254,
255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265,
266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276,
277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287,
288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,
299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309,
310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320,
321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331,
332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342,
343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353,
354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364,
365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375,
376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386,
387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397,
398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408,
409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419,
420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430,
431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441,
442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452,
453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463,
464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474,
475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485,
486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496,
497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507,
508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518,
519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529,
530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540,
541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551,
552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562,
563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573,
574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584,
585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595,
596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606,
607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617,
618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628,
629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639,
640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650,
651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661,
662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672,
673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683,
684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694,
695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705,
706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716,
717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727,
728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738,
739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749,
750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760,
761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771,
772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782,
783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793,
794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804,
805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815,
816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826,
827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837,
838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848,
849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859,
860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870,
871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881,
882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892,
893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903,
904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914,
915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925,
926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936,
937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947,
948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958,
959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969,


```
970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980,
981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991,
992, 993, 994, 995, 996, 997, 998, 999, 1000], dtype=int64),)
```

The values of ϵ , S , I_W and I_Z at that point are equal to:

```
In [28]: display(epsilon_vals[112],S_endem[112],I_W_endem[112],I_Z_endem[112])
```

0.112

0.3333333333333333

0

0.6666666666666667

Just before that point the values of ϵ , S , I_W and I_Z are equal to:

```
In [29]: display(epsilon_vals[111],S_endem[111],I_W_endem[111],I_Z_endem[111])
```

0.111

0.49975

0.50025

0

Meaning the jump is disease burden is:

```
In [30]: S_endem[111]-S_endem[112]
```

Out[30]: 0.16641666666666667

By derived calculation the the jump in disease burden should occur when $\epsilon =$

```
In [31]: epsilon_at_change_v2.subs(param_values)
```

Out[31]: 0.11111111111111111

The increase in disease burden should be:

```
In [32]: inc_burd.subs(param_values)
```

Out[32]: 0.16666666666666667

All seems correct, minor differences can be put down to rounding errors and that epsilon was altered in increments of 0.001 when plotting graphs of stable endemic equilibria.

4 Plots of cost of antimicrobial resistance (reduction in transmission $1 - \frac{\beta_Z}{\beta_W}$) against discontinuous increase in disease burden as a proportion of the population. (Figure 3)

```
In [33]: param_values = {'N':1, 'beta_W': 0.04 , 'beta_Z': 0.015, 'gamma': 0.01, 'gamma_T': 0.1}
dpi =96 # dpi for this screen is accoding to: https://www.infobyip.com/detectmonitordpi.pl
```

Remember the I_W strain's $R_0 = \frac{\beta_Z}{\gamma}$. When $\beta_Z = 0.25\beta_W = 0.01$ and I_W strain's $R_0 = 1$, therefore we will not plot below $\beta_Z = 0.25\beta_W = 0.01$.

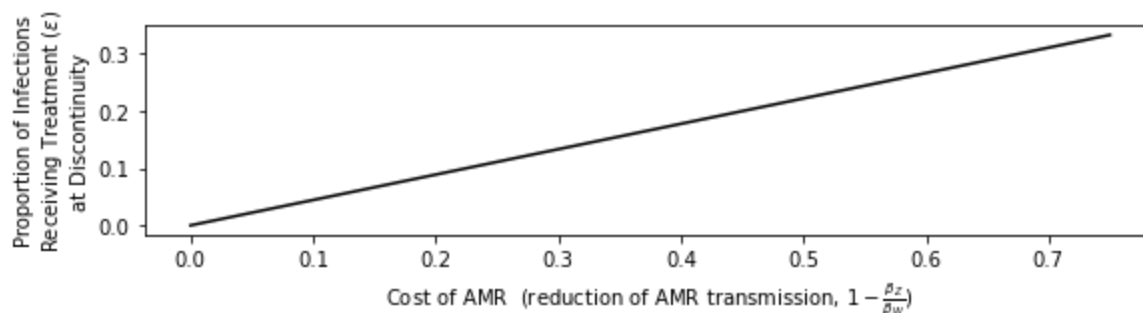
```
In [34]: beta_Z_props = np. arange (int(25), int(101), int(1))
beta_Z_props = np.divide(beta_Z_props,100)
level_of_treatment = []

for prop in beta_Z_props:
    param_values['beta_Z']= param_values['beta_W']*prop
    epsilon = epsilon_at_change_v2.subs(param_values)
    level_of_treatment.append(epsilon)

fig_epsilon = plt.figure(figsize=(870/dpi, 185/dpi))

ax = plt.subplot(111)
axes = plt.gca()

ax.plot(np.subtract(1,beta_Z_props),level_of_treatment,color='black')
ax.set_xlabel(r'Cost of AMR (reduction of AMR transmission,  $1-\frac{\beta_Z}{\beta_W}$ )')
ax.set_ylabel('Proportion of Infections\nReceiving Treatment ( $\epsilon$ )\nat Discontinuity')
plt.show()
```



```
In [35]: fig_epsilon.savefig('Epsilon at disease jump v AMR to AMS transmission.png', bbox_inches='')
```

```
In [36]: param_values = {'N':1, 'beta_W': 0.04 , 'beta_Z': 0.015, 'gamma': 0.01}
```

