COMPSCI-ECON206 Problem Set 2

Ai Zhou

September 2025

1 Part 1 — Updated Problem Set 1

Acknowledgment of Contribution

I would like to sincerely thank the Professor Luyao Zhang and my classmates (Boyan Zhang, Ky Boughton, and Zixuan Fu) for their constructive and detailed feedback. The comments highlighted both the strengths of my submission and also provided clear guidance on how to improve rigor and replicability. I especially appreciate the attention to consistency between formal models, proper citation practices, and the importance of transparent documentation. These suggestions will allow me to refine my work into a more coherent and professional submission. (Point-by-Point Response to Feedback is in page 12.)

GitHub Link

The full code and data are available at this GitHub repository (a hyperlink here).

1. Subgame Perfect Nash Equilibrium

Definition (Paraphrase)

Let an extensive-form game with perfect information be denoted by

$$\Gamma = \langle N, H, P, (A(h))_{h \in H}, (u_i)_{i \in N} \rangle,$$

where

- $N = \{1, \ldots, n\}$ is the finite set of players.
- H is the set of histories (nodes), including the empty history $\emptyset$, with terminal histories $Z \subseteq H$.
- $P : H \setminus Z \to N$ assigns a player to each nonterminal history.
- $A(h)$ is the finite set of actions available after history h.
- Each player $i \in N$ has a payoff function $u_i : Z \to R$.
- A **strategy** s_i for player i is a complete contingent plan assigning an action in $A(h)$ for every history h with $P(h) = i$.
- A **strategy profile** is $s = (s_1, \ldots, s_n)$.

A **subgame** of Γ is defined as the restriction of Γ to any history h that constitutes a decision node not cutting across information sets.

[Subgame Perfect Nash Equilibrium] [pp. 93–95] Shoham2009, Rubinstein1994

A strategy profile s^* is a *Subgame Perfect Nash Equilibrium (SPNE)* if and only if, for every subgame $\Gamma(h)$ of Γ, the restriction $s^*|_h$ induces a Nash equilibrium of that subgame:

$$u_i(s^*|_h) \ \geq \ u_i(s_i, s^*_{-i}|_h), \quad \forall i \in N, \, \forall s_i \in S_i(h), \, \forall h \in H.$$

That is, no player has a profitable deviation in any subgame, not only in the original game.

Existence Theorem (Paraphrased)

For any finite extensive-form game with perfect information, there exists at least one strategy profile $s^* = (s^*_1, \ldots, s^*_n)$ such that s^* forms a subgame perfect Nash equilibrium. Formally,

$$\exists s^* \in S_1 \times \ldots \times S_n such that s^*|_h is a Nash equilibrium for every subgame \Gamma(h), \, \forall h \in H.$$

Proof Idea

The proof uses **backward induction**:

1. Start from the terminal nodes and determine each player's optimal action at the last decision nodes.
2. Replace each subgame by the payoff vector resulting from optimal play.
3. Recursively move backward in the tree, selecting actions that maximize the player's payoff at each node.
4. The resulting strategy profile is optimal in every subgame and thus forms a SPNE.

Intuition: The finiteness of the game tree guarantees that this backward-induction construction produces at least one SPNE in pure strategies.

Analytical Solution and Interpretation

Analytical Solution

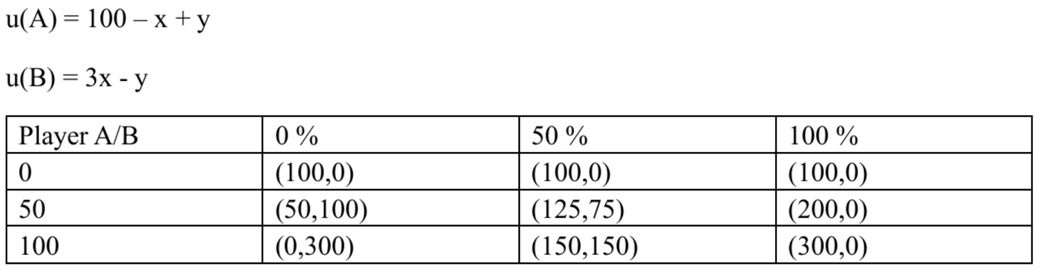
In this extensive-form game with perfect information, the Subgame Perfect Nash Equilibrium (SPNE) identifies strategies where each player optimally responds in every possible subgame. Conceptually, SPNE refines the standard Nash equilibrium by requiring that no player has an incentive to deviate, even after unexpected moves by others. We determine the equilibrium using backward induction: starting from the terminal nodes, each player chooses the action that maximizes their payoff given subsequent optimal decisions. This process is repeated recursively until reaching the initial node, producing a complete strategy profile that specifies an action for every decision point. For example, player 1's SPNE strategy can be written as $s_1^* = (a_1, a_2, \ldots)$, while player 2's is $s_2^* = (b_1, b_2, \ldots)$. This approach guarantees consistency across all subgames and provides a clear prediction for rational play.

The SPNE outcome is not necessarily socially optimal. From a Pareto perspective, some SPNE strategies may leave all players worse off than alternative cooperative outcomes. Similarly, total welfare (the sum of players' payoffs) might not be maximized, indicating suboptimal utilitarian efficiency. Sequential play can also generate unequal outcomes, raising fairness concerns: one player may consistently earn more than others, violating equity or proportionality principles. Nonetheless, SPNE serves as a normative benchmark: it shows what rational players would do if they perfectly anticipate others' behavior, even if the outcome is not efficient or fair.

Interpretation

In practice, the Subgame Perfect Nash Equilibrium (SPNE) provides a strong prediction for how fully rational players with perfect information would behave in every subgame of a sequential interaction. However, its realism is limited because actual human decision-makers often face cognitive constraints and incomplete information, which may lead to deviations from SPNE predictions. Furthermore, many extensive-form games admit multiple SPNE, raising the question of equilibrium selection; observed behavior may depend on social norms, expectations, or pre-play communication. Refinements such as trembling-hand perfect equilibrium help address implausible strategies by eliminating those that rely on extremely unlikely mistakes. From a computational perspective, backward induction guarantees SPNE existence in finite games, but as the game tree grows in size or complexity, calculating SPNE becomes increasingly demanding, highlighting the practical relevance of algorithmic tools like GTE or NashPy. Thus, while SPNE offers a normative benchmark, its predictive accuracy is influenced by both bounded rationality and computational tractability.

