**APPENDIX TO NUDGES IN EXERCISE COMMITMENT CONTRACTS: A RANDOMIZED TRIAL**

Jay Bhattacharya

Alan M. Garber

Jeremy D. Goldhaber-Fiebert

July 2015

**Appendix 1: Supplementary Results**

**A1.0 Variation in cut-offs for the first stage of the treatment on the treated analysis**

In this section, we explore how changing the cut-off points (*k*1 and *k*2) that define short, medium, and long exercise spells change the estimated treatment effect of contract duration on the number of successful weeks of exercise, and on subsequent take-up of exercise commitment contracts. Like in our main analysis (see Figure 4), across a range of other possible *k*1 and *k*2 cut-offs, users in the longest duration contracts tend to exercise the longest, while users in the shortest and middle categories exercise less (Table A1). Similarly, in Tables A2 and A3, we test the sensitivity of changing *k*1 and *k*2 on the probability of signing a subsequent exercise commitment contract within 30 and 90 days of finishing the initial contract (Figures 5 and 6). In neither case is our result sensitive to varying *k*1 and *k*2.

**Table A1. Effects of Exogenously Increased Exercise on Within Contract Exercise Successful Weeks: Variation in the Cutoffs Used in the First-Stage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Duration Category****<=k1** | **Duration Category****>k1 and <=k2** | **Duration Category****>k2** |
| **Low****cut-off****(<=k1)** | **Middle****cut-off****(<=k2)** | **Success****Weeks** | **95%** **CI** | **Success****Weeks** | **95%** **CI** | **Success****Weeks** | **95%** **CI** |
| 9 | 13 | 3.3 | 2.1 | 4.5 | 1.7 | -1.0 | 4.1 | 5.7 | 4.5 | 7.0 |
| 9 | 14 | 3.3 | 2.0 | 4.6 | 1.7 | -0.9 | 4.5 | 5.8 | 4.2 | 7.3 |
| 9 | 15 | 3.3 | 2.2 | 4.7 | 1.7 | -0.9 | 4.1 | 5.8 | 4.5 | 7.3 |
| 9 | 16 | 3.3 | 2.2 | 4.6 | 1.8 | -0.8 | 4.1 | 5.8 | 4.4 | 7.3 |
| 9 | 17 | 3.3 | 2.2 | 4.4 | 2.0 | 0.0 | 4.2 | 5.9 | 4.4 | 7.3 |
| 9 | 18 | 3.3 | 2.1 | 4.4 | 2.0 | -0.3 | 4.0 | 6.0 | 4.4 | 7.5 |
| 9 | 19 | 3.3 | 2.3 | 4.5 | 2.0 | -0.2 | 4.0 | 6.0 | 4.4 | 7.6 |
| 9 | 20 | 3.3 | 2.3 | 4.5 | 2.0 | -0.3 | 3.9 | 6.0 | 4.5 | 7.7 |
| 10 | 13 | 3.3 | 2.0 | 4.5 | 1.6 | -1.0 | 4.5 | 5.7 | 4.3 | 6.9 |
| 10 | 14 | 3.3 | 2.1 | 4.6 | 1.6 | -1.0 | 4.4 | 5.8 | 4.4 | 7.3 |
| 10 | 15 | 3.3 | 2.2 | 4.6 | 1.7 | -0.8 | 4.2 | 5.8 | 4.4 | 7.2 |
| 10 | 16 | 3.3 | 2.2 | 4.5 | 1.8 | -0.8 | 4.2 | 5.9 | 4.4 | 7.5 |
| 10 | 17 | 3.3 | 2.2 | 4.4 | 2.0 | -0.2 | 4.1 | 6.0 | 4.4 | 7.5 |
| 10 | 18 | 3.3 | 2.0 | 4.4 | 2.0 | -0.1 | 4.4 | 6.0 | 4.5 | 7.6 |
| 10 | 19 | 3.3 | 2.2 | 4.6 | 2.0 | -0.4 | 4.2 | 6.0 | 4.7 | 7.9 |
| 10 | 20 | 3.3 | 2.2 | 4.5 | 2.0 | -0.5 | 4.0 | 6.0 | 4.6 | 7.8 |
| 11 | 13 | 3.4 | 2.3 | 4.7 | 0.9 | -2.9 | 4.3 | 6.0 | 4.5 | 7.8 |
| 11 | 14 | 3.4 | 2.3 | 4.5 | 1.0 | -2.3 | 4.1 | 6.1 | 4.6 | 7.8 |
| 11 | 15 | 3.4 | 2.2 | 4.5 | 1.0 | -1.9 | 4.5 | 6.1 | 4.4 | 7.6 |
| 11 | 16 | 3.4 | 2.4 | 4.5 | 1.2 | -1.9 | 4.1 | 6.2 | 4.5 | 8.0 |
| 11 | 17 | 3.4 | 2.3 | 4.4 | 1.4 | -1.1 | 4.2 | 6.3 | 4.5 | 8.1 |
| 11 | 18 | 3.4 | 2.3 | 4.6 | 1.4 | -1.3 | 4.4 | 6.3 | 4.6 | 8.4 |
| 11 | 19 | 3.4 | 2.4 | 4.6 | 1.4 | -1.4 | 4.0 | 6.4 | 4.7 | 8.6 |
| 11 | 20 | 3.4 | 2.3 | 4.4 | 1.5 | -1.0 | 4.2 | 6.4 | 4.7 | 8.3 |
| 12 | 13 | 3.4 | 2.2 | 4.5 | 0.9 | -2.8 | 4.1 | 6.0 | 4.5 | 7.4 |
| 12 | 14 | 3.4 | 2.3 | 4.5 | 0.9 | -2.3 | 4.2 | 6.1 | 4.4 | 7.7 |
| 12 | 15 | 3.4 | 2.4 | 4.6 | 1.0 | -2.5 | 4.2 | 6.1 | 4.6 | 8.0 |
| 12 | 16 | 3.4 | 2.2 | 4.5 | 1.1 | -2.0 | 4.4 | 6.2 | 4.6 | 8.1 |
| 12 | 17 | 3.3 | 2.3 | 4.4 | 1.4 | -1.5 | 4.2 | 6.3 | 4.6 | 8.4 |
| 12 | 18 | 3.3 | 2.3 | 4.4 | 1.4 | -1.6 | 4.2 | 6.3 | 4.6 | 8.3 |
| 12 | 19 | 3.3 | 2.4 | 4.4 | 1.4 | -1.3 | 4.1 | 6.4 | 4.5 | 8.2 |
| 12 | 20 | 3.3 | 2.3 | 4.4 | 1.4 | -1.5 | 4.2 | 6.4 | 4.7 | 8.7 |

