CompEcon

Team Members

Shu Hu (ANU)

Wending Liu (ANU)

Chien Yeh (ANU)

Mentors

Aswin Kumar (NVIDIA)

Joseph John (ANU)

Solving 3D dynamic optimization problems in economics

Agents choose c consumptions and h working hours to optimize the following utility.

$$\sum_{t=0}^{\infty} \beta^t u(c_t, h_t) = \sum_{t=0}^{\infty} \beta^t [\frac{c^{1-\gamma} - 1}{1-\gamma} - \frac{h^a}{a}]$$

where β is discount factor, γ is risk aversion parameter, and $a > 0$.

Given asset A and human capital K, the consumptions and working hours follow the constraints

$$A_{t+1} = (1+r)A_t + RK_t h_t - C_t$$

Solving 3D dynamic

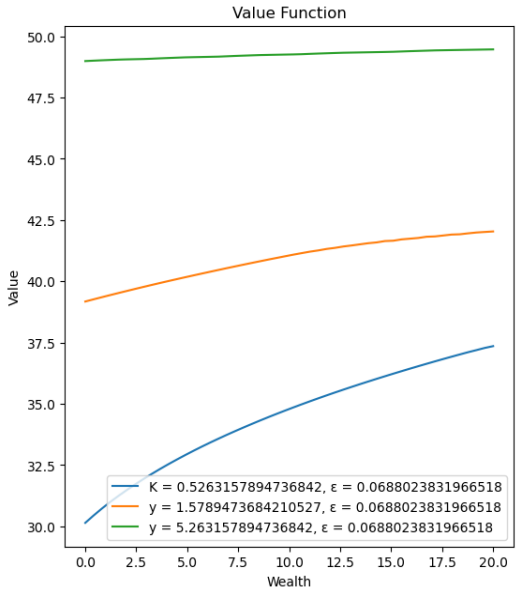
Given R, r we solve

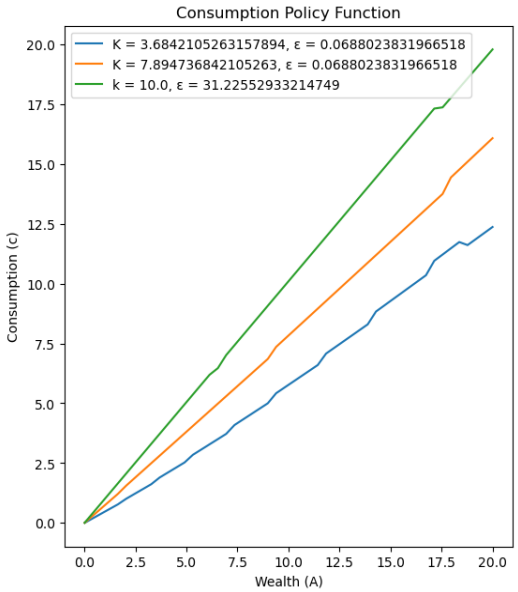
$$V(A_t, K_t, \epsilon_t) = \sup_{h_t, c_t} \left\{ \frac{c^{1-\gamma}}{1-\gamma} - \frac{h^a}{a} + \beta \mathbb{E}_t V(A_{t+1}, K_{t+1}, \epsilon_{t+1}) \right\}$$

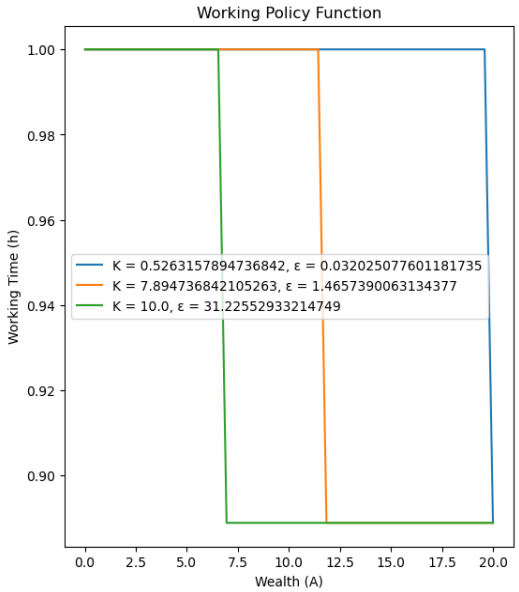
where

$\epsilon_{t+1} = \rho \epsilon_t + \sigma \eta_{t+1}$, where $\eta \sim N(0, \eta)$,
$0 \leq h_t \leq 1$,
$0 \leq A_{t+1} \leq A_t$.