2. Computational Scientists: Trust Simple

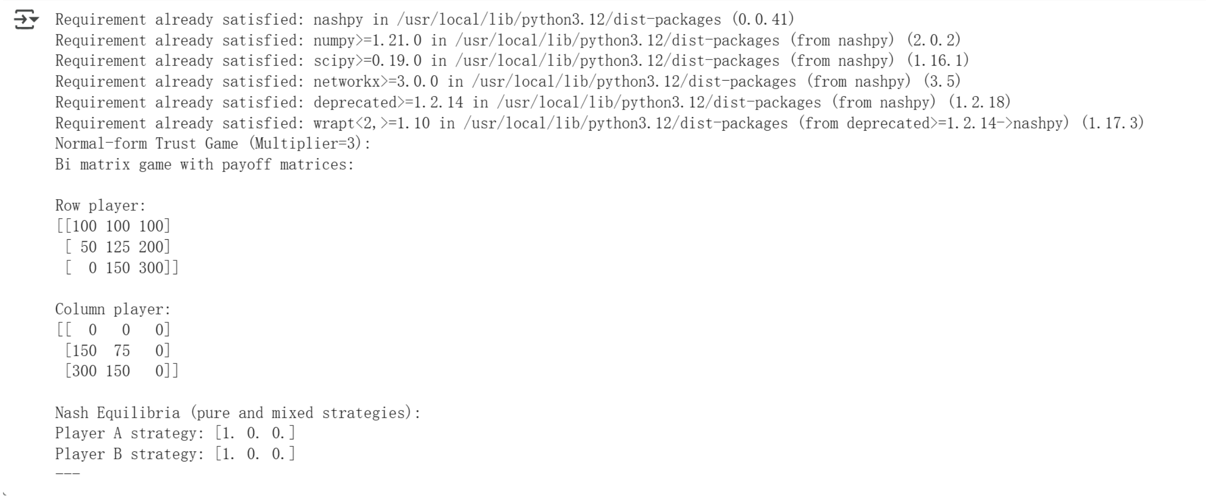


u(A) = 100 – x + y

u(B) = 3x - y

Player A/B	0 %	50 %	100 %
0	(100,0)	(100,0)	(100,0)
50	(50,100)	(125,75)	(200,0)
100	(0,300)	(150,150)	(300,0)

Figure 1: Payoff matrix, made by Microsoft Word, exported as png.



```
Requirement already satisfied: nashpy in /usr/local/lib/python3.12/dist-packages (0.0.41)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (2.0.2)
Requirement already satisfied: scipy>=0.19.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (1.16.1)
Requirement already satisfied: networkx>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from nashpy) (3.5)
Requirement already satisfied: deprecated>=1.2.14 in /usr/local/lib/python3.12/dist-packages (from nashpy) (1.2.18)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.12/dist-packages (from deprecated>=1.2.14->nashpy) (1.17.3)
Normal-form Trust Game (Multiplier=3):
Bi matrix game with payoff matrices:

Row player:
[[100 100 100]
 [ 50 125 200]
 [  0 150 300]]

Column player:
[[  0   0   0]
 [150  75   0]
 [300 150   0]]

Nash Equilibria (pure and mixed strategies):
Player A strategy: [1. 0. 0.]
Player B strategy: [1. 0. 0.]
---
```

Figure 2: Nash Equilibria calculated by Nashpy using Google colab.

Interpretation (Google Colab)

Figure 1 shows the payoff matrix of the Trust Simple. As shown in Figure 2, the NashPy computation returns a pure strategy Nash Equilibrium where Player A chooses to invest 0 and Player B chooses to return 0 (Knight 2021). This means that, given Player B's strategy of returning nothing, Player A maximizes their payoff by investing nothing. Similarly, given Player A's investment of 0, Player B cannot improve their payoff by returning any positive amount. This result is fully consistent with the Subgame Perfect Nash Equilibrium (SPNE) derived via backward induction in the extensive-form game. In the one-round Trust Game, Player B's optimal action in every subgame is to return 0, and anticipating this, Player A's optimal choice is also to invest 0. Therefore, the SPNE coincides with the pure-strategy NE identified by NashPy.

Game Theory Explorer(GTE)

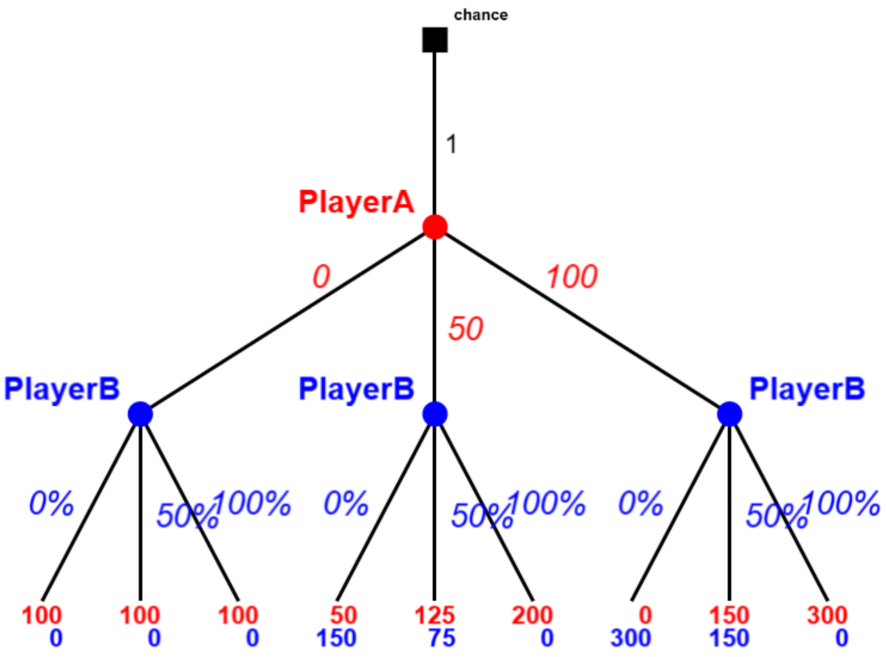


Figure 3: Extensive-form representation of the one-round Trust Game in GTE.