**Table A2. Effects of Exogenously Increased Exercise on Subsequent Demand for Exercise Commitment Contracts within 30 Days after the End of the First Contract**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Duration****Category****<=k1** | **Duration****Category****>k1 and <=k2** | **Duration****Category****>k2** |
| **Low****cut-off****(<=k1)** | **Middle****cut-off****(<=k2)** | **2nd****Contract****Signed** | **95%** **CI** | **2nd****Contract****Signed** | **95%** **CI** | **2nd****Contract****Signed** | **95%** **CI** |
| 9 | 13 | 3.6% | 0.3% | 53.1% | 3.3% | 0.5% | 80.4% | 7.1% | 0.7% | 87.6% |
| 9 | 14 | 3.6% | 0.3% | 42.2% | 3.6% | 0.9% | 92.3% | 7.1% | 0.8% | 75.7% |
| 9 | 15 | 3.6% | 0.4% | 45.8% | 3.7% | 0.7% | 83.6% | 7.1% | 1.0% | 84.0% |
| 9 | 16 | 3.6% | 0.3% | 36.1% | 4.0% | 0.8% | 87.1% | 7.1% | 0.8% | 70.3% |
| 9 | 17 | 3.6% | 0.2% | 32.3% | 1.5% | 0.0% | 9.5% | 8.7% | 0.5% | 77.8% |
| 9 | 18 | 3.6% | 0.2% | 35.5% | 1.6% | 0.0% | 10.6% | 8.8% | 0.7% | 83.6% |
| 9 | 19 | 3.6% | 0.2% | 49.1% | 1.6% | 0.0% | 11.8% | 8.9% | 0.7% | 91.6% |
| 9 | 20 | 3.6% | 0.3% | 39.1% | 1.6% | 0.0% | 11.8% | 8.9% | 0.6% | 77.3% |
| 10 | 13 | 3.6% | 0.4% | 48.3% | 2.9% | 0.4% | 74.7% | 7.2% | 0.7% | 78.8% |
| 10 | 14 | 3.6% | 0.3% | 53.9% | 3.2% | 0.4% | 79.4% | 7.2% | 0.8% | 88.5% |
| 10 | 15 | 3.6% | 0.2% | 40.2% | 3.3% | 0.6% | 82.0% | 7.2% | 0.6% | 76.0% |
| 10 | 16 | 3.6% | 0.3% | 39.2% | 3.6% | 0.5% | 90.5% | 7.1% | 0.8% | 83.8% |
| 10 | 17 | 3.5% | 0.2% | 36.4% | 1.3% | 0.0% | 8.4% | 9.3% | 0.6% | 83.9% |
| 10 | 18 | 3.5% | 0.2% | 37.8% | 1.3% | 0.0% | 8.9% | 9.4% | 0.7% | 90.6% |
| 10 | 19 | 3.6% | 0.2% | 46.4% | 1.4% | 0.0% | 10.2% | 9.5% | 0.7% | 97.1% |
| 10 | 20 | 3.6% | 0.1% | 35.5% | 1.4% | 0.0% | 10.2% | 9.5% | 0.7% | 94.4% |
| 11 | 13 | 3.4% | 0.2% | 64.3% | 1.6% | 0.0% | 33.1% | 7.8% | 0.8% | 96.2% |
| 11 | 14 | 3.4% | 0.1% | 43.9% | 1.9% | 0.0% | 61.6% | 7.7% | 1.0% | 94.8% |
| 11 | 15 | 3.4% | 0.2% | 52.6% | 2.1% | 0.0% | 75.1% | 7.7% | 0.6% | 95.3% |
| 11 | 16 | 3.4% | 0.2% | 37.1% | 2.5% | 0.1% | 84.8% | 7.4% | 0.7% | 92.0% |
| 11 | 17 | 3.4% | 0.1% | 49.8% | 0.6% | 0.0% | 6.0% | 13.3% | 0.8% | 98.4% |
| 11 | 18 | 3.4% | 0.0% | 53.8% | 0.6% | 0.0% | 6.1% | 13.4% | 0.6% | 97.0% |
| 11 | 19 | 3.4% | 0.0% | 57.4% | 0.6% | 0.0% | 6.0% | 13.5% | 0.8% | 98.6% |
| 11 | 20 | 3.4% | 0.1% | 62.1% | 0.6% | 0.0% | 6.1% | 13.5% | 0.6% | 98.6% |
| 12 | 13 | 3.4% | 0.1% | 47.5% | 1.7% | 0.0% | 54.5% | 7.6% | 0.7% | 96.2% |
| 12 | 14 | 3.4% | 0.3% | 69.5% | 2.1% | 0.0% | 75.3% | 7.5% | 0.8% | 97.8% |
| 12 | 15 | 3.4% | 0.2% | 50.4% | 2.2% | 0.1% | 83.9% | 7.4% | 0.8% | 96.5% |
| 12 | 16 | 3.4% | 0.2% | 40.1% | 2.6% | 0.2% | 83.4% | 7.2% | 0.7% | 91.5% |
| 12 | 17 | 3.4% | 0.1% | 60.8% | 0.6% | 0.0% | 6.6% | 13.3% | 0.6% | 98.6% |
| 12 | 18 | 3.4% | 0.1% | 60.7% | 0.6% | 0.0% | 5.4% | 13.4% | 0.7% | 97.5% |
| 12 | 19 | 3.4% | 0.1% | 59.3% | 0.6% | 0.0% | 6.4% | 13.5% | 0.7% | 99.5% |
| 12 | 20 | 3.4% | 0.0% | 51.0% | 0.6% | 0.0% | 7.7% | 13.5% | 0.6% | 98.0% |

**Table A3. Effects of Exogenously Increased Exercise on Subsequent Demand for Exercise Commitment Contracts within 90 Days after the End of the First Contract**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Duration****Category****<=k1** | **Duration****Category****>k1 and <=k2** | **Duration****Category****>k2** |
| **Low****cut-off****(<=k1)** | **Middle****cut-off****(<=k2)** | **2nd****Contract****Signed** | **95%****CI** | **2nd****Contract****Signed** | **95%****CI** | **2nd****Contract****Signed** | **95%****CI** |
| 9 | 13 | 5.4% | 0.7% | 39.7% | 3.9% | 0.5% | 65.7% | 8.5% | 1.3% | 80.8% |
| 9 | 14 | 5.4% | 0.7% | 57.1% | 3.4% | 0.4% | 77.7% | 8.8% | 1.4% | 85.1% |
| 9 | 15 | 5.4% | 0.6% | 52.5% | 3.2% | 0.2% | 49.3% | 9.0% | 1.5% | 83.4% |
| 9 | 16 | 5.4% | 0.8% | 43.8% | 3.7% | 0.6% | 70.9% | 9.0% | 1.6% | 88.6% |
| 9 | 17 | 5.3% | 0.4% | 44.9% | 1.7% | 0.0% | 7.2% | 10.6% | 1.2% | 86.3% |
| 9 | 18 | 5.3% | 0.4% | 44.0% | 1.6% | 0.0% | 6.9% | 10.8% | 1.2% | 87.9% |
| 9 | 19 | 5.3% | 0.3% | 47.2% | 1.7% | 0.0% | 7.0% | 10.9% | 1.0% | 86.5% |
| 9 | 20 | 5.3% | 0.3% | 38.1% | 1.7% | 0.0% | 7.1% | 11.0% | 1.1% | 83.1% |
| 10 | 13 | 5.3% | 0.6% | 44.5% | 3.3% | 0.2% | 70.3% | 8.6% | 1.4% | 72.9% |
| 10 | 14 | 5.3% | 0.6% | 65.3% | 2.9% | 0.1% | 46.1% | 9.0% | 1.3% | 91.5% |
| 10 | 15 | 5.3% | 0.6% | 52.8% | 2.7% | 0.1% | 35.3% | 9.2% | 1.6% | 87.4% |
| 10 | 16 | 5.3% | 0.7% | 44.3% | 3.2% | 0.2% | 51.4% | 9.2% | 1.5% | 79.1% |
| 10 | 17 | 5.3% | 0.3% | 42.8% | 1.4% | 0.0% | 5.8% | 11.4% | 0.9% | 86.7% |
| 10 | 18 | 5.3% | 0.3% | 42.3% | 1.4% | 0.0% | 5.9% | 11.6% | 1.1% | 92.5% |
| 10 | 19 | 5.3% | 0.3% | 50.8% | 1.4% | 0.0% | 6.2% | 11.8% | 1.1% | 93.1% |
| 10 | 20 | 5.3% | 0.3% | 46.7% | 1.4% | 0.0% | 5.9% | 11.8% | 1.0% | 87.7% |
| 11 | 13 | 5.1% | 0.5% | 59.0% | 1.8% | 0.0% | 22.0% | 9.4% | 1.6% | 91.1% |
| 11 | 14 | 5.1% | 0.4% | 70.8% | 1.5% | 0.0% | 19.5% | 10.3% | 1.5% | 98.0% |
| 11 | 15 | 5.1% | 0.5% | 68.6% | 1.4% | 0.0% | 16.2% | 10.8% | 1.5% | 96.4% |
| 11 | 16 | 5.1% | 0.4% | 57.2% | 1.9% | 0.0% | 14.6% | 10.5% | 1.5% | 96.4% |
| 11 | 17 | 5.0% | 0.1% | 65.9% | 0.6% | 0.0% | 6.4% | 16.7% | 1.1% | 97.2% |
| 11 | 18 | 5.0% | 0.1% | 66.8% | 0.6% | 0.0% | 5.5% | 17.6% | 1.2% | 98.8% |
| 11 | 19 | 5.0% | 0.1% | 62.8% | 0.6% | 0.0% | 5.4% | 17.8% | 1.1% | 97.2% |
| 11 | 20 | 5.0% | 0.1% | 68.9% | 0.6% | 0.0% | 5.7% | 17.8% | 1.1% | 98.6% |
| 12 | 13 | 5.0% | 0.5% | 60.2% | 1.9% | 0.0% | 21.1% | 9.3% | 1.6% | 92.9% |
| 12 | 14 | 5.0% | 0.3% | 79.6% | 1.6% | 0.0% | 15.9% | 10.2% | 1.1% | 98.4% |
| 12 | 15 | 5.0% | 0.4% | 63.1% | 1.5% | 0.0% | 14.3% | 10.7% | 1.6% | 97.1% |
| 12 | 16 | 5.0% | 0.4% | 52.8% | 1.9% | 0.0% | 14.9% | 10.3% | 1.3% | 96.2% |
| 12 | 17 | 5.0% | 0.1% | 72.1% | 0.6% | 0.0% | 5.9% | 16.9% | 1.1% | 98.3% |
| 12 | 18 | 5.0% | 0.1% | 72.7% | 0.6% | 0.0% | 6.1% | 17.8% | 1.1% | 98.8% |
| 12 | 19 | 5.0% | 0.1% | 68.7% | 0.6% | 0.0% | 7.1% | 18.0% | 1.2% | 98.3% |
| 12 | 20 | 5.0% | 0.1% | 66.8% | 0.6% | 0.0% | 5.1% | 17.9% | 1.1% | 98.8% |

