NewHarmony with Julia

Economic planning software in Julia

Brought to you by: paulcockshott

Paul Cockshott

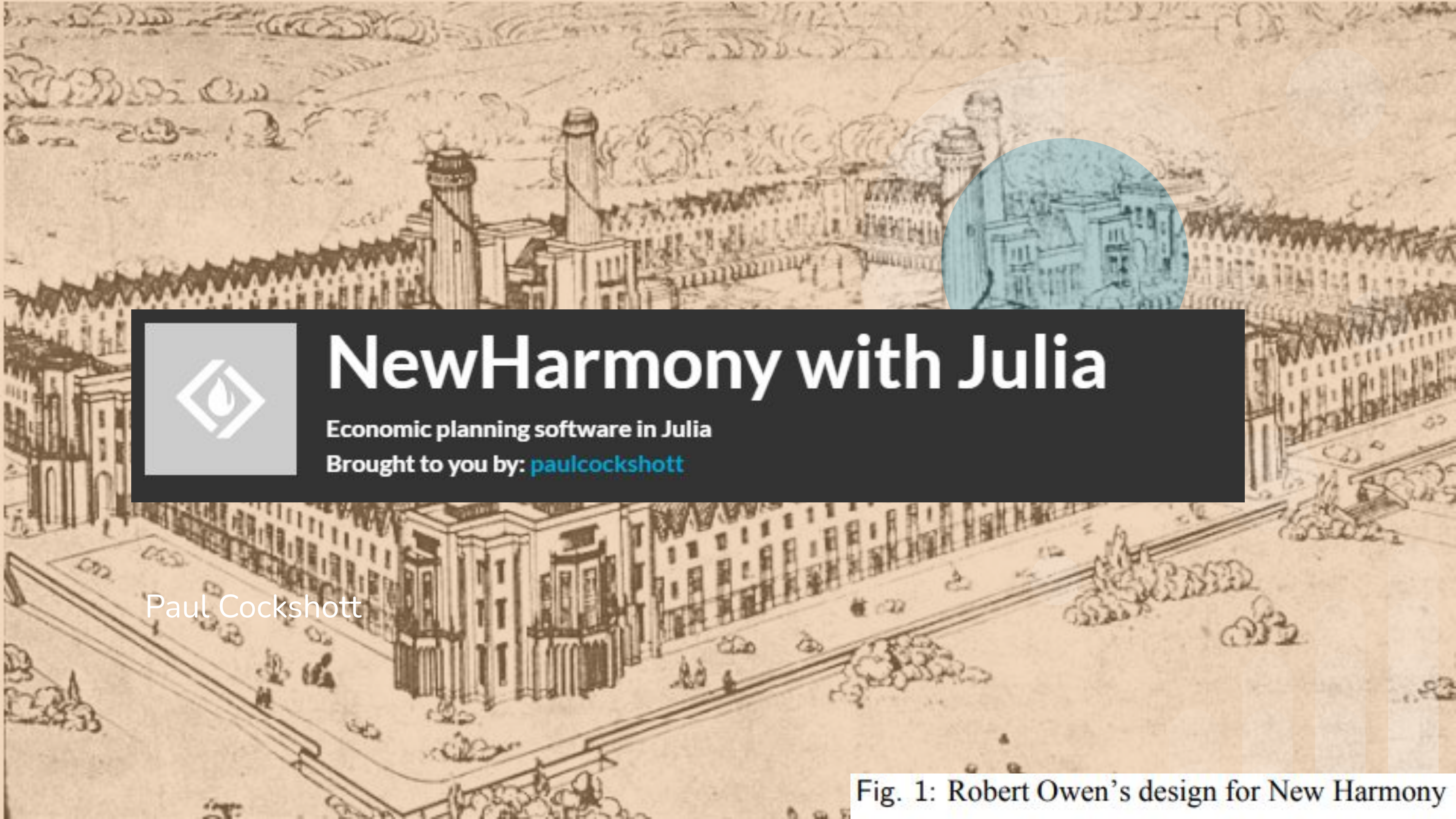


Fig. 1: Robert Owen's design for New Harmony

First release

I have made a first release of planning software in Julia

It is on Source Forge :NewHarmony with Julia download | SourceForge.net

Documentation

A detailed theoretical account of the maths involved is given in a dozen page pdf file.