Main result


Value Function
Value
Wealth
K = 0.5263157894736842, ε = 0.0688023831966518
y = 1.5789473684210527, ε = 0.0688023831966518
y = 5.263157894736842, ε = 0.0688023831966518


Consumption Policy Function
Consumption (c)
Wealth (A)
K = 3.6842105263157894, ε = 0.0688023831966518
K = 7.894736842105263, ε = 0.0688023831966518
k = 10.0, ε = 31.22552933214749


Working Policy Function
Working Time (h)
Wealth (A)
K = 0.5263157894736842, ε = 0.032025077601181735
K = 7.894736842105263, ε = 1.4657390063134377
K = 10.0, ε = 31.22552933214749

Our App

- Algorithic Motif: Value Function Iteration on 3D Grids.
- Libraries: NumPy, JAX.
- Language: Python.
- Focus: Parallezing Value Function Iteration (VFI) solver.
- GPU port path: CUDA, JAX.

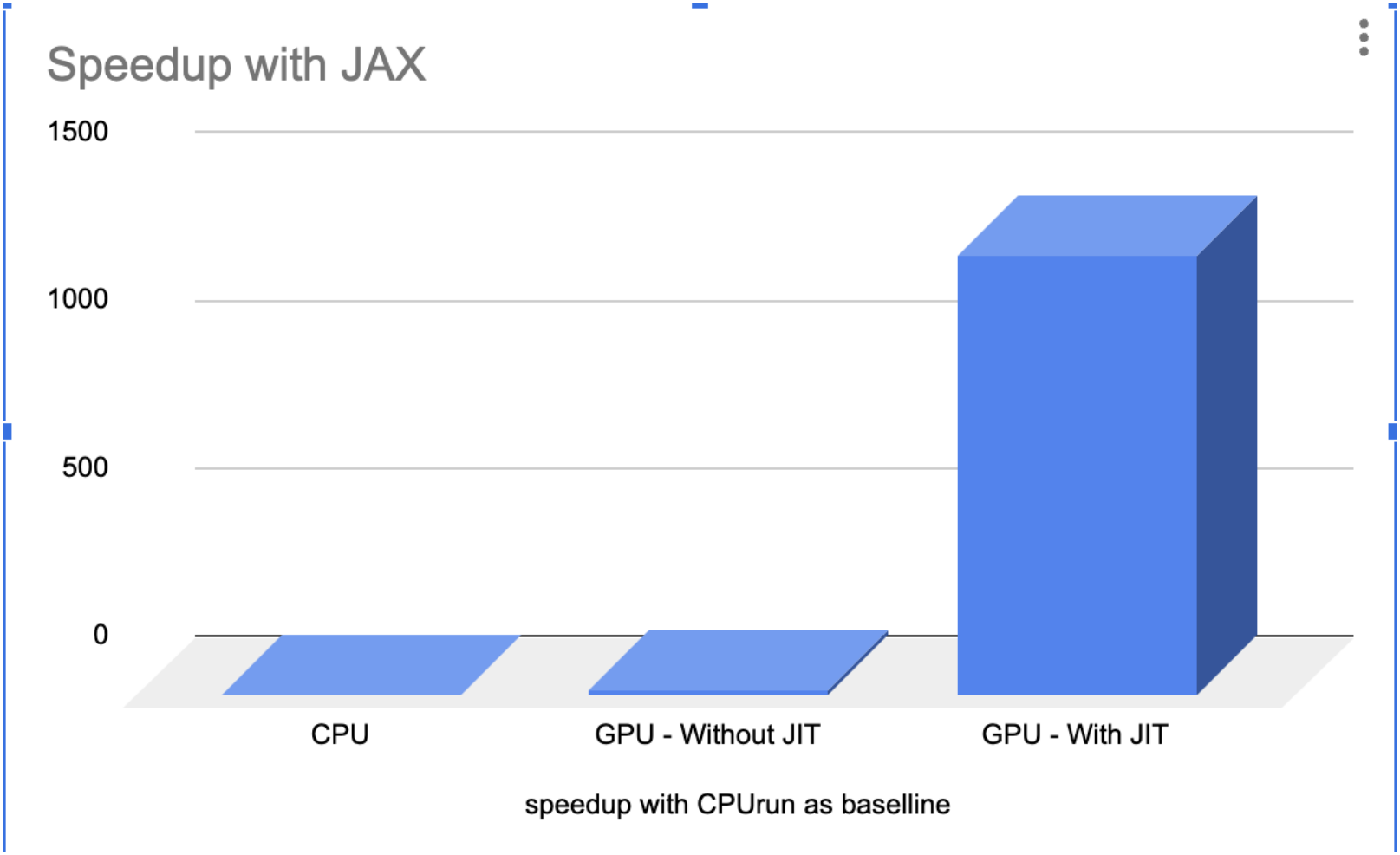
What we did during hackathon

- Finalized the model
- Implemented a new solver using Numpy
- Port the code to Gadi
- Profiling and analyse - nsys profile
- Parallelised the code with `jax`
- Tried to use `jax.pmap`
    - Working only on one thred

Goals

To build an efficient VFI solver which is useful for research in economics.