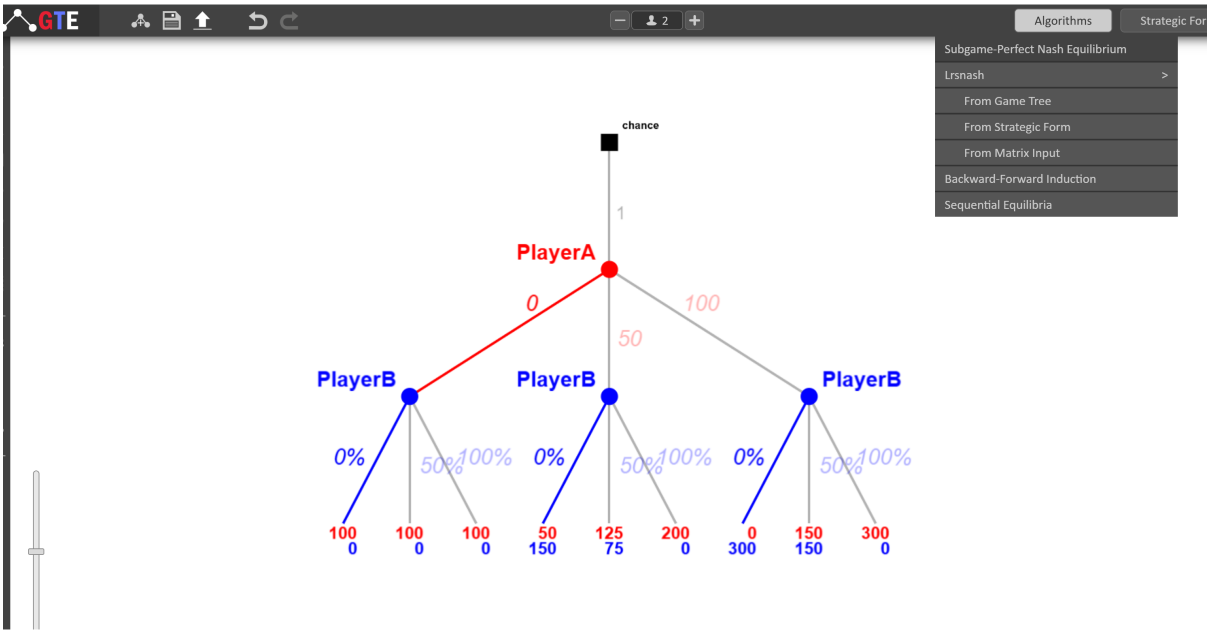
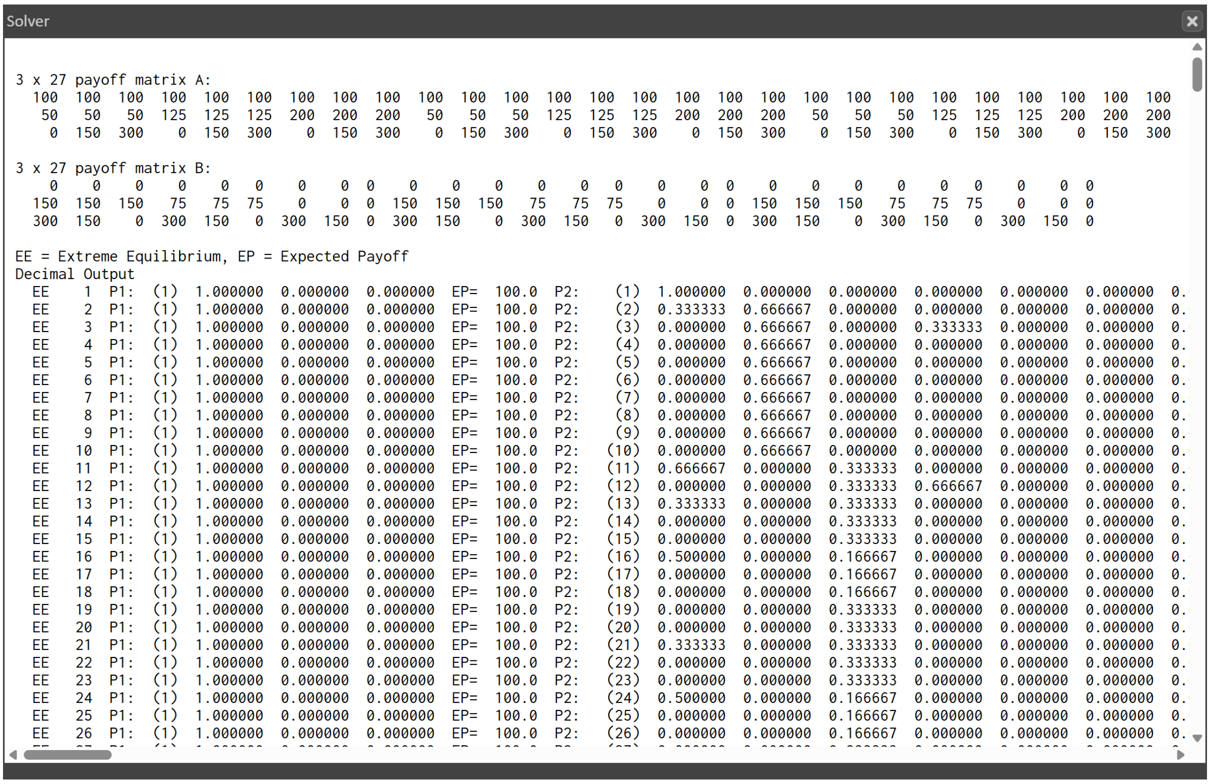