There are two example programmes

Harmony2.jl	illustrates scheduling capital investment for multi year plans
Csvplan.jl	Combines this with Leontief techniques for within year planning

Leontief techniques

Leontief invented input output analysis of economies

He allowed the planning problem to be treated as a branch of Matrix analysis

Subsequently had a big impact in Marxian economics - see for example work of Morishima

Aggregate IO table EU 2006

	Agriculture + hunting + fishing	Industry incl. Energy	Construct.	Services	Foreign trade
Agriculture + hunting + fishing	45984	173007	2824	33411	13075
Industry incl. Energy	77135	2420959	358832	995556	967607
Construction	2679	42210	338556	206254	2160
Services	64777	1450375	304733	4014520	436944
Foreign trade	10582	635260	38362	278754	117185
Labour power	62593	1183080	386808	4257560	0
Gross surplus	113528	880917	272814	3248484	-9758
total output	381876	6851944	1720275	13286792	1527213

This table was

1. Highly aggregated to start out with
2. I have aggregated it more for demonstration purposes
3. I have regularised the structure into a more Marxian form

Output of agriculture used in industry

	Agriculture + hunting + fishing	Industry incl. Energy	Construct.	Services	Foreign trade
Agriculture + hunting + fishing	45984	173007	2824	33411	13075
Industry incl. Energy	77135	2420959	358832	995556	967607
Construction	2679	42210	338556	206254	2160
Services	64777	1450375	304733	4014520	436944
Foreign trade	10582	635260	38362	278754	117185
Labour power	62593	1183080	386808	4257560	0
Gross surplus	113528	880917	272814	3248484	-9758
total output	381876	6851944	1720275	13286792	1527213

Output of industry used in agriculture

	Agriculture + hunting + fishing	Industry incl. Energy	Construct.	Services	Foreign trade
Agriculture + hunting + fishing	45984	173007	2824	33411	13075
Industry incl. Energy	77135	2420959	358832	995556	967607
Construction	2679	42210	338556	206254	2160
Services	64777	1450375	304733	4014520	436944
Foreign trade	10582	635260	38362	278754	117185
Labour power	62593	1183080	386808	4257560	0
Gross surplus	113528	880917	272814	3248484	-9758
total output	381876	6851944	1720275	13286792	1527213

Foreign imports used in EU industry

	Agriculture + hunting + fishing	Industry incl. Energy	Construct.	Services	Foreign trade
Agriculture + hunting + fishing	45984	173007	2824	33411	13075
Industry incl. Energy	77135	2420959	358832	995556	967607
Construction	2679	42210	338556	206254	2160
Services	64777	1450375	304733	4014520	436944
Foreign trade	10582	635260	38362	278754	117185
Labour power	62593	1183080	386808	4257560	0
Gross surplus	113528	880917	272814	3248484	-9758
total output	381876	6851944	1720275	13286792	1527213

Exports of industrial products

	Agriculture + hunting + fishing	Industry incl. Energy	Construct.	Services	Foreign trade
Agriculture + hunting + fishing	45984	173007	2824	33411	13075
Industry incl. Energy	77135	2420959	358832	995556	967607
Construction	2679	42210	338556	206254	2160
Services	64777	1450375	304733	4014520	436944
Foreign trade	10582	635260	38362	278754	117185
Labour power	62593	1183080	386808	4257560	0
Gross surplus	113528	880917	272814	3248484	-9758
total output	381876	6851944	1720275	13286792	1527213

The 'A matrix'

$$A = \begin{bmatrix} 0.1204 & 0.0252 & 0.0016 & 0.0025 & 0.0086 \\ 0.2020 & 0.3533 & 0.2086 & 0.0749 & 0.6336 \\ 0.0070 & 0.0062 & 0.1968 & 0.0155 & 0.0014 \\ 0.1696 & 0.2117 & 0.1771 & 0.3021 & 0.2861 \\ 0.0277 & 0.0927 & 0.0223 & 0.0210 & 0.0767 \\ 0.1639 & 0.1727 & 0.2249 & 0.3204 & 0.0000 \end{bmatrix}$$

