Beanstalk State Space

understanding the state (*i.e.*, health) of beanstalk and the peg maintenance model that responds to it

“a picture is worth a thousand words.” - unknown

Ben Weintraub, Brendan Sanderson, Guy, Brean

February 1st, 2024 — Post Class Version

Peg Maintenance

at the beginning of each Season, Beanstalk adjusts various parameters within itself as a response to its current state in an attempt to regularly oscillate the Bean price over its value peg (V) indefinitely.

therefore, the two primary questions with respect to peg maintenance are:

(1) how should Beanstalk classify its state at the beginning of each Season?

(2) how should Beanstalk respond to its state at the beginning of each Season in an attempt to return to its ideal state?

(1) how should Beanstalk classify its state at the beginning of each Season?

Classifying State

Some key questions to answer in determining what the Beanstalk state is are:

(1.1) what are the relevant axes/dimensions on/in which to evaluate Beanstalk‘s state?

(1.2) which data to use to evaluate Beanstalk’s current position along each axis?

(1.3) what are the features of each relevant axis (*i.e.*, continuous or discrete, shape, scale)?

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

(1.5) how should Beanstalk evaluate its position, direction and acceleration at the beginning of each Season in practice?

(1.1) what are the relevant axes/dimensions on/in which to evaluate Beanstalk‘s state?

State Intuition

Beanstalk is a stablecoin protocol. The main goal of Beanstalk is to regularly oscillate the Bean price across V, indefinitely. Furthermore, non collateralized stablecoins are highly reflexive with respect to price: demand for them tends to grow when $P > V$ and shrink when $P < V$. Therefore, some notion of PRICE should be considered as part of the Beanstalk state.

Beanstalk’s primary peg maintenance mechanism, and theoretical basis for existence, is the Field, the Beanstalk credit facility. The ability for Beanstalk to borrow Beans from the open market at a reasonable interest rate is essential for long term peg maintenance. Because the level of indebtedness is an essential, universal metric in determining the health of a debt based system, some notion of DEBT LEVEL should be considered as part of the Beanstalk state.

With the successful addition to Beanstalk of Conversions within the Silo in December ‘21, a second peg maintenance mechanism was discovered. The ability to add and remove Beans from liquidity pools to reduce and increase the Bean price, respectively, changes the Bean price without any outflows or inflows from the system. Conversions reallocate liquidity along AMM pricing curves. accordingly, some notion of LIQUIDITY LEVEL should be considered as part of the Beanstalk state.

(1.1) what are the relevant axes/dimensions on/in which to evaluate Beanstalk‘s state?

State Intuition pt 2

There are no other axes that seem to be as important to the state and health of Beanstalk than PRICE, DEBT LEVEL and LIQUIDITY LEVEL

Therefore, the primary evaluation of Beanstalk’s state will be occur along these three axes.

Our goal is to understand the implications of optimizing for long term stability along these three axes. By better understanding the problem we can better tailor its solution.

3 axes for primary state

- PRICE
- DEBT LEVEL
- LIQUIDITY LEVEL

(1.2) which data to use to evaluate Beanstalk’s current position along each axis?

Price

There are a variety of potential options for Beanstalk to evaluate its state with respect to PRICE. For the purposes of this exercise multi-block MEV manipulation resistance of values is not treated thoroughly. Separate discussion to ensure multi-block MEV resistance of each input to the peg maintenance mechanism is warranted.

(1) price

(2) deltaB (i.e., the number of Beans that would have to be bought or sold to return the Bean price to V)

(3) deltaV (i.e., the number of V that would have to be bought or sold to return the Bean price to V)

Any of these values could be evaluated as an average over the previous Season or at the start of the Season using a multi-block MEV manipulation resistant oracle.

While both deltaB and deltaV are highly correlated with Price, both can be different values even if there is no change in price (i.e., via a change in the amount of liquidity). Therefore, it is better to simply use the price as the dimension over which to evaluate Beanstalk‘s PRICE.

Beanstalk‘s response to its state may (and, in fact, does) take into account deltaB or deltaV (e.g., soil issuance) in practice. However, price is the best dimension over which to evaluate Beanstalk‘s position with respect to PRICE to understand Beanstalk in theory.