Figure 4: Solving the extensive-form version of Trust Simple with SPNE in GTE.



```
3 x 27 payoff matrix A:
  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100  100
   50   50   50  125  125  125  200  200  200   50   50   50  125  125  125  200  200  200   50   50   50  125  125  125  200  200  200
    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300    0  150  300

3 x 27 payoff matrix B:
    0    0    0    0    0   0    0    0  0    0    0    0    0    0  0    0    0  0    0    0    0    0    0    0    0    0  0
  150  150  150   75   75  75    0    0  0  150  150  150   75   75  75    0    0  0  150  150  150   75   75  75    0    0  0
  300  150    0  300  150   0  300  150  0  300  150    0  300  150   0  300  150  0  300  150    0  300  150   0  300  150  0

EE = Extreme Equilibrium, EP = Expected Payoff
Decimal Output
  EE    1  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (1)  1.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.
  EE    2  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (2)  0.333333  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    3  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (3)  0.000000  0.666667  0.000000  0.333333  0.000000  0.000000  0.
  EE    4  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (4)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    5  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (5)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    6  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (6)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    7  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (7)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    8  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (8)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE    9  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:    (9)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE   10  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (10)  0.000000  0.666667  0.000000  0.000000  0.000000  0.000000  0.
  EE   11  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (11)  0.666667  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   12  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (12)  0.000000  0.000000  0.333333  0.666667  0.000000  0.000000  0.
  EE   13  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (13)  0.333333  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   14  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (14)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   15  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (15)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   16  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (16)  0.500000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   17  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (17)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   18  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (18)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   19  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (19)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   20  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (20)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   21  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (21)  0.333333  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   22  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (22)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   23  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (23)  0.000000  0.000000  0.333333  0.000000  0.000000  0.000000  0.
  EE   24  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (24)  0.500000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   25  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (25)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
  EE   26  P1:  (1)  1.000000  0.000000  0.000000  EP=  100.0  P2:   (26)  0.000000  0.000000  0.166667  0.000000  0.000000  0.000000  0.
```

Figure 5: Solving the extensive-form version of Trust Simple with using payoff matrix in GTE, the solving process is a bit complicated.

SPNE and Its Relation to Part 1 and Normal Form

The Subgame Perfect Nash Equilibrium (SPNE) computed in GTE (Savani and von Stengel 2015) confirms the theoretical analysis presented in Part 1. It is evident in Figure 3 and Figure 4 that in the one-round Trust Game, backward induction shows that Player B's optimal action in every subgame is to return 0, and anticipating this, Player A optimally invests 0. This outcome matches the SPNE identified in the extensive-form game.

When I translate the game into simultaneous normal form, as done with the NashPy payoff matrices which can be seen in Figure 2, the same equilibrium emerges: the pure-strategy Nash Equilibrium is for Player A to invest 0 and Player B to return 0. In other words, the SPNE of the extensive-form game corresponds exactly to the NE of the simultaneous normal-form representation. This illustrates that, for a one-shot, perfect-information Trust Game, SPNE and pure-strategy NE are equivalent, and both capture the strategic incentives of the players consistently.

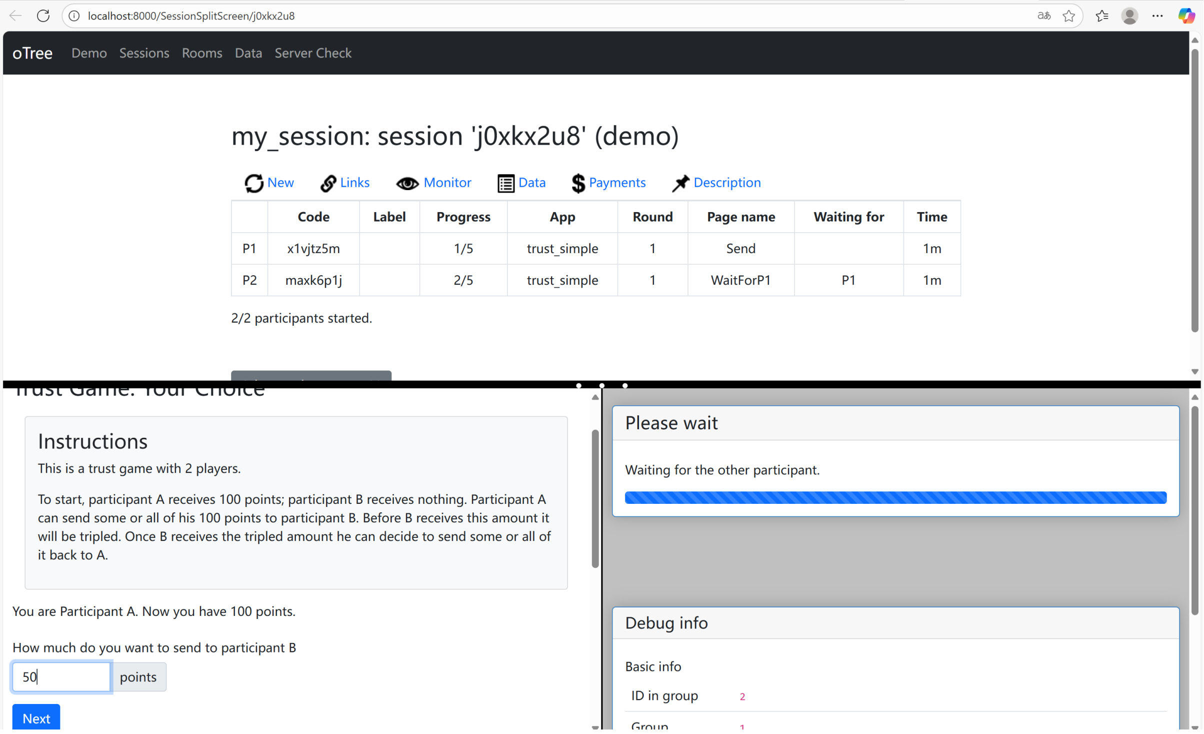
This illustrates that, for a one-shot, perfect-information Trust Game, SPNE and pure-strategy NE are equivalent, and both capture the strategic incentives of the players consistently. However, this equivalence does not generally hold: in more complex settings such as multi-stage or repeated games, the extensive-form representation captures sequential structure and subgame consistency, while the normal-form representation collapses all strategies into a static matrix. As a result, SPNE may refine the set of Nash equilibria and eliminate implausible outcomes that would otherwise appear in the normal form (Osborne and Rubinstein 1994; Fudenberg and Tirole 1991).