**A1.1 Optimal exercise varies with exercise-specific capital depreciation (**$γ$**) and myopia (**$δ$**)**

In this section, we plot how optimal exercise demand changes as $γ$ and $δ$ vary, holding fixed the other utility parameters of our hyperbolic discounting model at fixed values. While the optimal level changes with these other parameters ($α$, etc.), the pattern is typically like the one shown in Figure A1. In this figure, the top left panel corresponds to exercise demand in period 1, the top right to period 2, bottom left to period 3, and bottom right to period 4. The demand is color coded, from dark blue (no exercise) to orange (highest level of exercise). As the degree of myopia decreases ($δ$ increases), optimal exercise increases in all periods. Likewise, as the rate of depreciation of exercise-related capital decreases ($γ$ increases), optimal exercise increases in all periods. Optimal exercise for high levels of $δ$ (low myopia), increases more quickly with an increase in $γ$. Finally, zero optimal exercise occupies a large fraction of the $δ$-$γ$ parameter space in later periods. All these results are intuitively plausible for our hyperbolic discounting model.

**Figure A1. Optimal Exercise in Each of the Four Periods for an Typical Example Set of Utility Function Parameters as Delta and Gamma are Varied**

S1

**Appendix 2: Simulation Methods**

**A2.0 Overview**

The following sections of text provide the code used to conduct the analyses in the paper. We have annotated them using commenting styles particular to Mathematica code and Stata code to facilitate review, understanding, and use. Code sections are shown in blue text in Courier font and smaller point size. These sections should be usable in the respective programs.

**A2.1 Finding Optimal Exercise Paths (Mathematica)**

(\* Annotated Mathematica code to compute optimal exercise vectors, e\*, corresponding period utility, u1...u4, and total utility vectors, U1...U4, for a 4 period hyperbolic discounting model with exercise commitment. \*)

(\* Setup the workspace and clear out the memory \*)

$HistoryLength = 500

ClearMemory := Module[{},

 Unprotect[In, Out];

 Clear[In, Out];

 Protect[In, Out];

 ClearSystemCache[];

 ];

(\* Define our habit equations, k0...k3, our period utility, and total utility equations. g, d, and b correspond to gamma, delta, and beta in the text. a1...a9 correspond to the alphas of our quadratic form: a1 (alpha\_ee); a2 (alpha\_cc); a3 (alpha\_kk); a4 (alpha\_ec); a5 (alpha\_ek); a6 (alpha\_ck); a7 (alpha\_e); a8 (alpha\_c); and a9 (alpha\_k). \*)

k1 = g\*(k0+e1)

k2 = g\*(k1+e2)

k3 = g\*(k2+e3)

k[et\_,kt\_] := g\*(kt+et)

ut[et\_,kt\_]:=a1\*(et)^2 + a2 (1-p\*et)^2 + a3\*(kt)^2 + a4\*et\*(1-p\*et) + a5\*et\*kt + a6\*(1-p\*et)\*kt + a7\*et + a8\*(1-p\*et) + a9\*kt

U4[e4\_]=ut[e4,k3]

U3[e3\_,e4\_]=ut[e3,k2] + d\*b\*ut[e4,k3]

U2[e2\_,e3\_,e4\_]=ut[e2,k1] + d\*b\*ut[e3,k2] + d\*(b^2)\*ut[e4,k3]

U1[e1\_,e2\_,e3\_,e4\_]=ut[e1,k0] + d\*b\*ut[e2,k1] + d\*(b^2)\*ut[e3,k2] + d\*(b^3)\*ut[e4,k3]

(\* Take the derivative of the period 4’s total utility with respect to exercise. Then solve for its maximum -- s4i for an interior solution; s40 for a lower corner solution; s4p for an upper corner solution. \*)

D[U4[e4], e4]

s4i = Solve[% == 0, e4]

s4i = Simplify[s4i]

s40 = {e4->0}

s4p = {e4->p}

(\* Take the derivative of the period 3’s total utility with respect to exercise . There are 3 cases to consider for the period 3 interior optimum: depending on whether e4 is at a lower corner (s3iu40), upper corner (s3iu4p), or interior max (s3iu4i). Likewise, the period 3 optimum could be a corner (s30 and s3p).\*)

U3u4i[e3,e4] = ReplaceRepeated[U3[e3,e4], s4i]

U3u40[e3,e4] = ReplaceRepeated[U3[e3,e4], s40]

U3u4p[e3,e4] = ReplaceRepeated[U3[e3,e4], s4p]

D[U3u4i[e3,e4],e3]

s3iu4i = Solve[% == 0, e3]

s3iu4i = Simplify[s3iu4i]

D[U3u40[e3,e4],e3]

s3iu40 = Solve[% == 0, e3]

s3iu40 = Simplify[s3iu40]

D[U3u4p[e3,e4],e3]

s3iu4p = Solve[% == 0, e3]

s3iu4p = Simplify[s3iu4p]

s30 = {e3->0}

s3p = {e3->p}

(\* Repeat this step for the second period’s total utility. There are of course more possible cases since corners or interiors can happen at either the third or fourth periods and the optimum at the second period will be conditional on those if interior or potentially a corner at 0 or p itself . The nomenclature keeps track of the path of subsequent optimal exercise, and in particular whether it is a corner or interior. \*)

U2u3iu4i[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4i]

U2u3iu40[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s40]

U2u3iu4p[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4p]

U2u3iu4i[e2,e3,e4] = ReplaceRepeated[U2u3iu4i[e2,e3,e4], s3iu4i]

U2u3iu40[e2,e3,e4] = ReplaceRepeated[U2u3iu40[e2,e3,e4], s3iu40]