Laid out same way as io table

Each cell A_{ij} says what fraction of a Euro of i was used to make one Euro worth of j

Using the A matrix

Suppose we specify the the gross output as a column vector o, then the Matrix vector product $A.o$ evaluates to the vector of intermediate products needed to make output mix o. We will define this vector of resources as r such that

$$r = A.o$$

Illustration using Julia

```
A=[
0.1204 0.0252 0.0016 0.0025 0.0086;
0.2020 0.3533 0.2086 0.0749 0.6336;
0.0070 0.0062 0.1968 0.0155 0.0014;
0.1696 0.2117 0.1771 0.3021 0.2861;
0.0277 0.0927 0.0223 0.0210 0.0767;
0.1639 0.1727 0.2249 0.3204 0.0000]
o=[
381876
6851944
1720275
13286792
1527213
]
display(A*o)
```

This is the actual gross output in 2006

Illustration using Julia

Matches the actual intermediate products used

```
A=[
0.1204 0.0252 0.0016 0.0025 0.0086;
0.2020 0.3533 0.2086 0.0749 0.6336;
0.0070 0.0062 0.1968 0.0155 0.0014;
0.1696 0.2117 0.1771 0.3021 0.2861;
0.0277 0.0927 0.0223 0.0210 0.0767;
0.1639 0.1727 0.2249 0.3204 0.0000]
o=[
381876
6851944
1720275
13286792
1527213
]
display(A*o)
```

```
6-element Array{Float64,1}:
 267750.31100000005
      4.8196030098e6
 591788.679
      6.2708589193999999e6
      1.0802751756e6
      5.8898982095e6

------------------
(program exited with code: 0)

Press any key to continue . . .
```

Final consumption?

we can also work out how much will be available for final consumption (f) for given a particular vector of gross output:

$$f = o - A.o$$

Lets try this in Julia

```
f= o - A*o
display(f)
```

```
ERROR: LoadError: DimensionMismatch("dimensions must match: a has dims (Base.OneTo(5),), b has dims (Base.OneTo(6),), mi
smatch at 1")
Stacktrace:
 [1] promote_shape at .\indices.jl:178 [inlined]
 [2] promote_shape at .\indices.jl:169 [inlined]
 [3] -(::Array{Int64,1}, ::Array{Float64,1}) at .\arraymath.jl:38
 [4] top-level scope at E:\OneDrive\src\ioutopia\demo.jl:16
 [5] include(::Function, ::Module, ::String) at .\Base.jl:380
 [6] include(::Module, ::String) at .\Base.jl:368
 [7] exec_options(::Base.JLOptions) at .\client.jl:296
 [8] _start() at .\client.jl:506
in expression starting at E:\OneDrive\src\ioutopia\demo.jl:16


------------------
(program exited with code: 1)

Press any key to continue . . .
```

Lets try this in Julia

What is wrong?

```
f= o - A*o
display(f)
```

```
ERROR: LoadError: DimensionMismatch("dimensions must match: a has dims (Base.OneTo(5),), b has dims (Base.OneTo(6),), mi
smatch at 1")
Stacktrace:
 [1] promote_shape at .\indices.jl:178 [inlined]
 [2] promote_shape at .\indices.jl:169 [inlined]
 [3] -(::Array{Int64,1}, ::Array{Float64,1}) at .\arraymath.jl:38
 [4] top-level scope at E:\OneDrive\src\ioutopia\demo.jl:16
 [5] include(::Function, ::Module, ::String) at .\Base.jl:380
 [6] include(::Module, ::String) at .\Base.jl:368
 [7] exec_options(::Base.JLOptions) at .\client.jl:296
 [8] _start() at .\client.jl:506
in expression starting at E:\OneDrive\src\ioutopia\demo.jl:16


------------------
(program exited with code: 1)

Press any key to continue . . .
```