(1.2) which data to use to evaluate Beanstalk’s current position along each axis?

Debt Level

There are a variety of potential options for Beanstalk to evaluate its state with respect to DEBT LEVEL, including:

(1) debt to supply ratio (D^R)

(2) total amount of outstanding debt, denominated in Beans

(3) total amount of outstanding debt, denominated in V

Intuitively, a larger system can handle more debt. Therefore, DEBT LEVEL is dependent on the Bean supply. This makes the debt to supply ratio preferred over total outstanding debt.

Using the ratio of debt to new Beans minted over a period of time may provide meaningful data for a future model of Beanstalk’s peg maintenance model. Beanstalk’s peg maintenance model has always been designed with simplicity in mind, and additional complexity has only been added after both the problem and the benefits of the proposed solution were deeply understood. Properly handling growth over time with respect to debt (and supply) is a complex problem that should be given consideration for future study. Therefore, Beanstalk currently uses the debt to supply ratio as its indicator for health with respect to DEBT LEVEL.

Note, the current implementation uses the Pod Rate instead of D^R. It was decided not to implement the change to the debt level calculation to include outstanding Fertilizer because in practice the Pod Rate is so high at the moment that all outstanding Fertilizer will be paid off before a substantive change in the Pod Rate.

(1.2) which data to use to evaluate Beanstalk's current position along each axis?

Liquidity Level - New Addition To Beanstalk State with the New Seed Gauge System

There are a variety of potential options for Beanstalk to evaluate its state with respect to LIQUIDITY LEVEL, including:

(1) liquidity to supply ratio (L2SR)

(2) total amount of liquidity, denominated in Beans

(3) total amount of liquidity, denominated in V

Intuitively, a larger system requires more liquidity to be similarly healthy. Therefore, LIQUIDITY LEVEL is dependent on the Bean supply. This makes the L2SR preferred over total amount of liquidity. The total liabilities of the Bean supply is denominated in V. Therefore, Beanstalk uses the L2SR with liquidity denominated in V as its indicator for health with respect to LIQUIDITY LEVEL.

Note, the L2SR will begin to be used by Beanstalk for peg maintenance with the implementation of the gauge system.

(1.2) which data to use to evaluate Beanstalk's current position along each axis?

Liquidity Level pt 2

Other important components of measuring liquidity levels are:

(1) distribution of liquidity

(2) distribution of liabilities (*i.e.*, in a Beanstalk implementation with multiple V, the portion of the intended value of all Beans with different V).

the former does not effect peg maintenance at any given time (only over time). Therefore, while it may be used by Beanstalk as part of the LP gauge system, there is no need to consider liquidity distribution in evaluating its overall state.

the latter only needs to be considered in Beanstalk supports multiple V, which it currently does not. furthermore, it does not effect peg maintenance at any given time (only over time). Therefore, while it may be used by Beanstalk as part of a future liability gauge system, there is no need to consider liability distribution in evaluating its overall state.

the former will be treated lightly herein. the latter will not be treated.

(1.1) what are the relevant axes/dimensions on/in which to evaluate Beanstalk‘s state?

(1.2) which data to use to evaluate Beanstalk’s current position along each axis?

Answering questions 1.1 and 1.2

In addition to PRICE, Beanstalk should evaluate its overall state in terms of DEBT LEVEL and LIQUIDITY LEVEL.

PRICE should be evaluated along the dimension of price.