U2u3iu4p[e2,e3,e4] = ReplaceRepeated[U2u3iu4p[e2,e3,e4], s3iu4p]

U2u30u4i[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4i]

U2u30u40[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s40]

U2u30u4p[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4p]

U2u30u4i[e2,e3,e4] = ReplaceRepeated[U2u30u4i[e2,e3,e4], s30]

U2u30u40[e2,e3,e4] = ReplaceRepeated[U2u30u40[e2,e3,e4], s30]

U2u30u4p[e2,e3,e4] = ReplaceRepeated[U2u30u4p[e2,e3,e4], s30]

U2u3pu4i[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4i]

U2u3pu40[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s40]

U2u3pu4p[e2,e3,e4] = ReplaceRepeated[U2[e2,e3,e4], s4p]

U2u3pu4i[e2,e3,e4] = ReplaceRepeated[U2u3pu4i[e2,e3,e4], s3p]

U2u3pu40[e2,e3,e4] = ReplaceRepeated[U2u3pu40[e2,e3,e4], s3p]

U2u3pu4p[e2,e3,e4] = ReplaceRepeated[U2u3pu4p[e2,e3,e4], s3p]

D[U2u3iu4i[e2,e3,e4],e2]

s2iu3iu4i = Solve[% == 0, e2]

s2iu3iu4i = Simplify[s2iu3iu4i]

D[U2u3iu40[e2,e3,e4],e2]

s2iu3iu40 = Solve[% == 0, e2]

s2iu3iu40 = Simplify[s2iu3iu40]

D[U2u3iu4p[e2,e3,e4],e2]

s2iu3iu4p = Solve[% == 0, e2]

s2iu3iu4p = Simplify[s2iu3iu4p]

D[U2u30u4i[e2,e3,e4],e2]

s2iu30u4i = Solve[% == 0, e2]

s2iu30u4i = Simplify[s2iu30u4i]

D[U2u30u40[e2,e3,e4],e2]

s2iu30u40 = Solve[% == 0, e2]

s2iu30u40 = Simplify[s2iu30u40]

D[U2u30u4p[e2,e3,e4],e2]

s2iu30u4p = Solve[% == 0, e2]

s2iu30u4p = Simplify[s2iu30u4p]

D[U2u3pu4i[e2,e3,e4],e2]

s2iu3pu4i = Solve[% == 0, e2]

s2iu3pu4i = Simplify[s2iu3pu4i]

D[U2u3pu40[e2,e3,e4],e2]

s2iu3pu40 = Solve[% == 0, e2]

s2iu3pu40 = Simplify[s2iu3pu40]

D[U2u3pu4p[e2,e3,e4],e2]

s2iu3pu4p = Solve[% == 0, e2]

s2iu3pu4p = Simplify[s2iu3pu4p]

s20 = {e2->0}

s2p = {e2->p}

(\* While ideally, one would extend the above approach to period 1 and be done, this is not computationally feasible as the symbolic manipulation to find the maximum exceeds 128 GB of RAM. Rather than recruit a supercomputer, we adopt a different approach.

We set up the first period total utility as a function that includes the equations for the optimal levels of exercise in the subsequent periods for all of the 27 (=3^3) cases of interior and corner solutions at each of those subsequent periods. The nomenclature for utility levels is consistent with the code above. \*)

U1u2iu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2iu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2iu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2iu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu4i[e1,e2,e3,e4], s3iu4i]

U1u2iu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu40[e1,e2,e3,e4], s3iu40]

U1u2iu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu4p[e1,e2,e3,e4], s3iu4p]

U1u2iu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu4i[e1,e2,e3,e4], s2iu3iu4i]

U1u2iu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu40[e1,e2,e3,e4], s2iu3iu40]

U1u2iu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3iu4p[e1,e2,e3,e4], s2iu3iu4p]

U1u2iu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2iu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2iu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2iu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u4i[e1,e2,e3,e4], s30]

U1u2iu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u40[e1,e2,e3,e4], s30]

U1u2iu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u4p[e1,e2,e3,e4], s30]

U1u2iu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u4i[e1,e2,e3,e4], s2iu30u4i]

U1u2iu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u40[e1,e2,e3,e4], s2iu30u40]

U1u2iu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu30u4p[e1,e2,e3,e4], s2iu30u4p]

U1u2iu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2iu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2iu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2iu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu4i[e1,e2,e3,e4], s3p]

U1u2iu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu40[e1,e2,e3,e4], s3p]

U1u2iu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu4p[e1,e2,e3,e4], s3p]

U1u2iu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu4i[e1,e2,e3,e4], s2iu3pu4i]

U1u2iu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu40[e1,e2,e3,e4], s2iu3pu40]

U1u2iu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2iu3pu4p[e1,e2,e3,e4], s2iu3pu4p]

U1u20u3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u20u3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u20u3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u20u3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu4i[e1,e2,e3,e4], s3iu4i]

U1u20u3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu40[e1,e2,e3,e4], s3iu40]

U1u20u3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu4p[e1,e2,e3,e4], s3iu4p]

U1u20u3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu4i[e1,e2,e3,e4], s20]

U1u20u3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu40[e1,e2,e3,e4], s20]

U1u20u3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3iu4p[e1,e2,e3,e4], s20]

U1u20u30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u20u30u40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u20u30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u20u30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u4i[e1,e2,e3,e4], s30]

U1u20u30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u40[e1,e2,e3,e4], s30]

U1u20u30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u4p[e1,e2,e3,e4], s30]

U1u20u30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u4i[e1,e2,e3,e4], s20]

U1u20u30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u40[e1,e2,e3,e4], s20]

U1u20u30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u30u4p[e1,e2,e3,e4], s20]

U1u20u3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u20u3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u20u3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u20u3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu4i[e1,e2,e3,e4], s3p]

U1u20u3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu40[e1,e2,e3,e4], s3p]

U1u20u3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu4p[e1,e2,e3,e4], s3p]

U1u20u3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu4i[e1,e2,e3,e4], s20]

U1u20u3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu40[e1,e2,e3,e4], s20]

U1u20u3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u20u3pu4p[e1,e2,e3,e4], s20]

U1u2pu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2pu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2pu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2pu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu4i[e1,e2,e3,e4], s3iu4i]

U1u2pu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu40[e1,e2,e3,e4], s3iu40]

U1u2pu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu4p[e1,e2,e3,e4], s3iu4p]

U1u2pu3iu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu4i[e1,e2,e3,e4], s2p]

U1u2pu3iu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu40[e1,e2,e3,e4], s2p]

U1u2pu3iu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3iu4p[e1,e2,e3,e4], s2p]

U1u2pu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2pu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2pu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2pu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u4i[e1,e2,e3,e4], s30]

U1u2pu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u40[e1,e2,e3,e4], s30]

U1u2pu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u4p[e1,e2,e3,e4], s30]

U1u2pu30u4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u4i[e1,e2,e3,e4], s2p]

U1u2pu30u40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u40[e1,e2,e3,e4], s2p]

U1u2pu30u4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu30u4p[e1,e2,e3,e4], s2p]

U1u2pu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4i]

U1u2pu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s40]

U1u2pu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1[e1,e2,e3,e4], s4p]

U1u2pu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu4i[e1,e2,e3,e4], s3p]

U1u2pu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu40[e1,e2,e3,e4], s3p]

U1u2pu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu4p[e1,e2,e3,e4], s3p]

U1u2pu3pu4i[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu4i[e1,e2,e3,e4], s2p]

U1u2pu3pu40[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu40[e1,e2,e3,e4], s2p]

U1u2pu3pu4p[e1,e2,e3,e4] = ReplaceRepeated[U1u2pu3pu4p[e1,e2,e3,e4], s2p]

(\* To find the first period optimum, differentiate the first period total utility for all the cases \*)

DU1u2iu3iu4i[e1,e2,e3,e4] = D[U1u2iu3iu4i[e1,e2,e3,e4] , e1]