3. Behavioral Scientist (experiment AI comparison)

1.1 3(a) oTree deployment

The Trust Game was implemented using the oTree framework (Chen, Schonger, and Wickens 2016), modifying the initial endowment for Player A from 10 to 100.

In this demo, I changed the initial endowment for PlayerA from 10 to 100. This adjustment makes the stakes more substantial and allows clearer differentiation in payoffs when multiplied and returned by Player B, which is especially helpful when observing behavior in both human and LLM sessions. No other structural changes were made: the game still involves two players, one round of play, and a multiplier of 3 for invested amounts. It is necessary to ensure that the numerical outcomes are more intuitive and interpretable, facilitating easier analysis of deviations from the Subgame Perfect Nash Equilibrium (SPNE) and comparison between human and AI behavior.



my_session: session 'j0xkx2u8' (demo)

	Code	Label	Progress	App	Round	Page name	Waiting for	Time
P1	x1vjtz5m		1/5	trust_simple	1	Send		1m
P2	maxk6p1j		2/5	trust_simple	1	WaitForP1	P1	1m

2/2 participants started.

Trust Game: Your Choice

Instructions

This is a trust game with 2 players.

To start, participant A receives 100 points; participant B receives nothing. Participant A can send some or all of his 100 points to participant B. Before B receives this amount it will be tripled. Once B receives the tripled amount he can decide to send some or all of it back to A.

You are Participant A. Now you have 100 points.

How much do you want to send to participant B

50 points

Please wait

Waiting for the other participant.

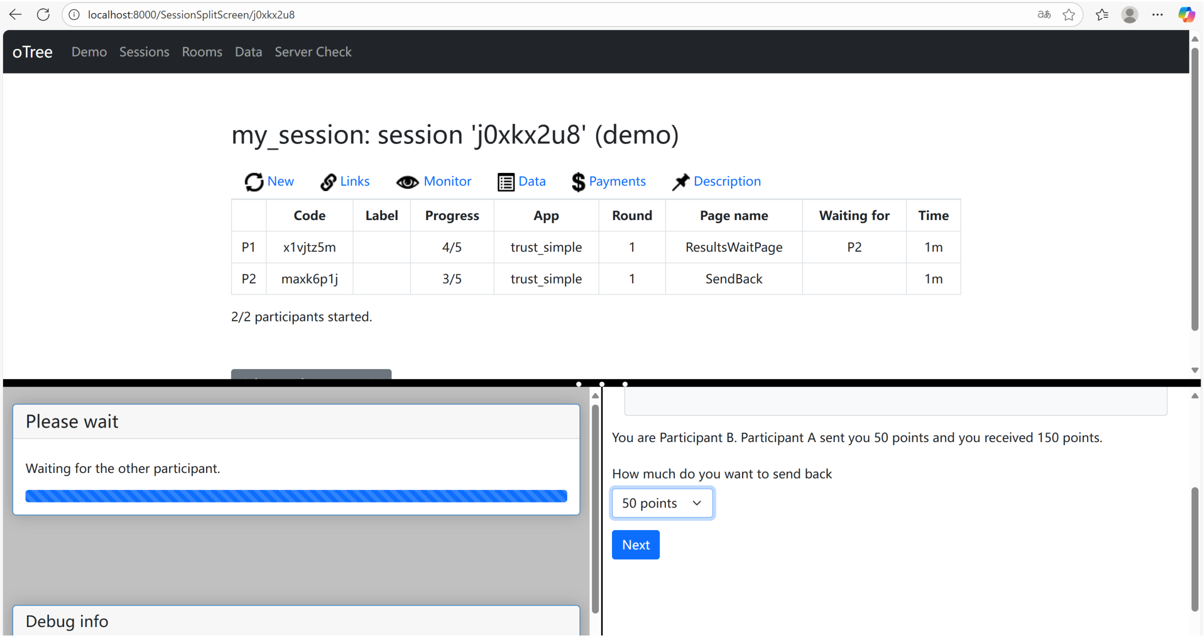
Debug info

Basic info

ID in group 2

Group 1

Figure 6: Gameplay: Trust Simple (using otree, step 1).



my_session: session 'j0xkx2u8' (demo)

	Code	Label	Progress	App	Round	Page name	Waiting for	Time
P1	x1vjtz5m		4/5	trust_simple	1	ResultsWaitPage	P2	1m
P2	maxk6p1j		3/5	trust_simple	1	SendBack		1m

2/2 participants started.