Our A matrix was not square

```
A=[
0.1204 0.0252 0.0016 0.0025 0.0086;
0.2020 0.3533 0.2086 0.0749 0.6336;
0.0070 0.0062 0.1968 0.0155 0.0014;
0.1696 0.2117 0.1771 0.3021 0.2861;
0.0277 0.0927 0.0223 0.0210 0.0767;
0.1639 0.1727 0.2249 0.3204 0.0000]
```

6 rows

Only 5 columns

Our A matrix was not square

```
A=[
0.1204 0.0252 0.0016 0.0025 0.0086;
0.2020 0.3533 0.2086 0.0749 0.6336;
0.0070 0.0062 0.1968 0.0155 0.0014;
0.1696 0.2117 0.1771 0.3021 0.2861;
0.0277 0.0927 0.0223 0.0210 0.0767;
0.1639 0.1727 0.2249 0.3204 0.0000]
```

6 rows

We have a labour input row but no labour output column
Because this is capitalism not slavery, no industry produces labour power

Only 5 columns

Create a dummy labour power column of zeros

```
A=[ 0.1204 0.0252 0.0016 0.0025 0.0086 0;
    0.2020 0.3533 0.2086 0.0749 0.6336 0;
    0.0070 0.0062 0.1968 0.0155 0.0014 0;
    0.1696 0.2117 0.1771 0.3021 0.2861 0;
    0.0277 0.0927 0.0223 0.0210 0.0767 0;
    0.1639 0.1727 0.2249 0.3204 0.0000 0]
o=[381876
6851944
1720275
13286792
1527213
0]
```

Zero gross production of labour power by the capitalist sector

Try this

```
f=o-A*o

display(f)
```

```
6-element Array{Float64,1}:
 114125.68899999995
      2.0323409902e6
      1.128486321e6
      7.015933080600001e6
 446937.82440000004
     -5.8898982095e6

------------------
(program exited with code: 0)

Press any key to continue . . .
```

Positive consumption of commodities by the domestic sector

Negative final consumption of labour power by the domestic economy!

Working backwards

Given that $o = A.o + f$ thus $o – A.o = f$ and then taking o as a common factor

$(I – A).o = f$

Where I is the identity matrix. Dividing through by (I – A) we get

$$o = (I - A)^{-1} . f$$

We can derive the gross output required for any given pattern of final consumption!

In Julia

```
f[6]=0 # remove the negative labour
display(( I -A)\ f)
```

```
6-element Array{Float64,1}:
 381875.99999999994
      6.851944e6
      1.720275e6
      1.3286792000000002e7
      1.5272130000000002e6
      5.8898982095e6

------------------
(program exited with code: 0)

Press any key to continue . . .
```

To within
rounding errors
Matches original o

```
o=[
381876
6851944
1720275
13286792
1527213
]
```

This is all standard

This is all standard Marxian/ Leontief economics since the 1950s

The next problem is how to link up a sequence of years to form a single 5 year plan

I will cover how I did that in the next video, this is already enough math for one video

The only thing to note is that Julia, like Octave or Euler, makes it very easy to express the calculations