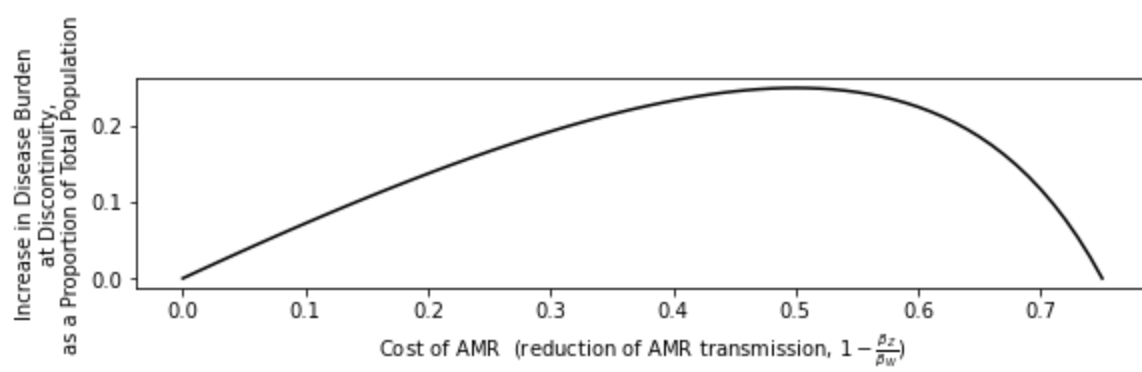
```
In [37]: increases_in_burden = []

for prop in beta_Z_props:
    param_values['beta_Z']= param_values['beta_W']*prop
    increase = inc_burd.subs(param_values)
    increases_in_burden.append(increase)

fig_increase_burden = plt.figure(figsize=(870/dpi, 185/dpi))

ax = plt.subplot(111)
axes = plt.gca()

ax.plot(np.subtract(1,beta_Z_props),increases_in_burden,color='black')
ax.set_xlabel(r'Cost of AMR (reduction of AMR transmission,  $1-\frac{\beta_Z}{\beta_W}$ )')
ax.set_ylabel('Increase in Disease Burden\nat Discontinuity,\nas a Proportion of Total Pop')
plt.show()
```



```
In [38]: fig_increase_burden.savefig('Increase in disease burden v AMR to AMS transmission.png', bk
```

```
In [39]: np.where(beta_Z_props==0.9)[0][0]
```

```
Out[39]: 65
```

```
In [40]: display(increases_in_burden[np.where(beta_Z_props==0.9)[0][0]],level_of_treatment[np.where
```

```
0.0722222222222221
```

```
0.0444444444444444
```

```
In [41]: display(increases_in_burden[np.where(beta_Z_props==0.5)[0][0]],level_of_treatment[np.where
```

```
0.25
```

```
0.222222222222222
```

```
In [ ]:
```