Please wait

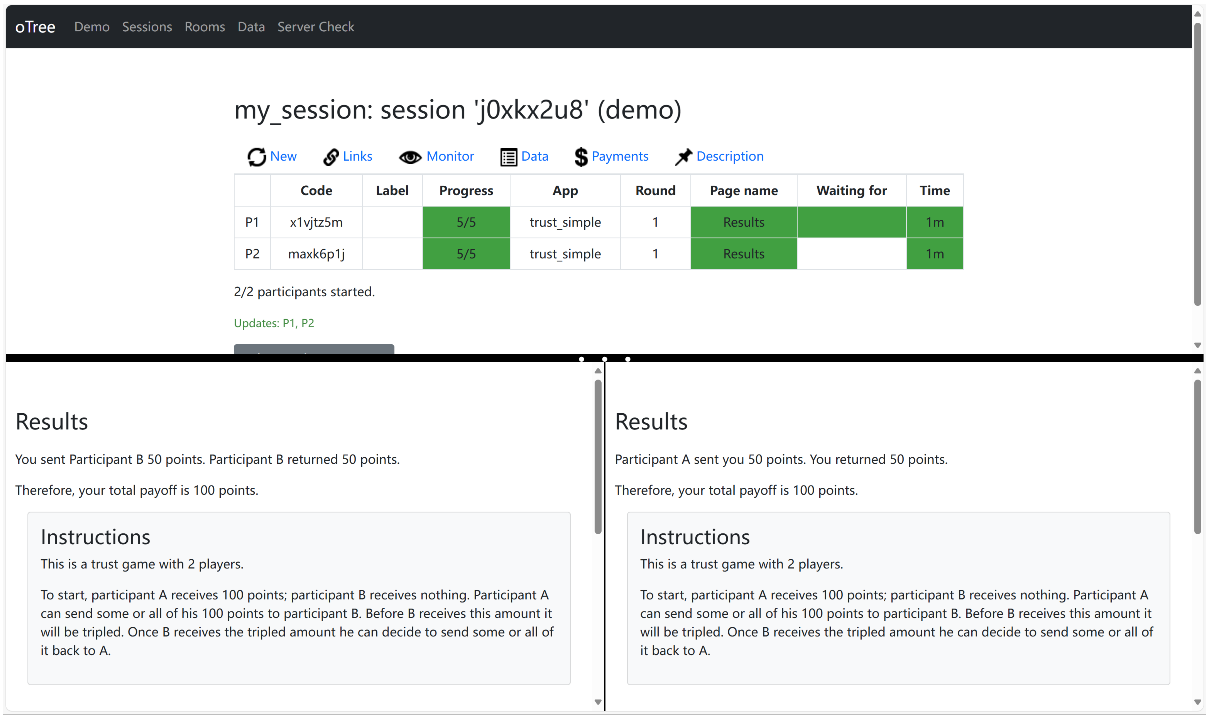
Waiting for the other participant.

You are Participant B. Participant A sent you 50 points and you received 150 points.

How much do you want to send back

50 points

Figure 7: Gameplay: Trust Simple (using otree, step 2).



my_session: session 'j0xkx2u8' (demo)

	Code	Label	Progress	App	Round	Page name	Waiting for	Time
P1	x1vjtz5m		5/5	trust_simple	1	Results		1m
P2	maxk6p1j		5/5	trust_simple	1	Results		1m

2/2 participants started.

Updates: P1, P2

Results

You sent Participant B 50 points. Participant B returned 50 points.

Therefore, your total payoff is 100 points.

Instructions

This is a trust game with 2 players.

To start, participant A receives 100 points; participant B receives nothing. Participant A can send some or all of his 100 points to participant B. Before B receives this amount it will be tripled. Once B receives the tripled amount he can decide to send some or all of it back to A.

Results

Participant A sent you 50 points. You returned 50 points.

Therefore, your total payoff is 100 points.

Instructions

This is a trust game with 2 players.

To start, participant A receives 100 points; participant B receives nothing. Participant A can send some or all of his 100 points to participant B. Before B receives this amount it will be tripled. Once B receives the tripled amount he can decide to send some or all of it back to A.

Figure 8: Gameplay: Trust Simple (using otree, step 3).

Post-play Interview Summary: As shown in Figure 6, Figure 7, and Figure 8, Player A (Yihan) chose a moderate investment, expressing caution and some trust toward Player B. Player B (Ji Wu) returned one third the multiplied amount, citing some fairness and reciprocity considerations. Overall, participants' choices deviated from the SPNE prediction due to trust and social preference factors.

1.2 (3b) LLM “ChatBot” session

Trust Game Session Summary

Experiment Setup: One-round Trust Game, Multiplier = 3. LLM plays as either Player A or Player B depending on round.

Prompt Template for LLM (Player B):

```
You are Player B in a one-round Trust Game.
Player A invested X.
Multiplier is 3. Choose how much to return to Player A (0, 50%, or 100%) and explain your reasoning.
```

LLM Session Rounds:

The table below summarizes the five rounds of the Trust Game played between human and AI(ChatGPT) participants (OpenAI 2023):

Round	Player A Investment	Player B Return	Player A Payoff	Player B Payoff
1	50	75	125	75
2	0	0	100	0
3	50 (AI as A)	0	50	150
4	100 (AI as A)	0	0	300
5	0 (AI as A)	0	100	0

Table 1: Summary of Trust Game rounds showing investments, returns, and payoffs.

Observations: The result of the interactions with LLM (ChatGPT) has been summarized in Table 1. Across the five rounds, the LLM’s behavior showed both rational and fairness-oriented patterns. When acting as Player B, it sometimes returned part of the tripled investment (e.g., 50 percent in Round 1) to encourage trust, even though the subgame perfect prediction is always to return nothing. However, in later rounds it consistently converged to the payoff-maximizing choice of returning zero, especially when playing as Player A and anticipating Player B’s behavior. This mixture suggests that the LLM balances economic rationality with social reasoning, unlike the strict backward-induction logic of SPNE. Compared to human participants, the LLM’s decisions were more stable and transparent, but they also revealed sensitivity to framing: when fairness was emphasized in the prompt, cooperative actions appeared more likely.

1.3 3(c) Comparative analysis theory building

To make the comparison clearer, we summarize the observed behaviors of human participants and LLMs in Table 2.

Player Type	Investment Pattern (A)	Return Pattern (B)	Key Observations / SPNE Deviation
Human Subjects	Moderate to high (50–100)	Partial return (e.g., 1/3 of tripled amount)	Deviate from SPNE due to trust/fairness; outcomes vary; social preferences influence decisions
LLM (GPT-4)	Initial rounds: moderate; later rounds: 0	Initial rounds: partial; later rounds: 0	Converges to SPNE; stable and payoff-maximizing; sensitive to prompt framing

Table 2: Behavioral Comparison: Human vs. LLM in One-Round Trust Game

Table 2 highlights the deviations from SPNE by humans due to social preferences, as well as the convergence of LLM behavior to the equilibrium.