Performance Results



Energy Utilization

INPUTS	
# CPU Cores	12
# GPUs (A100)	1
Application Speedup	1315.0x
Node Replacement	986.3x

GPU NODE POWER SAVINGS			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (W)	1,084,875	6,500	1,078,375
Networking Power (W)	45,798	93	45,705
Total Power (W)	**1,130,673**	**6,593**	1,124,080

Node Power efficiency	**171.5x**

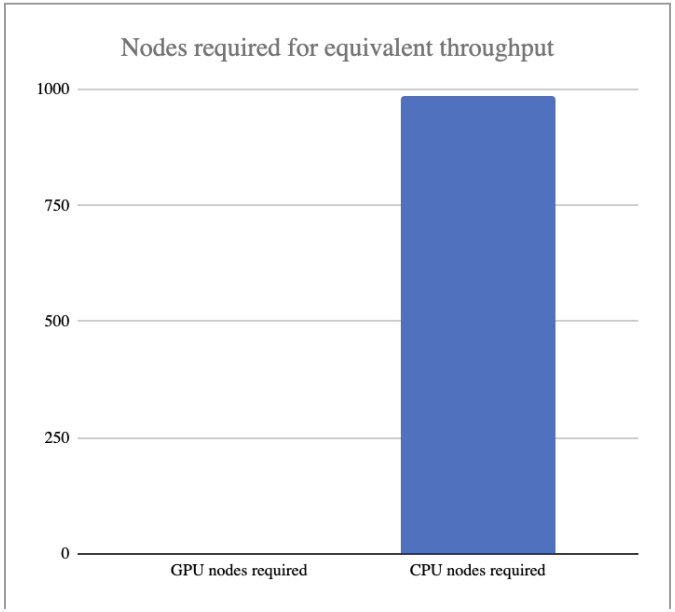
ANNUAL ENERGY SAVINGS PER GPU NODE			
	AMD Dual Rome 7742	8x A100 80GB SXM4	Power Savings
Compute Power (kWh/year)	9,503,505	56,940	9,446,565
Networking Power (kWh/year)	401,192	814	400,379
Total Power (kWh/year)	**9,904,697**	**57,754**	**9,846,944**

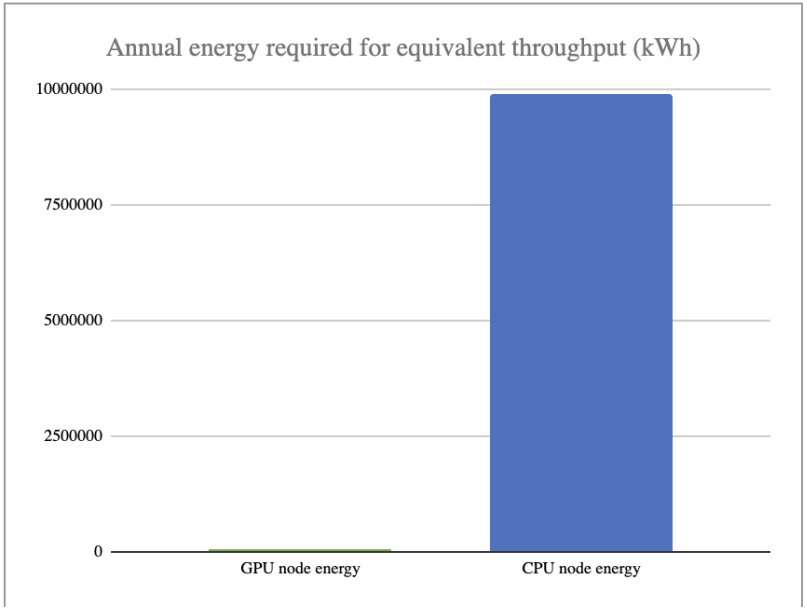
Energy Utilization

$/kWh	$	0.34
Annual Cost Savings	$	3,347,960.91
3-year Cost Savings	$	10,043,882.74

Metric Tons of CO2	6,981
Gasoline Cars Driven for 1 year	1,507
Seedlings Trees grown for 10 years	115,406

Energy Utilization





Energy Utilization

1,507

Gasoline cars driven for a year

115,40

Trees growing for 10 years

6,981

Metric tons of CO2