The time dimension

Planning with Julia

Paul Cockshott

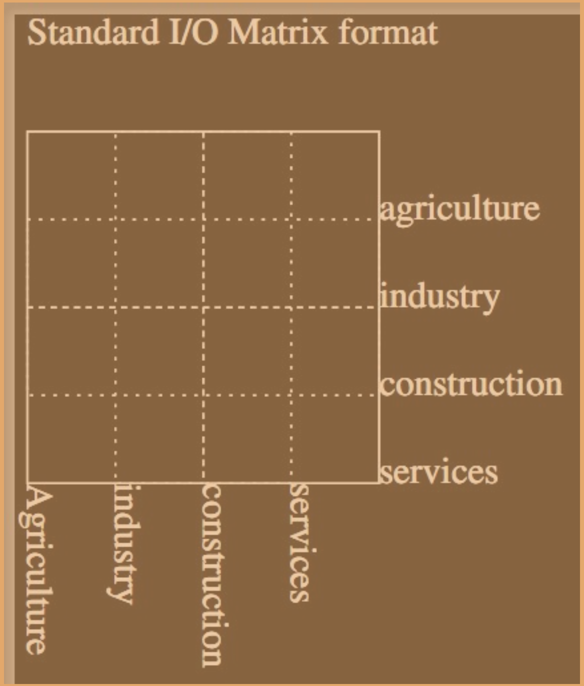
Response to comments

Cant download ?: *To download from sourceforge use subversion to do a checkout*

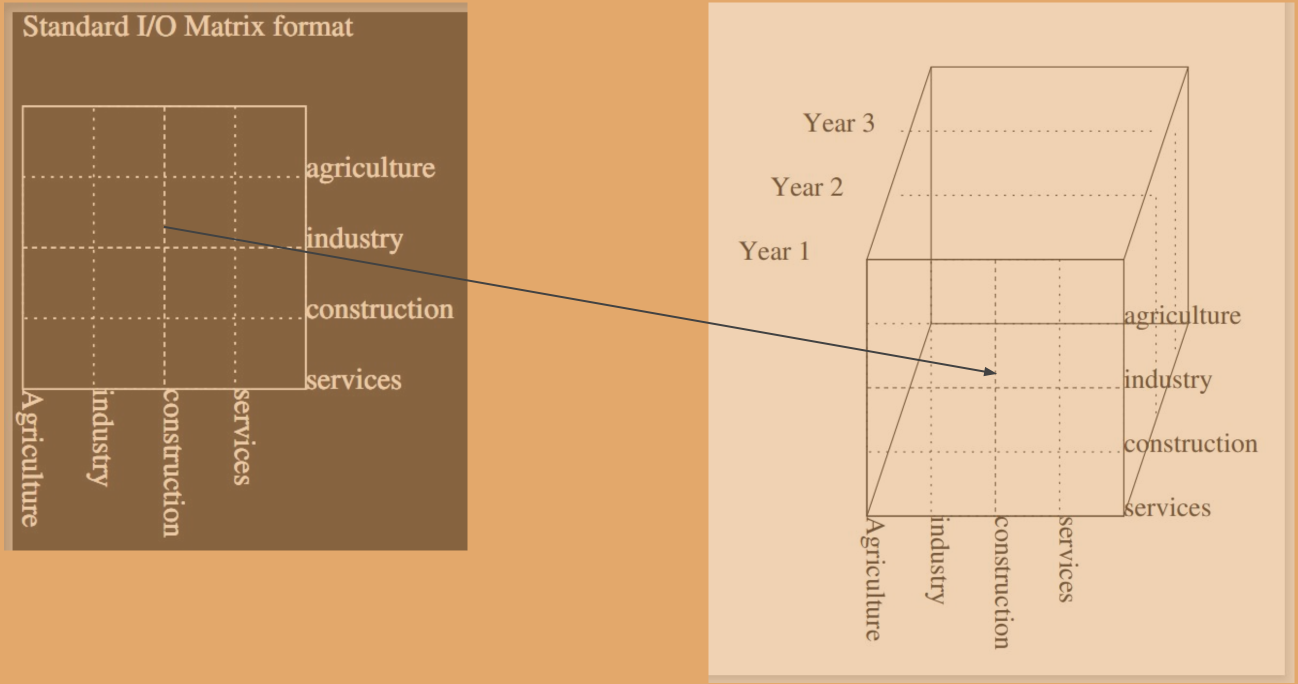
Why not Github ?: *I have also uploaded to github* *wc22m/NewHarmonywithJulia (github.com)*

Is this the same as the Jack Ma article ?: *Not identical, in that the within year approach is now Leontief based, the between year approach remains the same as in that article*