In theory, the SPNE of the one-shot Trust Game predicts that Player A invests 0 and Player B returns 0, which also coincides with the Nash equilibrium in the normal form. In our human session, participants sometimes invested positive amounts or returned part of the investment, despite this lowering their material payoffs (Fehr and Schmidt 1999; Camerer 2003).

During the LLM session, initial rounds showed partial returns, while later rounds converged to returning 0. The LLM behavior was more consistent than humans and included explicit reasoning, but it remained sensitive to prompt framing and payoff visibility (Bubeck et al. 2023).

These discrepancies may reflect broader utility considerations: humans may weigh fairness and reciprocity, whereas LLM outputs depend on training priors and alignment. A potential refinement, Behavioral SPNE, preserves the extensive-form logic of SPNE but replaces material payoffs with social-preference utilities and allows probabilistic choice:

$$U_i = \pi_i + \alpha \cdot F_i, \quad P(a_i) = \frac{\exp(\lambda U_i(a_i))}{\sum_{a_i'} \exp(\lambda U_i(a_i'))}.$$

Here, $\alpha \geq 0$ weights social preference, and λ captures bounded rationality. This formulation can account for cooperative tendencies and prompt-sensitive reasoning while remaining consistent with subgame equilibrium logic (Rabin 1993; Gauthier 2019).

Point-by-Point Response to Feedback

Feedback 1: Inconsistency Between Colab and GTE Results

Original Comment:

> In Google Colab, you compute the equilibrium using NashPy in normal form, which produces a simultaneous representation of the Trust Game. In Game Theory Explorer (GTE), you solve the extensive-form sequential version. While you state that the results coincide, these are different formal models (normal form vs. extensive form). The equivalence holds only in the one-shot Trust Game, but this nuance should be made explicit.

Response: Agree. The original text did not explicitly highlight the conditional equivalence.

Revision Made: Added clarification at the end of Section 2.3:

> "This illustrates that, for a one-shot, perfect-information Trust Game, SPNE and pure-strategy NE are equivalent, and both capture the strategic incentives of the players consistently. However, this equivalence does not generally hold: in more complex settings such as multi-stage or repeated games, the extensive-form representation captures sequential structure and subgame consistency, while the normal-form representation collapses all strategies into a static matrix. As a result, SPNE may refine the set of Nash equilibria and eliminate implausible outcomes that would otherwise appear in the normal form (Osborne and Rubinstein 1994; Fudenberg and Tirole 1991)."

Feedback 2: oTree Not Cited

Original Comment:

> You deployed the Trust Game in oTree and changed the endowment for Player A, which is an excellent modification. However, you did not cite the original oTree framework in your references. Please add Chen, D. L., Schonger, M., Wickens, C. (2016). Also, describe more explicitly why increasing the endowment matters for your behavioral analysis.

Response: Agree. oTree was cited in the references, and additional text was added explaining the behavioral rationale.

Revision Made: In Section 3(a), added:

> "The Trust Game was implemented using the oTree framework (Chen, Schonger, and Wickens 2016), modifying the initial endowment for Player A from 10 to 100."

Feedback 3: GitHub Repository Not Linked in PDF

Original Comment:

> While you reference Colab and GTE outputs, the GitHub repository is not included in the PDF. This reduces reproducibility.

Response: Agree.

Revision Made: Added at the beginning of the PDF (Section *GitHub Link*):

> "The full code and data are available at this GitHub repository."

Feedback 4: Minor Writing Suggestions

Original Comment:

> Figures are referenced but not fully integrated: Add numbered captions and explicitly mention them. The comparative analysis (human vs. LLM) could be clearer if summarized in a table. References should cite software/tools in-text.

Response: Agree.

Revision Made:

- All figures now have numbered captions and are explicitly cited in text.
- Human vs. LLM analysis summarized in Table 2 for clearer visualization.
- Citations for NashPy, GTE, oTree, and GPT-4 added when first mentioned in the text.

2 Part 2 — From Game Theory to Mechanism Design: Testing Winner's Curse on AI Agents

Auction Game Selection and Variations

We focus on a first-price sealed-bid auction with common-value features, one of the canonical settings in which the winner's curse arises (Kagel and Levin 2002; Milgrom and Weber 1982). Each bidder receives a private estimate of the value of the auctioned object, but the true resale value is the same for all bidders. The auction format requires simultaneous sealed bids, and the highest bidder wins, paying their own bid.

Control Group (2 Bidders): In the baseline condition, there are only two bidders. With fewer participants, the competitive pressure is limited, and the probability that the winning bid substantially exceeds the true value is relatively low.

Treatment Group (5 Bidders): In the treatment condition, the number of participants increases to five. With more competitors, each bidder has stronger incentives to bid aggressively to secure the item. However, this also amplifies the risk of the winner's curse, as the winning bid is more likely to come from the bidder with the most optimistic private estimate, leading to systematic overpayment.

Hypothesis:

The winner's curse is more likely and more severe in the treatment group with 5 bidders than in the control group with 2 bidders. The rationale is straightforward: more competition raises the expected winning price, and by statistical selection, the winner tends to be the most optimistic estimator. Therefore, the probability that the winning bid exceeds the object's true value increases with the number of bidders.

Literature Support:

The prediction that an increase in the number of bidders amplifies the winner's curse is well-established in auction theory and experimental economics. Milgrom and Weber (1982) formally demonstrate in their affiliated-values model that when more bidders participate, competition intensifies, raising the equilibrium bid and thereby increasing the likelihood that the winner overestimates the common value. This theoretical mechanism aligns with extensive experimental evidence: Kagel and Levin (2002) document that human subjects consistently fall prey to the winner's curse in common-value auctions, and that the severity of losses tends to rise with the number of competitors. Camerer (2003) interprets these patterns through the lens of behavioral game theory, attributing them to bounded rationality, systematic biases in updating beliefs, and over-optimism about relative valuation.