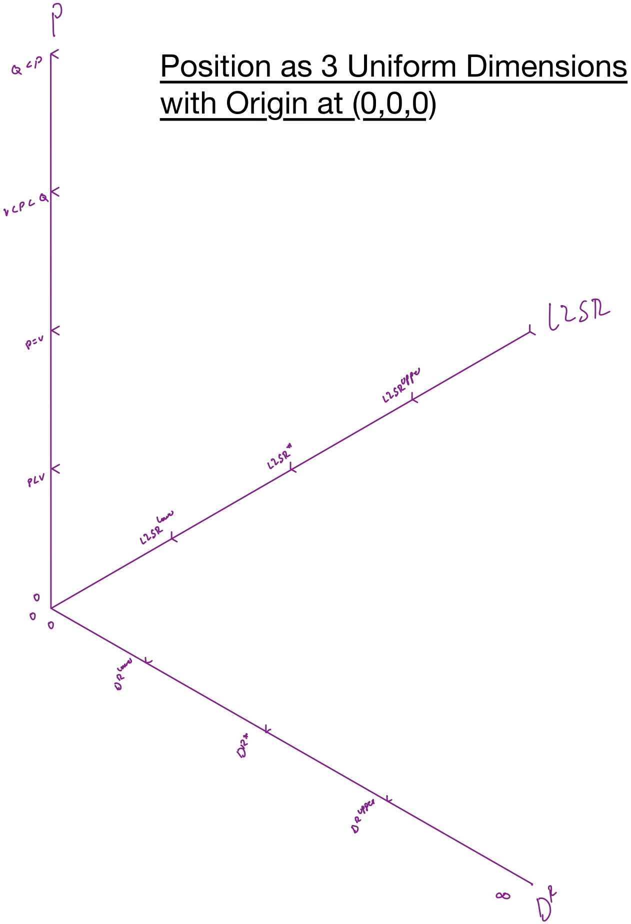
DEBT LEVEL should be evaluated along the dimension of debt to supply ratio.

LIQUIDITY LEVEL should be evaluated along the dimension of liquidity to supply ratio.

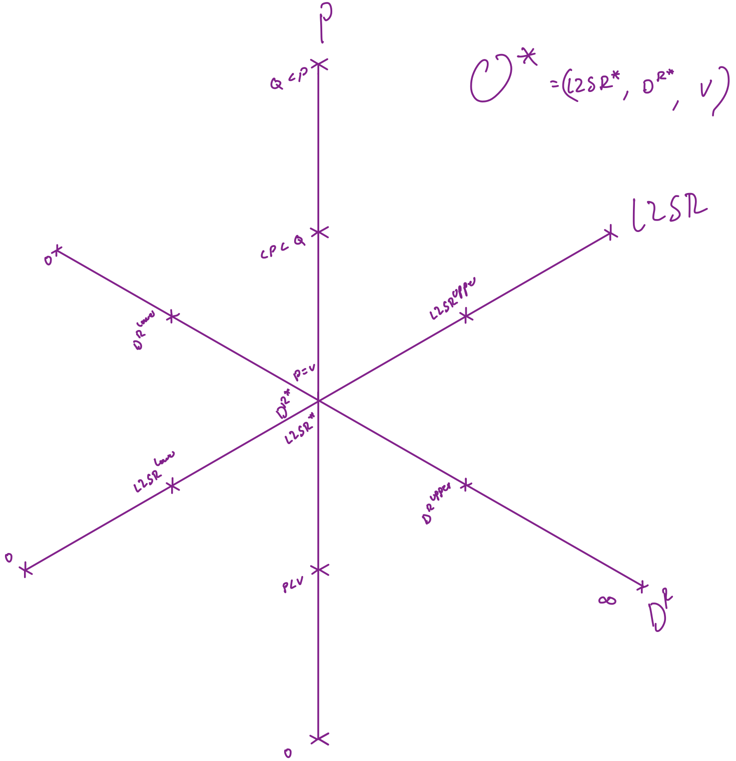
(1.1) what are the relevant axes/dimensions on/in which to evaluate Beanstalk‘s state?

(1.2) which data to use to evaluate Beanstalk’s current position along each axis?

Position as 3 Uniform Dimensions with Origin at (0,0,0)



Position as 3 Uniform Dimensions with Origin at Ideal Equilibrium



$O^* = (L2SR^*, DR^*, V)$

(1.3) what are the features of each relevant axis (*i.e.*, continuous or discrete, shape, scale)?

Understanding each axis

Each axis can be evaluated independently of the others. Through independent evaluation of each axis, one can reach a deeper understanding of it. The axes should only be evaluated together after careful consideration of each axis independently.