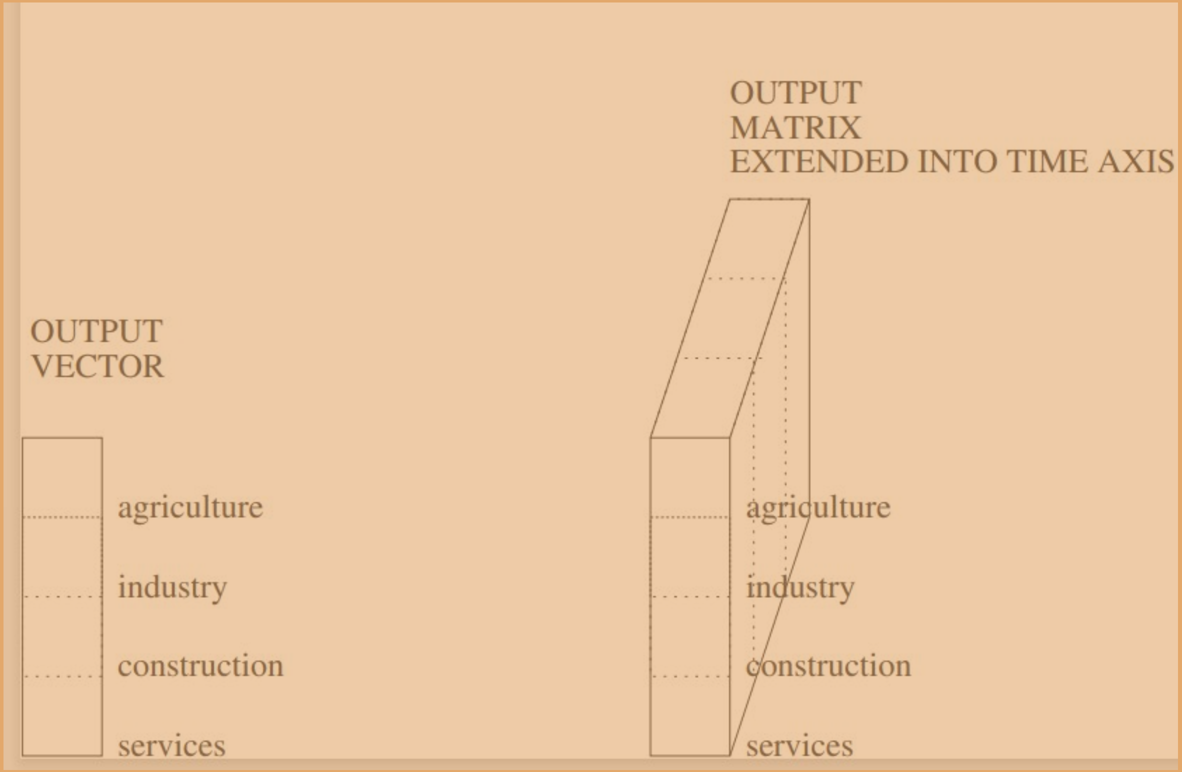
Matrix format used so far



Tensor format with time axis



Output vector becomes a matrix



The main tensors

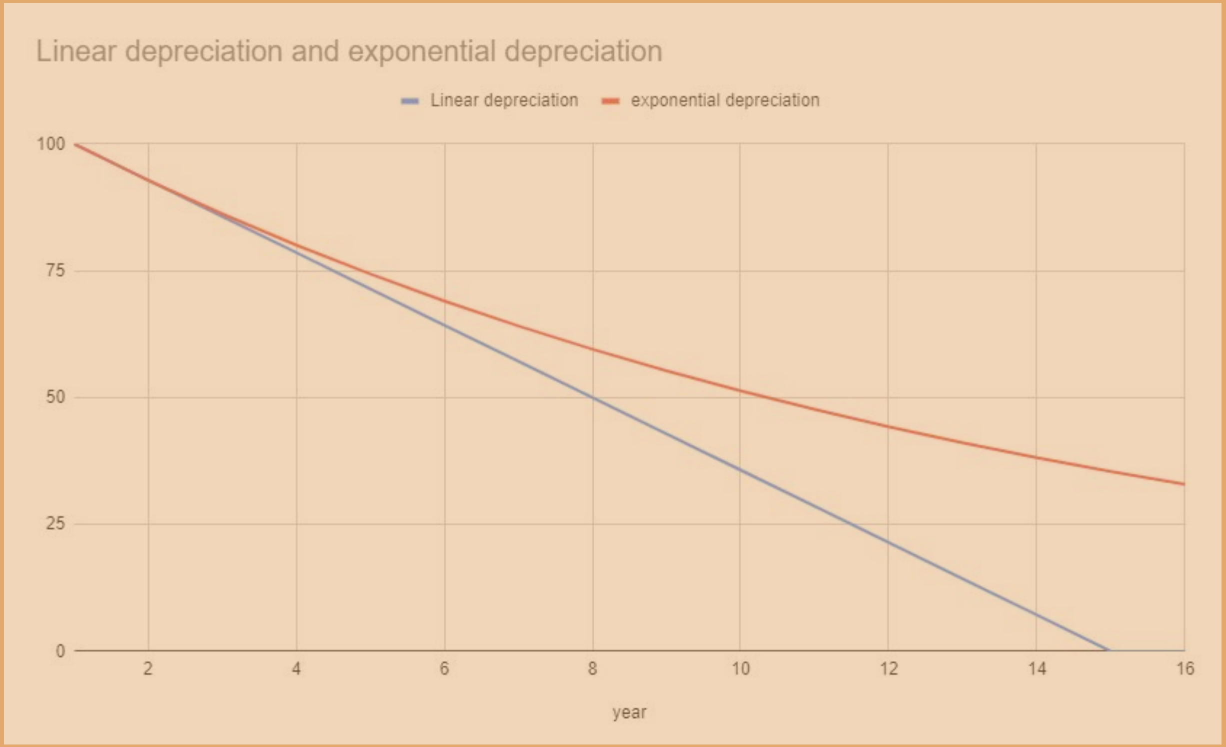
1. The capital stock tensor, indexed by year, product, product
    - For any given year this is matrix of stocks year by year
2. The investment tensor, indexed by year, product, product