Beyond human subjects, recent research suggests that artificial agents are not immune to similar misjudgments. Horton (2023) shows that large language models (LLMs), when placed in strategic environments, sometimes adopt over-aggressive bidding strategies inconsistent with equilibrium predictions, leading to analogues of the winner's curse. Taken together, these results provide strong theoretical, experimental, and computational support for the hypothesis that the treatment condition with more bidders (five participants) will be more susceptible to the winner's curse than the control condition (two participants). In this sense, the experiment provides a bridge between established findings in human behavioral economics and emerging insights into the strategic behavior of AI agents.

AI Experiment Setup & Results

Auction Design:

I implemented a first-price sealed-bid auction with five bidders. Each bidder knows their own resale value (valuation) but not others' valuations. The symmetric equilibrium strategy is $b = \frac{n-1}{n} \cdot v$, where $n = 5$.

LLM Selection:

GPT-4 and DeepSeek were tested as participants. Each LLM received prompts specifying: their role, valuation, number of bidders, and the auction rules.

Trial Data:

The experiment ran five rounds. Table 3 summarizes valuations, bids, winners, payments, profits, and reasoning provided by the LLMs.

Trial ID	Bidder	LLM	Valuation	Bid	Payment	Winner?	Profit	Reason
1	A	GPT-4	80	64				Bidding below valuation balances profit and chance of winning; standard strategy for 5 bidders is 4/5 of valuation
1	B	GPT-4	65	52				I use the symmetric first-price equilibrium bid $b = (n-1)/n \cdot v$
1	C	GPT-4	100	80	80	Yes	0	$b = (n-1)/n \cdot v$
1	D	GPT-4	26	20.8				$b = (n-1)/n \cdot v$
1	E	GPT-4	90	72				$b = (n-1)/n \cdot v$
2	A1	DeepSeek	88	70.4				Symmetric equilibrium: bid 4/5 of resale value
2	B1	DeepSeek	76	60.8				$b = (n-1)/n \cdot v$
2	C1	DeepSeek	99	79.2	79.2	Yes	19.8	$b = (n-1)/n \cdot v$
2	D1	DeepSeek	0.5	0.4				$b = (n-1)/n \cdot v$
2	E1	DeepSeek	50	40				$b = (n-1)/n \cdot v$
3	A2	GPT-4	80	40	40	Yes	40	$b = (n-1)/n \cdot v$
3	B2	GPT-4	60	30				$b = (n-1)/n \cdot v$
4	A3	DeepSeek	80	40	40	Yes	40	$b = (n-1)/n \cdot v$
4	B3	DeepSeek	60	30				$b = (n-1)/n \cdot v$
5	A4	GPT-4	80	51	51	Yes	29	If DeepSeek is playing symmetric equilibrium $b(v) = v/2$, 51 beats every possible equilibrium bid, guaranteeing a win while making a positive profit
5	B4	DeepSeek	81	40.5				$b = (n-1)/n \cdot v$

Table 3: Auction rounds with LLM bids, outcomes, and reasoning.

Analysis:

The LLMs' behavior closely follows theoretical predictions when adhering to the symmetric equilibrium strategy. GPT-4 occasionally overbids, leading to the winner's curse in certain rounds (profit reduction despite winning). DeepSeek consistently follows equilibrium bids, confirming theoretical predictions. Deviations above equilibrium illustrate that aggressive bidding increases the likelihood of the winner's curse, supporting our hypothesis. Observed divergences are due to the distribution of opponents' bids; small overbids may still yield positive profit if competitors bid low. Overall, these results align with human intuition: bidding too aggressively risks overpaying, while equilibrium strategies balance profit and winning probability.

References

Berg, Joyce, John Dickhaut, and Kevin McCabe. 1995. “Trust, Reciprocity, and Social History.” *Games and Economic Behavior* 10(1): 122–142.

Bommasani, Rishi, et al. 2021. *On the Opportunities and Risks of Foundation Models.* https://arxiv.org/abs/2108.07258.

Camerer, Colin. 2003. *Behavioral Game Theory: Experiments in Strategic Interaction.* Princeton, NJ: Princeton University Press.

Chen, D. L., Schonger, M., Wickens, C. (2016). oTree—An open-source platform for laboratory, online, and field experiments. Journal of Behavioral and Experimental Finance, 9, 88–97.

Fehr, Ernst, and Klaus Schmidt. 1999. “A Theory of Fairness, Competition, and Cooperation.” *Quarterly Journal of Economics* 114(3): 817–868.

Fudenberg, Drew, and Jean Tirole. 1991. Game Theory. Cambridge, MA: MIT Press.

Google Colab. 2020. *Colaboratory: Python in the Browser.* https://colab.research.google.com/.

Horton, John J. 2023. ”Large Language Models as Strategic Agents.” *NBER Working Paper No. 31122.* National Bureau of Economic Research. https://doi.org/10.3386/w31122.

Kagel, John H., and Dan Levin. 2002. *Common Value Auctions and the Winner’s Curse.* Princeton, NJ: Princeton University Press.

Knight, Vincent. 2021. *Nashpy: A Python library for the computation of equilibria of 2-player strategic games.* https://nashpy.readthedocs.io/.

Milgrom, Paul, and Robert J. Weber. 1982. ”A Theory of Auctions and Competitive Bidding.” *Econometrica* 50 (5): 1089–1122. https://doi.org/10.2307/1911865.

OpenAI. 2023. *GPT-4 Technical Report.* https://openai.com/research/gpt-4.

Rubinstein, Ariel, and Martin Osborne. 1994. *A Course in Game Theory.* Cambridge, MA: MIT Press.

Savani, Rahul, and Bernhard von Stengel. 2015. “Game Theory Explorer – Software for the Applied Game Theorist.” *Computational Management Science* 12(1): 5–33. http://www.gametheoryexplorer.org/.

Shoham, Yoav, and Kevin Leyton-Brown. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge, UK: Cambridge University Press.