The qualities of each axis can be evaluated in terms of:

(1) discrete or continuous

(2) shape (*i.e.*, force, reflexivity)

(3) scale (*i.e.*, friction, volatility)

Discrete or Continuous

should beanstalk's evaluation of position on an axis change discretely or continuously

(1.3) what are the features of each relevant axis (*i.e.*, continuous or discrete, shape, scale)?

shape and scale can be thought of as two halves of the same whole. as reflexivity is to volatility and force is to friction.

reflexivity is the force of the push and pull of the position on itself relative to some point. volatility is the amount of ground covered given an amount of force (i.e., friction).

reflexivity can be understood as a force, whereas volatility is neutral in terms of force. likewise reflexivity can be understood as neutral in terms of friction, whereas volatility effects friction.

an understanding of the interplay between reflexivity and volatility can be used to construct an understanding of the shape and scale of each axis in the Beanstalk state space.

from a visual perspective, imagine balancing a ball on a surface. the shape and scale of the surface (axis) determines the ease with which Beanstalk can return the value to its optimal value by applying force to the ball.

(1.3) what are the features of each relevant axis (i.e., continuous or discrete, shape, scale)?

<u>Shape (i.e., force, reflexivity)</u>

the position of Beanstalk along each axis exhibits force on itself. the shape of each axis is determined by the marginal change in force given a marginal change in position.

1st derivative of position: slope of the axis is inversely correlated with force. the steeper the angle the greater the force.

- positive slope -> downward force
- no slope -> no force
- negative slope -> upward force

2nd derivative: the degree of curvature of the axis is correlated with the force's effect on itself.

- increasing slope -> more downward force -> convex
- constant slope -> same force -> neither convex not concave
- decreasing slope -> more upward force -> concave

3rd derivative: rate of change of reflexivity determines rate of change of degree of curvature.

- accelerating slope -> accelerating change in force -> increasing convexity/concavity
- constant change in slope -> constant change in force -> constant convexity/concavity
- decelerating slope -> decelerating change in force -> decreasing convexity/concavity

(1.3) what are the features of each relevant axis (*i.e.*, continuous or discrete, shape, scale)?

Scale (*i.e.*, friction, volatility)

the position of Beanstalk along each axis changes more quickly at some positions vs others. the volatility of position can be understood as the inverse of friction. if slope is an indication of the force of position on itself, friction is an indication of the effect of a given force on position (*i.e.*, how *slippery* the axis is). the scale of each axis is determined by the marginal change in friction given a marginal change in value.

1st derivative of position: changes in scale of the axis are inversely correlated with changes in friction

- expansion -> decreasing volatility
- no expansion/compression -> constant volatility
- compression -> increasing volatility

2nd derivative: magnitude of changes in scale of the axis are positively correlated with volatility

- more expansion/compression -> acceleration of change in volatility
- constant expansion/compression -> constant change in volatility
- less expansion/compression -> deceleration of change in volatility

3rd derivative: rate of change of magnitude of expansion/compression. not as helpful to evaluate as the 1st and 2nd derivatives relative to complexity at this time.

Visualizing the shape & scale

of each axis in

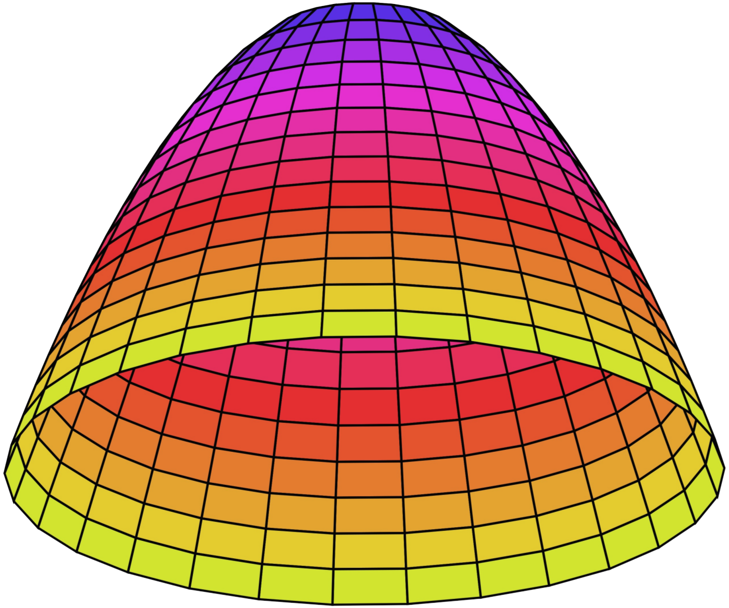
1D & 2D

Visualizing Beanstalk State Space

Imagine trying to balance a ball on a bowl with a leafblower.