DU1u2iu3iu40[e1,e2,e3,e4] = D[U1u2iu3iu40[e1,e2,e3,e4] , e1]

DU1u2iu3iu4p[e1,e2,e3,e4] = D[U1u2iu3iu4p[e1,e2,e3,e4] , e1]

DU1u2iu30u4i[e1,e2,e3,e4] = D[U1u2iu30u4i[e1,e2,e3,e4] , e1]

DU1u2iu30u40[e1,e2,e3,e4] = D[U1u2iu30u40[e1,e2,e3,e4] , e1]

DU1u2iu30u4p[e1,e2,e3,e4] = D[U1u2iu30u4p[e1,e2,e3,e4] , e1]

DU1u2iu3pu4i[e1,e2,e3,e4] = D[U1u2iu3pu4i[e1,e2,e3,e4] , e1]

DU1u2iu3pu40[e1,e2,e3,e4] = D[U1u2iu3pu40[e1,e2,e3,e4] , e1]

DU1u2iu3pu4p[e1,e2,e3,e4] = D[U1u2iu3pu4p[e1,e2,e3,e4] , e1]

DU1u20u3iu4i[e1,e2,e3,e4] = D[U1u20u3iu4i[e1,e2,e3,e4] , e1]

DU1u20u3iu40[e1,e2,e3,e4] = D[U1u20u3iu40[e1,e2,e3,e4] , e1]

DU1u20u3iu4p[e1,e2,e3,e4] = D[U1u20u3iu4p[e1,e2,e3,e4] , e1]

DU1u20u30u4i[e1,e2,e3,e4] = D[U1u20u30u4i[e1,e2,e3,e4] , e1]

DU1u20u30u40[e1,e2,e3,e4] = D[U1u20u30u40[e1,e2,e3,e4] , e1]

DU1u20u30u4p[e1,e2,e3,e4] = D[U1u20u30u4p[e1,e2,e3,e4] , e1]

DU1u20u3pu4i[e1,e2,e3,e4] = D[U1u20u3pu4i[e1,e2,e3,e4] , e1]

DU1u20u3pu40[e1,e2,e3,e4] = D[U1u20u3pu40[e1,e2,e3,e4] , e1]

DU1u20u3pu4p[e1,e2,e3,e4] = D[U1u20u3pu4p[e1,e2,e3,e4] , e1]

DU1u2pu3iu4i[e1,e2,e3,e4] = D[U1u2pu3iu4i[e1,e2,e3,e4] , e1]

DU1u2pu3iu40[e1,e2,e3,e4] = D[U1u2pu3iu40[e1,e2,e3,e4] , e1]

DU1u2pu3iu4p[e1,e2,e3,e4] = D[U1u2pu3iu4p[e1,e2,e3,e4] , e1]

DU1u2pu30u4i[e1,e2,e3,e4] = D[U1u2pu30u4i[e1,e2,e3,e4] , e1]

DU1u2pu30u40[e1,e2,e3,e4] = D[U1u2pu30u40[e1,e2,e3,e4] , e1]

DU1u2pu30u4p[e1,e2,e3,e4] = D[U1u2pu30u4p[e1,e2,e3,e4] , e1]

DU1u2pu3pu4i[e1,e2,e3,e4] = D[U1u2pu3pu4i[e1,e2,e3,e4] , e1]

DU1u2pu3pu40[e1,e2,e3,e4] = D[U1u2pu3pu40[e1,e2,e3,e4] , e1]

DU1u2pu3pu4p[e1,e2,e3,e4] = D[U1u2pu3pu4p[e1,e2,e3,e4] , e1]

(\* Generate a list (lstDU1) to be able to refer to all of these cases by name. \*)

lstDU1 = {

"DU1u2iu3iu4i[e1,e2,e3,e4]",

"DU1u2iu3iu40[e1,e2,e3,e4]",

"DU1u2iu3iu4p[e1,e2,e3,e4]",

"DU1u2iu30u4i[e1,e2,e3,e4]",

"DU1u2iu30u40[e1,e2,e3,e4]",

"DU1u2iu30u4p[e1,e2,e3,e4]",

"DU1u2iu3pu4i[e1,e2,e3,e4]",

"DU1u2iu3pu40[e1,e2,e3,e4]",

"DU1u2iu3pu4p[e1,e2,e3,e4]",

"DU1u20u3iu4i[e1,e2,e3,e4]",

"DU1u20u3iu40[e1,e2,e3,e4]",

"DU1u20u3iu4p[e1,e2,e3,e4]",

"DU1u20u30u4i[e1,e2,e3,e4]",

"DU1u20u30u40[e1,e2,e3,e4]",

"DU1u20u30u4p[e1,e2,e3,e4]",

"DU1u20u3pu4i[e1,e2,e3,e4]",

"DU1u20u3pu40[e1,e2,e3,e4]",

"DU1u20u3pu4p[e1,e2,e3,e4]",

"DU1u2pu3iu4i[e1,e2,e3,e4]",

"DU1u2pu3iu40[e1,e2,e3,e4]",

"DU1u2pu3iu4p[e1,e2,e3,e4]",

"DU1u2pu30u4i[e1,e2,e3,e4]",

"DU1u2pu30u40[e1,e2,e3,e4]",

"DU1u2pu30u4p[e1,e2,e3,e4]",

"DU1u2pu3pu4i[e1,e2,e3,e4]",

"DU1u2pu3pu40[e1,e2,e3,e4]",

"DU1u2pu3pu4p[e1,e2,e3,e4]"

}

(\* Generate similar lists (lstS2...lstS4) to be able to refer to the optimal exercise in various periods conditional on the interior or corner solutions that follow \*)

lstS2 = {

"s2iu3iu4i",

"s2iu3iu40",

"s2iu3iu4p",

"s2iu30u4i",

"s2iu30u40",

"s2iu30u4p",

"s2iu3pu4i",

"s2iu3pu40",

"s2iu3pu4p",

"s20",

"s20",

"s20",

"s20",

"s20",

"s20",

"s20",

"s20",

"s20",

"s2p",

"s2p",

"s2p",

"s2p",

"s2p",

"s2p",

"s2p",

"s2p",

"s2p"

}

lstS3 = {

"s3iu4i",

"s3iu40",

"s3iu4p",

"s30",

"s30",

"s30",

"s3p",

"s3p",

"s3p",

"s3iu4i",

"s3iu40",

"s3iu4p",

"s30",

"s30",

"s30",

"s3p",

"s3p",

"s3p",

"s3iu4i",

"s3iu40",

"s3iu4p",

"s30",

"s30",

"s30",

"s3p",

"s3p",

"s3p"

}

lstS4 = {

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p",

"s4i",

"s40",

"s4p"

}

(\* The code below finds optimal values of exercise, permitting for the possibility of a corner solution for it an all subsequent periods, conditional on a particular numerical value of the alphas, beta, delta, and gamma. Alpha values are chosen randomly as are the value for the initial amount of exercise-related capital (kval) and gamma (gval) which is the rate of exercise-related capital depreciates. The discount rate beta (bval) and the amount of myopia delta (dval) are also fixed in this example at 0.97 and 0.9 respectively. This is repeated 10,000 times to sample the solutions that correspond to the values in the parameter space, though in our work we explored 100,000s of parameter sets. Variable definitions: pval is the relative price of exercise (assumed to be one); fff is the file handle where the results are saved. \*)

pval = 1;

bval = 0.97;

dval = 0.90;

ccc = 1;

DistributeDefinitions["Global`"];

ParallelDo[

 ClearMemory[];

 fname = StringJoin["D:\Jeremy\Adherence\_StickK\out\_hyperbolic\_", ToString[$KernelID], ".txt"];

 fff = OpenAppend[fname];