Relation between them

The capital stock tensor is the integral of the investment tensor convolved with a depreciation function

That is to say

The stock in year n is the sum of investment years 1..n weighted by depreciation



Two ways of computing depreciation
Harmony2.jl uses linear, csvplan.jl uses exponential

Inter-temporal costs

If society invests in stocks of means of production this year, it increases its future consumption potential at the expense of this year’s consumption.

How do you balance these?

Linear or non-linear?

The simplest solution is to just schedule investment so that the total consumption over N years is maximised.

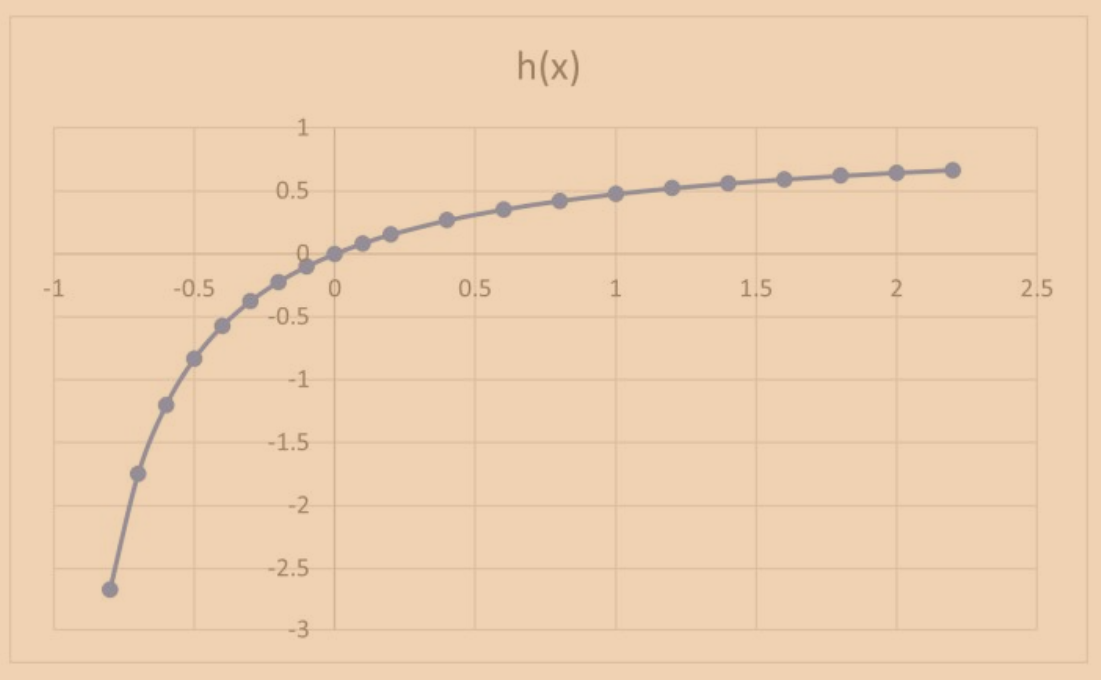
But this can lead to cuts in consumption in the early years that might be unacceptable.