The bowl represents the reflexivity of Beanstalk's state space.

The leafblower represents Beanstalk's peg maintenance tools.

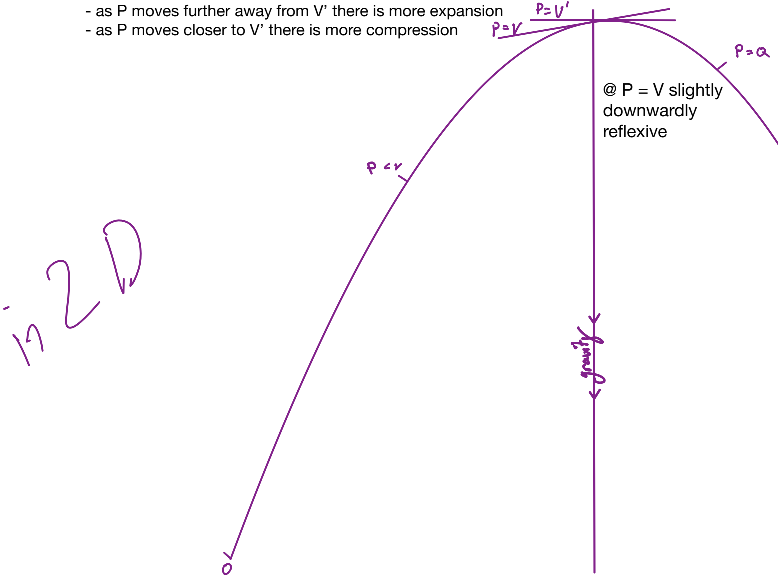


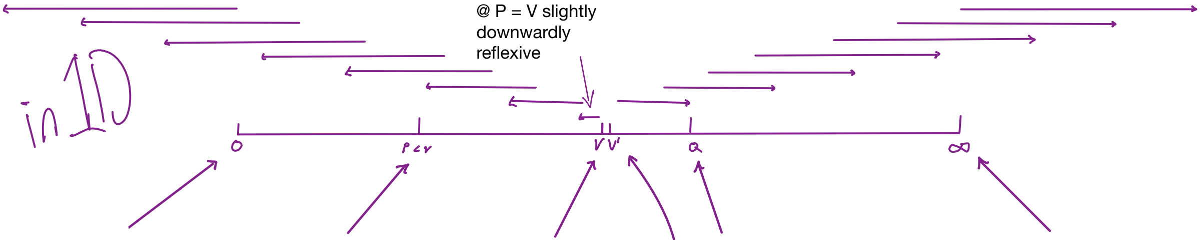
Shape and Scale of PRICE

(1.3) what are the features of each relevant axis (i.e., continuous or discrete, shape, scale)?

Price:

- shape: highly reflexive, max concavity slightly above V (V')
  - 1st derivative: positive slope until V', then negative slope
    - if P > V' there is upward price pressure
    - if P < V' there is downward price pressure
  - 2nd derivative: concavity throughout
    - as P increases further above V', P experiences more upward price pressure
    - as P decreases further below V', P experiences more downwards price pressure
  - 3rd derivative: more concave closer to V'
    - as P increases further above V', the rate of increase of upward price pressure decreases
    - as P decreases further below V', the rate of increase of downward price pressure decreases
- scale: relationship between price and volatility of price is highly influenced in practice by the pricing function of liquidity
  - 1st derivative: expansion if far from V', compression if close to V'
    - if P far from V' there is expansion
    - if P is close to V' there is compression
  - 2nd derivative : more compression closer to V', more expansion further from V'
    - as P moves further away from V' there is more expansion
    - as P moves closer to V' there is more compression





theoretically, P can approach the limit at 0 but will never hit 0.