(\* Draw parameters from uniform distributions in their appropriate ranges as described in the text \*)

 a1val = RandomReal[{-1, 0}];

 a2val = RandomReal[{0, 1}];

 a3val = RandomReal[{0, 1}];

 a4val = RandomReal[{-1, 1}];

 a5val = RandomReal[{0, 1}];

 a6val = RandomReal[{-1, 1}];

 a7val = RandomReal[{-1, 0}];

 a8val = RandomReal[{0, 1}];

 a9val = RandomReal[{0, 1}];

 kval = RandomReal[{0, 1}];

 gval = RandomReal[{0, 1}];

(\* Apply the correct formula for U1\* and e1\*, corresponding to the path of corners and future maxima for subsequent exercise \*)

 Do[

 Do[

(\* Interior solution for e1\* \*)

 If[j == 1 && i == 1,

 Clear[s1temp];

 Clear[s1n];

 D1temp = ToExpression[lstDU1[[i]]];

 s1temp = ReplaceRepeated[D1temp, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s1n = Solve[s1temp == 0, e1];

 ,

 nnn=1;

 ];

(\* Lower corner for e1\* \*)

 If[j == 2,

 Clear[s1temp];

 Clear[s1n];

 s1n={e1->0};

 ,

 nnn=2;

 ];

(\* Upper corner for e1\* \*)

 If[j == 3,

 Clear[s1temp];

 Clear[s1n];

 s1n={e1->pval};

 ,

 nnn=3;

 ];

 Clear[s2n];

 Clear[s3n];

 Clear[s4n];

(\* Move to periods 2...4 and compute the optima conditional on the parameter values by using the previously constructed lists (lstS2...lstS4) and corresponding equations \*)

 s2n = ToExpression[lstS2[[i]]];

 s2n = ReplaceRepeated[s2n, s1n];

 s2n = ReplaceRepeated[s2n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s3n = ToExpression[lstS3[[i]]];

 s3n = ReplaceRepeated[s3n, s1n];

 s3n = ReplaceRepeated[s3n, s2n];

 s3n = ReplaceRepeated[s3n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s4n = ToExpression[lstS4[[i]]];

 s4n = ReplaceRepeated[s4n, s1n];

 s4n = ReplaceRepeated[s4n, s2n];

 s4n = ReplaceRepeated[s4n, s3n];

 s4n = ReplaceRepeated[s4n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

(\* Move to periods 2...4 and compute the total utility at the optimal exercise levels \*)

 U4a = U4[e4];

 U4a = ReplaceRepeated[U4a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U4a = ReplaceRepeated[U4a, s4n];

 U4a = ReplaceRepeated[U4a, s3n];

 U4a = ReplaceRepeated[U4a, s2n];

 U4a = ReplaceRepeated[U4a, s1n];

 U3a = U3[e3,e4];

 U3a = ReplaceRepeated[U3a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U3a = ReplaceRepeated[U3a, s4n];

 U3a = ReplaceRepeated[U3a, s3n];

 U3a = ReplaceRepeated[U3a, s2n];

 U3a = ReplaceRepeated[U3a, s1n];

 U2a = U2[e2,e3,e4];

 U2a = ReplaceRepeated[U2a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U2a = ReplaceRepeated[U2a, s4n];

 U2a = ReplaceRepeated[U2a, s3n];

 U2a = ReplaceRepeated[U2a, s2n];

 U2a = ReplaceRepeated[U2a, s1n];

 U1a = U1[e1,e2,e3,e4];

 U1a = ReplaceRepeated[U1a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U1a = ReplaceRepeated[U1a, s4n];

 U1a = ReplaceRepeated[U1a, s3n];

 U1a = ReplaceRepeated[U1a, s2n];

 U1a = ReplaceRepeated[U1a, s1n];

(\* Output the utility function parameters and resulting optima to a file and print it occasionally to the screen to show progress. In this case, every 500 parameter sets. \*)

 WriteString[fff, "\n", ExportString[List[Flatten[{ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]],"Table"]];

If[Mod[ccc,500] == 0,

Print[Flatten[{ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]];

 ,

 mmm=1;

 ];

 ccc = ccc + 1;

 , {i, 1, Length[lstDU1]}

 ];

 , {j, 1, 3}

 ];

 Close[fff];

 , {outerloop, 1, 10000}

];

**A2.2 Identifying Stable Optimal Exercise for Time-Inconsistent People (Stata)**

The Mathematica code produces 81 possible optima for each parameter set considered (interior, lower corner, and upper corner for each of the 4 time periods = 3^4 = 81). Determining which optima would be selected involved solving the following problem. The self in the first time period gets to choose a value of exercise with a plan of how he would like subsequent selves to choose. However, this limits the potential optima from which the self in the second period chooses to 27. Similarly, the second period’s self’s choice leaves the third period self with only nine options, and finally period four self chooses from among 3 possibilities .

Because of time-inconsistency the period four self may not want to follow the plan made by the period three self and hence may choose a different exercise level than the period three self would have liked. Likewise, the same is true for period three relative to period two, etc. Hence, we need to find the stable set of exercise levels chosen.

The algorithm to find the stable solution for a sophisticated hyperbolic discounter involves solving backwards. We start with period 4 and determine what would be his best choice for each of the 27 sets of exercise choices in periods 1 through 3. Likewise, for the self in period 3, we consider the best choice he can make for each of 9 choices made by periods 1 and 2 conditional on what period 4 will choose given the period 3 self’s choice. Similarly for periods 2 and 1 so that the self in period 1 choices from his 3 optima given what the best conditional choices will be at all subsequent periods. Since no self’s divergence from this plan will make him better off the choice is stable.

Below we show the Stata code that processes the output of the Mathematica code and then performs this determination of the “stable optimum” from the 81 candidate optima.

\*\* GLOBALS

global g\_grouping "runtype runid fileid group"

global g\_dirprefix "hyperbolic"

global g\_start 6

global g\_end 6

\*\* END GLOBALS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* There are twelve cores that Mathematica ran on in parallel

\* For each file, read it into Stata

\* Add an id that is just the row number (1 of the 81 possible optima on each parameter set)

\* Create a group identifier (which is the same for the 81 possible optima)

\* If a group has less than 81 possible optima (because the Mathematica code was terminated

\* prematurely, then delete that group

\* record which file these runs are from and potentially which directory they are from

\* in case there have been multiple sets of 12 Mathematica simulations run

\* save this file

forval idir=$g\_start(1)$g\_end {

 forval ifile=1(1)12 {

 insheet using ".\\${g\_dirprefix}\_`idir'\\out\_hyperbolic\_`ifile'.txt", clear double

 gen id = \_n

 bysort id: gen group = ((id-1) - mod((id-1),81))/81

 bysort group: gen todrop = \_N<81

 drop if todrop==1

 gen fileid=`ifile'

 gen runid=`idir'

 save ".\\${g\_dirprefix}\_COMBINING\h\_`idir'\_`ifile'.dta", replace

 }

}

\* Combine the files that were just processed into one master file

forval idir=$g\_start(1)$g\_end {

 forval ifile=1(1)12 {

 if `idir'==$g\_start & `ifile'==1 {

 use ".\\${g\_dirprefix}\_COMBINING\h\_`idir'\_`ifile'.dta", clear

 }

 else {

 append using ".\\${g\_dirprefix}\_COMBINING\h\_`idir'\_`ifile'.dta"

 }

 }

}

\* rename the variables in these files to be meaningful and to correspond to the Mathematica code

\* the optimal values for exercise (s1n, s2n, s3n, and s4n) may have other characters in them

\* and may have different notation as they are strings. These are converted appropriately before

\* the variable is converted into a real number.

rename v1 ccc

rename v2 s1n

replace s1n=substr(s1n,7,.)

replace s1n=regexr(s1n,"\\*\^-","e-")

replace s1n=regexr(s1n,"\\*\^+","e+")

replace s1n=regexr(s1n,"\\*\^","e+")

gen s1nn = real(s1n)

drop s1n

rename s1nn s1n

rename v3 s2n

replace s2n=substr(s2n,7,.)