You will find that this happens if you use the Kantorovich style planner nyearplan.java release in my earlier GitHub project (GitHub - wc22m/5yearplan: Software to construct socialist 5 year plan , https://github.com/wc22m/5yearplan)

This famously happened in soviet industrialisation

Real consumption from Allen, Child Growth from my calculations

Non linear weighting



H(x) = x/(1.1 + x)

Where *x* is the ratio of output achievable in year *n* to output desired

Better than old harmony function

I used to use a different harmony function in TNS, but my co-worker Allin pointed out to me that the old harmony function did not have a continuous first derivative.

This can lead to instability in algorithms preparing the plans

The new one does have a continuous first derivative

Harmony for a year

The aim of the plan algorithm is to maximise the sum of the harmonies over all the years of the plan.

How do you compute the harmony for a year as opposed to a product in a year?

Two possible ways

- take mean of all industries harmonies
- Take the lowest harmony of all the industries in that year

Second approach seems better, I find that the first can lead to excessive cuts in consumption of individual products.

No Future?

Problem of short time periods is that if you optimise over 5 years, the harmony maximisation will assume that no investment is needed to support years 6,7,8 etc

Avoid this by automatically adding more years to the plan, assuming stable goals and stable labour available for all subsequent years.

Number of extra years is set by a global variable ‘depreciationhorizon’. I set this to 14 as that is a number often used for the depreciation in national accounts.

Convergence criteria

Algorithm looks at years of below average harmony and determines how much investment would be needed to bring them up to average harmony call this accumulation *Ä*

It then scales this by some small fraction ε (to prevent adjustment overshooting) and looks at which of the preceding years will produce the biggest increase in overall gain if accumulation ε*Ä* is scheduled that year

Convergence terminates either when

- A max number of attempts has been made
- The coefficient of variation of harmony has fallen below a threshold
- No possible accumulation raising overall harmony can be found

Input data

Driven by

- IO flow table - basically a standard io table as shown in first video
- A capital stock table - you will have to do some research into published data to estimate this, fair bit of work constructing real examples
- A depreciation rate table, laid out as the other ones
- A Labtargs file

Layout of the labtargs csv file

	A	B	C	D	E	F	G
1	Year	Agricultur	Industry	Constructi	Services	foreigntra	Labour
2	1	101764.9	1469520	48082	6559004	283372.4	5890041
3	2	101997.8	1470520	49082	6559004	283372.4	5990336
4	3	102230.6	1471520	50082	6559004	283372.4	6047519
5	4	102696.4	1472520	51082	6559004	283372.4	6075071
6	5	102463.5	1473520	52082	6559004	283372.4	6061295
7							

Labour available year 2

	A	B	C	D	E	F	G
1	Year	Agricultur	Industry	Constructi	Services	foreigntra	Labour
2	1	101764.9	1469520	48082	6559004	283372.4	5890041
3	2	101997.8	1470520	49082	6559004	283372.4	5990336
4	3	102230.6	1471520	50082	6559004	283372.4	6047519
5	4	102696.4	1472520	51082	6559004	283372.4	6075071
6	5	102463.5	1473520	52082	6559004	283372.4	6061295
7							

Target final consumption agriculture year 4

Year	Agricultur	Industry	Constructi	Services	foreigntra	Labour
1	101764.9	1469520	48082	6559004	283372.4	5890041
2	101997.8	1470520	49082	6559004	283372.4	5990336
3	102230.5	1471520	50082	6559004	283372.4	6047519
4	102696.4	1472520	51082	6559004	283372.4	6075071
5	102463.5	1473520	52082	6559004	283372.4	6061295