however, anytime P decreases further below V', demand for Beans is still expected to decrease further. therefore, it is reasonable to show the curve as having a negative second derivative at all points. Although one could also argue to have a concave end close to 0, this would be primarily due to the pricing function.

it does not seem that there is a natural opposite to Q. the closest thing would be a value very close to but less than V at which point Beanstalk would expect a debt cycle to begin, if it has not already. however, a more sophisticated model could have a debt cycle indicator which is arbitrarily composed as opposed to only being a function of P.

when P = V there are no Beans being minted so there is still a slightly negative trend in P.

when P = V' there is neither a positive nor negative force acting on P as a result of P. However, V' is unknown in practice.

Q is a new parameter where above Q P is considered very high. very high P compromises utility of Bean because potential for price spikes makes Beans less attractive to price other assets against. However, a little upside volatility is helpful to system health.

Q is probably ideally between $1.02 and $1.05 until Beanstalk is larger. then it could decrease below $1.01.

theoretically P can approach infinity, in practice even at small market caps P topped out at $~4.00, $~1.67 and $~1.30.

however, as P increases further above V', demand for Beans is still expected to increase further. therefore, it is reasonable to show the curve as having a negative second derivative at all points. Although one could also argue to have a concave end close to infinity, this would be primarily due to the pricing function.

Shape and Scale of DEBT LEVEL

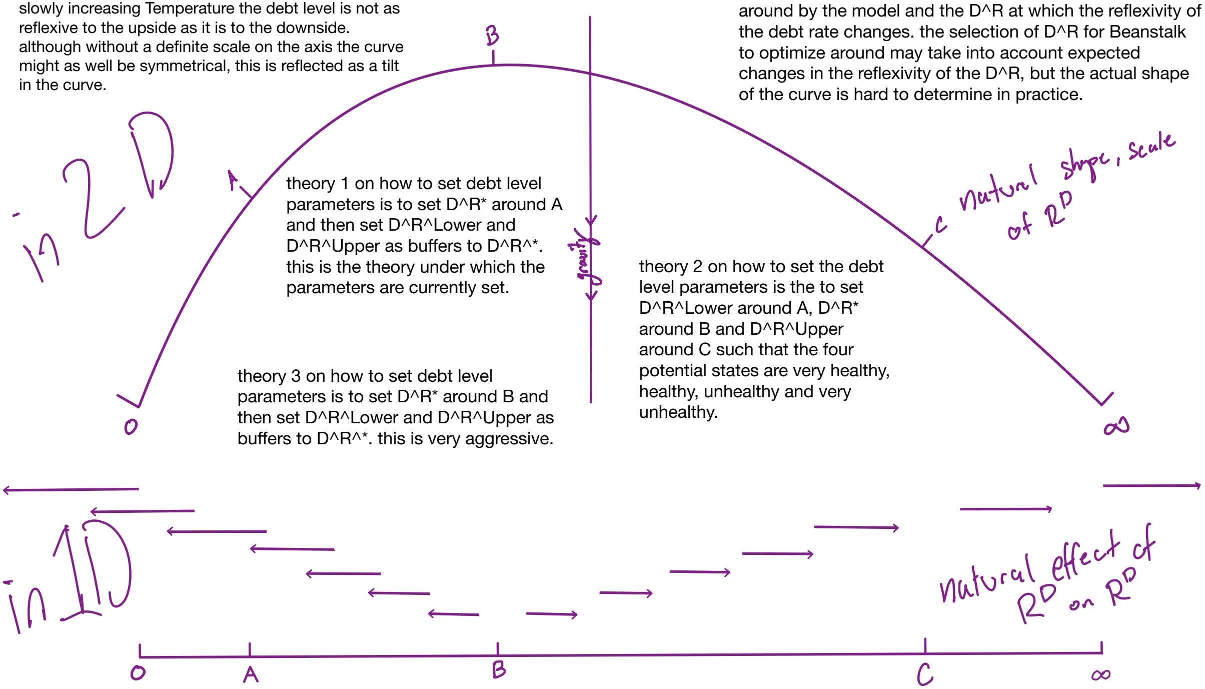
(1.3) what are the features of each relevant axis (i.e., continuous or discrete, shape, scale)?