replace s2n=regexr(s2n,"\\*\^-","e-")

replace s2n=regexr(s2n,"\\*\^+","e+")

replace s2n=regexr(s2n,"\\*\^","e+")

gen s2nn = real(s2n)

drop s2n

rename s2nn s2n

rename v4 s3n

replace s3n=substr(s3n,7,.)

replace s3n=regexr(s3n,"\\*\^-","e-")

replace s3n=regexr(s3n,"\\*\^+","e+")

replace s3n=regexr(s3n,"\\*\^","e+")

gen s3nn = real(s3n)

drop s3n

rename s3nn s3n

rename v5 s4n

replace s4n=substr(s4n,7,.)

replace s4n=regexr(s4n,"\\*\^-","e-")

replace s4n=regexr(s4n,"\\*\^+","e+")

replace s4n=regexr(s4n,"\\*\^","e+")

gen s4nn = real(s4n)

drop s4n

rename s4nn s4n

rename v6 U1a

rename v7 U2a

rename v8 U3a

rename v9 U4a

rename v10 a1val

rename v11 a2val

rename v12 a3val

rename v13 a4val

rename v14 a5val

rename v15 a6val

rename v16 a7val

rename v17 a8val

rename v18 a9val

rename v19 kval

rename v20 pval

rename v21 gval

rename v22 bval

rename v23 dval

gen runtype = "H"

save ".\\${g\_dirprefix}\_COMBINING\h\_ALL.dta", replace

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* This is the code that processes the runs to determine the stable optima

use ".\\${g\_dirprefix}\_COMBINING\h\_ALL.dta", clear

\* Since exercise in a given period must be between 0 and p (in this case 1),

\* all infeasible interior solutions are removed as a corner will be optimal for

\* that particular vector of exercise levels

gen is\_infeasible = 0

replace is\_infeasible = 1 if (s1n<0|s1n>1) | (s2n<0|s2n>1) | (s3n<0|s3n>1) | (s4n<0|s4n>1)

drop if is\_infeasible == 1

\* For each combination of exercise paths in the prior periods, identify the max that

\* period 4’s total utility can be

\* For each combination of exercise path in the prior periods, identify which choice of

\* exercise in period 4 achieves this max

bysort $g\_grouping s1n s2n s3n: egen double max\_U4 = max(U4a)

gen is\_U4\_max = 0

bysort $g\_grouping s1n s2n s3n: replace is\_U4\_max = U4a == max\_U4

\* Likewise, for period 3, determine the maximum of the total utility function for each

\* combination of exercise paths in prior periods and conditional on period 4 choosing the

\* optimal path from period 4’s perspective conditional on period 3’s choice

bysort $g\_grouping s1n s2n is\_U4\_max: egen double max\_U3 = max(U3a)

gen is\_U3\_max = 0

bysort $g\_grouping s1n s2n is\_U4\_max: replace is\_U3\_max = U3a == max\_U3 if is\_U4\_max==1

\* Repeat the above for period 2

bysort $g\_grouping s1n is\_U3\_max is\_U4\_max: egen double max\_U2 = max(U2a)

gen is\_U2\_max = 0

bysort $g\_grouping s1n is\_U3\_max is\_U4\_max: replace is\_U2\_max = U2a == max\_U2 if is\_U3\_max==1 & is\_U4\_max==1

\* Repeat the above for period 1

bysort $g\_grouping is\_U2\_max is\_U3\_max is\_U4\_max: egen double max\_U1 = max(U1a)

gen is\_U1\_max = 0

bysort $g\_grouping is\_U2\_max is\_U3\_max is\_U4\_max: replace is\_U1\_max = U1a == max\_U1 if is\_U2\_max==1 & is\_U3\_max==1 & is\_U4\_max==1

\* Identify the exercise choices for each parameter set that stably maximize utility for all

\* 4 periods conditional on the choices that selves in prior periods make (bbb=1)

\* In the extremely unlikely event of a tie for the best,

\* randomly select one of the exercise paths that ties for the best as the

\* chosen path and set the others to be the non-optimal path (bbb=0)

gen bbb = (is\_U1\_max==1&is\_U2\_max==1&is\_U3\_max==1&is\_U4\_max==1)

bysort $g\_grouping: gen sumbbb = sum(bbb)

bysort $g\_grouping: egen totbbb = total(bbb)

bysort $g\_grouping: gen randombbb = 1 + round(uniform(),1) if \_n==1

bysort $g\_grouping: replace randombbb = randombbb[1]

replace bbb = 0 if randombbb != sumbbb & totbbb==2 & bbb==1

**A2.3 Exploring Comparative Statics on Optimal Exercise Paths (Mathematica)**

In this section, we annotate the Mathematica code that we use to estimate the comparative statics of changing gamma (gval) and delta (dval) – the rate of exercise-related human capita depreciation and the extent of myopia respectively – on exercise outcomes. Perhaps not surprisingly, this code is similar to the we used to identify the optimal exercise paths earlier.

We fixed the discount rate, beta (bval), at 0.97 and randomly draw 5,000 sets of utility function parameters – alphas (avals) – and initial exercise related human capital (kval). For each random set of alpha’s and kval’s, we solve the model for a grid of dval and gval levels, where these two parameters take on values between 0.50 and 1.00 in 0.05 increments (11x11 combinations=121). In total, we examined 605,000 parameter sets.

Before running this code, one must first run the code up through the construction of the lists (lstS1...lstS4) above in Section A2.1. After running this code, one can use the Stata code listed above in Section A2.2 to process the output.

pval = 1;

bval = 0.97;

ccc = 1;

DistributeDefinitions["Global`"];

ParallelDo[

 ClearMemory[];

 ClearSystemCache[];

 $HistoryLength = 5;

 a1val = RandomReal[{-1, 0}];

 a2val = RandomReal[{0, 1}];

 a3val = RandomReal[{0, 1}];

 a4val = RandomReal[{-1, 1}];

 a5val = RandomReal[{0, 1}];

 a6val = RandomReal[{-1, 1}];

 a7val = RandomReal[{-1, 0}];

 a8val = RandomReal[{0, 1}];

 a9val = RandomReal[{0, 1}];

 kval = RandomReal[{0, 1}];

 fffinconsistent = OpenAppend[StringJoin["D:\\Jeremy\\Adherence\_StickK\\out\_hyperbolic\_", ToString[$KernelID], ".txt"]];

 Do[

 Do[

 Do[

 Do[

 If[j == 1 && i == 1,

 Clear[s1temp];

 Clear[s1n];

 D1temp = ToExpression[lstDU1[[i]]];

 s1temp = ReplaceRepeated[D1temp, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s1n = Solve[s1temp == 0, e1];

 ,

 nnn=1;

 ];

 If[j == 2,

 Clear[s1temp];

 Clear[s1n];

 s1n={e1->0};

 ,

 nnn=2;

 ];

 If[j == 3,

 Clear[s1temp];

 Clear[s1n];

 s1n={e1->pval};

 ,

 nnn=3;

 ];

 Clear[s2n];

 Clear[s3n];

 Clear[s4n];

 s2n = ToExpression[lstS2[[i]]];

 s2n = ReplaceRepeated[s2n, s1n];

 s2n = ReplaceRepeated[s2n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s3n = ToExpression[lstS3[[i]]];

 s3n = ReplaceRepeated[s3n, s1n];

 s3n = ReplaceRepeated[s3n, s2n];

 s3n = ReplaceRepeated[s3n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s4n = ToExpression[lstS4[[i]]];

 s4n = ReplaceRepeated[s4n, s1n];

 s4n = ReplaceRepeated[s4n, s2n];

 s4n = ReplaceRepeated[s4n, s3n];

 s4n = ReplaceRepeated[s4n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U4a = U4[e4];

 U4a = ReplaceRepeated[U4a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U4a = ReplaceRepeated[U4a, s4n];

 U4a = ReplaceRepeated[U4a, s3n];

 U4a = ReplaceRepeated[U4a, s2n];

 U4a = ReplaceRepeated[U4a, s1n];

 U3a = U3[e3,e4];

 U3a = ReplaceRepeated[U3a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U3a = ReplaceRepeated[U3a, s4n];

 U3a = ReplaceRepeated[U3a, s3n];

 U3a = ReplaceRepeated[U3a, s2n];

 U3a = ReplaceRepeated[U3a, s1n];

 U2a = U2[e2,e3,e4];

 U2a = ReplaceRepeated[U2a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U2a = ReplaceRepeated[U2a, s4n];

 U2a = ReplaceRepeated[U2a, s3n];

 U2a = ReplaceRepeated[U2a, s2n];

 U2a = ReplaceRepeated[U2a, s1n];

 U1a = U1[e1,e2,e3,e4];

 U1a = ReplaceRepeated[U1a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U1a = ReplaceRepeated[U1a, s4n];

 U1a = ReplaceRepeated[U1a, s3n];

 U1a = ReplaceRepeated[U1a, s2n];

 U1a = ReplaceRepeated[U1a, s1n];

 WriteString[fffinconsistent, "\n", ExportString[List[Flatten[{ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]],"Table"]];

 If[Mod[ccc,500] == 0,

 Print[Flatten[{ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]];

 ,

 mmm=1;

 ];

 ccc = ccc + 1;

 , {i, 1, Length[lstDU1]}

 ];

 , {j, 1, 3}

 ];

 , {gval, 0.5 , 1, 0.05}

 ];

 , {dval, 0.5, 1, 0.05}

 ];

 Close[fffinconsistent];

 , {outerloop, 1, 5000}

];

**A2.4 Exploring the Effect of Nudges on Optimal Exercise Paths (Mathematica)**

We next explore the effect of nudges using the following Mathematica code. The code below assumes the following scenario. The first period self exercises, becomes aware of pre-commitment contracts as an alternative to the stable optimal exercise choices that would have otherwise followed and makes a pre-commitment contract for the self in period 2. To explore how different nudges on this contract’s length (intensity) in period 2 could potentially alter exercise choices in periods 3 and 4 and total utility from each period’s perspective, the code fixes the exercise chosen in period 2 at many different “nudged” levels (with no option for period 2 to deviate). For each of these levels, it then determines the 9 potential optimal choices (interior, lower corner, upper corner) for the selves in period 3 and period 4. Thus, conditional on exercise in period 1 and the nudged level in period 2, it computes potential optimal subsequent exercise choices as well as total utility under these conditions.

Before running this code, one must first run the code up through the construction of the lists (lstS1...lstS4) above in Section A2.1. After running this code, one can use the Stata code listed above in Section A2.2 to process the output.

(\* Import parameter sets for the utility function \*)

(\* For each set, the effects of nudges of various sizes will be considered \*)

ccc = 1;

ParamSets = Import["D:\\Jeremy\\Adherence\_StickK\\ParamSets\_ALL.txt", "Table"];

DistributeDefinitions["Global`"];

ParallelDo[

 ClearMemory[];

 ClearSystemCache[];

 $HistoryLength = 5;

(\* Load the next set of parameters \*)

 a1val = ParamSets[[outerloop, 1]];

 a2val = ParamSets[[outerloop, 2]];

 a3val = ParamSets[[outerloop, 3]];

 a4val = ParamSets[[outerloop, 4]];

 a5val = ParamSets[[outerloop, 5]];

 a6val = ParamSets[[outerloop, 6]];

 a7val = ParamSets[[outerloop, 7]];

 a8val = ParamSets[[outerloop, 8]];

 a9val = ParamSets[[outerloop, 9]];

 kval = ParamSets[[outerloop, 10]];

 gval = ParamSets[[outerloop, 11]];

 pval = ParamSets[[outerloop, 12]];

 bval = ParamSets[[outerloop, 13]];

 dval = ParamSets[[outerloop, 14]];

 s1tempval = ParamSets[[outerloop, 15]];

 fffinconsistent = OpenAppend[StringJoin["D:\\Jeremy\\Adherence\_StickK\\out\_hyperbolic\_explore", ToString[$KernelID], ".txt"]];

 Do[

 Do[

 Do[

(\* Unlike the previous Mathematica code, here it is assumed that period 1 chooses exercise based on the stable optimum and afterwards makes a decision to commit period 2.\*)

(\* The loops sweep across possible commitments considering commitments across the range of nudges from 0 to 1 \*)

 If[j == 2,

 Clear[s1temp];

 Clear[s1n];

 Clear[s2n];

 s1n={e1->s1tempval};

 s2n={e2->s2temp};

 ,

 nnn=2;

 ];

 Clear[s3n];

 Clear[s4n];

(\* Compute optimal period 3 and period 4 exercise conditional on model parameters and the nudged exercise path \*)

 s3n = ToExpression[lstS3[[i]]];

 s3n = ReplaceRepeated[s3n, s1n];

 s3n = ReplaceRepeated[s3n, s2n];

 s3n = ReplaceRepeated[s3n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 s4n = ToExpression[lstS4[[i]]];

 s4n = ReplaceRepeated[s4n, s1n];

 s4n = ReplaceRepeated[s4n, s2n];

 s4n = ReplaceRepeated[s4n, s3n];

 s4n = ReplaceRepeated[s4n, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

(\* Compute the total utility implications of this exercise consumption path \*)

 U4a = U4[e4];

 U4a = ReplaceRepeated[U4a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U4a = ReplaceRepeated[U4a, s4n];

 U4a = ReplaceRepeated[U4a, s3n];

 U4a = ReplaceRepeated[U4a, s2n];

 U4a = ReplaceRepeated[U4a, s1n];

 U3a = U3[e3,e4];

 U3a = ReplaceRepeated[U3a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U3a = ReplaceRepeated[U3a, s4n];

 U3a = ReplaceRepeated[U3a, s3n];

 U3a = ReplaceRepeated[U3a, s2n];

 U3a = ReplaceRepeated[U3a, s1n];

 U2a = U2[e2,e3,e4];

 U2a = ReplaceRepeated[U2a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U2a = ReplaceRepeated[U2a, s4n];

 U2a = ReplaceRepeated[U2a, s3n];

 U2a = ReplaceRepeated[U2a, s2n];

 U2a = ReplaceRepeated[U2a, s1n];

 U1a = U1[e1,e2,e3,e4];

 U1a = ReplaceRepeated[U1a, {a1 -> a1val, a2 -> a2val, a3 -> a3val, a4 -> a4val, a5 -> a5val, a6 -> a6val, a7 -> a7val, a8 -> a8val, a9 -> a9val, k0 -> kval, p -> pval, g -> gval, b -> bval, d -> dval}];

 U1a = ReplaceRepeated[U1a, s4n];

 U1a = ReplaceRepeated[U1a, s3n];

 U1a = ReplaceRepeated[U1a, s2n];

 U1a = ReplaceRepeated[U1a, s1n];

 WriteString[fffinconsistent, "\n", ExportString[List[Flatten[{outerloop, ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]],"Table"]];

 If[Mod[1000\*s2temp,100] == 0,

 Print[Flatten[{outerloop, ccc, s1n, s2n, s3n, s4n, U1a, U2a, U3a, U4a, a1val, a2val, a3val, a4val, a5val, a6val, a7val, a8val, a9val, kval, pval, gval, bval, dval}]];

 ,

 mmm=1;

 ];

 ccc = ccc + 1;

 , {i, 1, Length[lstDU1]}

 ];

 , {j, 2, 2}

 ];

 , {s2temp, 0, 1, 0.001}

 ];

 Close[fffinconsistent];

 , {outerloop, 1, 197}

];