Debt Level:

- shape: modestly reflexive, max concavity at unknown value (B)
    - 1st derivative: positive slope until B, then negative slope
        - if D^R > B there is upward pressure on D^R
        - if D^R < B there is downward pressure on D^R
    - 2nd derivative: concave throughout
        - higher D^R -> higher temps to attract sowers -> more debt issuance
        - lower D^R -> lower temps to attract sowers -> less debt issuance
    - 3rd derivative: concavity increasing if R^D < B and decreasing if B < D^R
        - at higher debt levels above B marginal increases in debt level will lead to less increases in debt issuance
- scale: negatively correlated with volatility until B, then positively correlated
    - 1st derivative: expansion if far from B, compression if close to B
        - if D^R far from B there is expansion
        - if D^R is close to B there is compression
    - 2nd derivative : more compression closer to B, more expansion further from B
        - as D^R moves further away from B there is more expansion
        - as D^R moves closer to B there is more compression

in practice D^R levels much higher than A, B or C have been observed with minimal negative consequences to Beanstalk. therefore, with proper Soil issuance and slowly increasing Temperature the debt level is not as reflexive to the upside as it is to the downside. although without a definite scale on the axis the curve might as well be symmetrical, this is reflected as a tilt in the curve.

unlike the case of price, where the change in the first derivative is directly related to V, in the case of D^R there is no direct relation between the D^R levels being optimized around by the model and the D^R at which the reflexivity of the debt rate changes. the selection of D^R for Beanstalk to optimize around may take into account expected changes in the reflexivity of the D^R, but the actual shape of the curve is hard to determine in practice.



theory 1 on how to set debt level parameters is to set D^R* around A and then set D^R^Lower and D^R^Upper as buffers to D^R^*. this is the theory under which the parameters are currently set.

theory 2 on how to set the debt level parameters is the to set D^R^Lower around A, D^R* around B and D^R^Upper around C such that the four potential states are very healthy, healthy, unhealthy and very unhealthy.

theory 3 on how to set debt level parameters is to set D^R* around B and then set D^R^Lower and D^R^Upper as buffers to D^R^*. this is very aggressive.

A - the debt level below which the system is so healthy that it is able to issue as much debt as desired at low interest rates such that the debt level does not increase quickly.

likely somewhere between 25% and 50%.

D^R^* should be somewhere near A, conservatively set slightly below where A is estimated to be or aggressively set slightly above.

B - the debt level below which the system is healthy such that it becomes easy to attract creditors at reasonable rates and above which the system is unhealthy such that it becomes difficult to attract creditors at reasonable rates.

likely somewhere between 80% and 150%.

there is an argument to be made D^R^Upper should be close to B, but doing so is somewhat aggressive. a more conservative approach is to keep R^D^Upper closer to A.

C - the debt level above which the system is so unhealthy that the marginal increase in debt level does not lead to an increase in temperature because the debt level is so high that demand for debt becomes less correlated with temperature.

likely somewhere between 500% and 1000%.

Beanstalk has experienced extended periods out beyond C. in practice in such instances the Silo ends up playing a larger role in peg maintenance.

Shape and Scale of L2SR

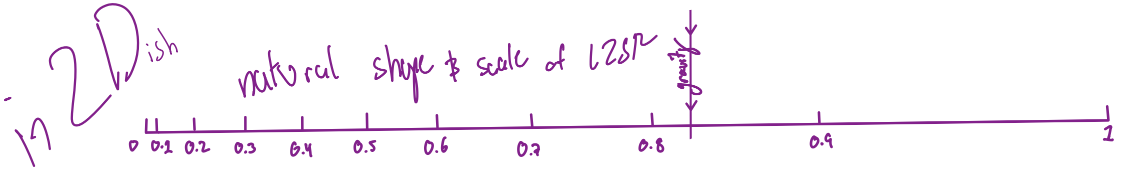
(1.3) what are the features of each relevant axis (i.e., continuous or discrete, shape, scale)?

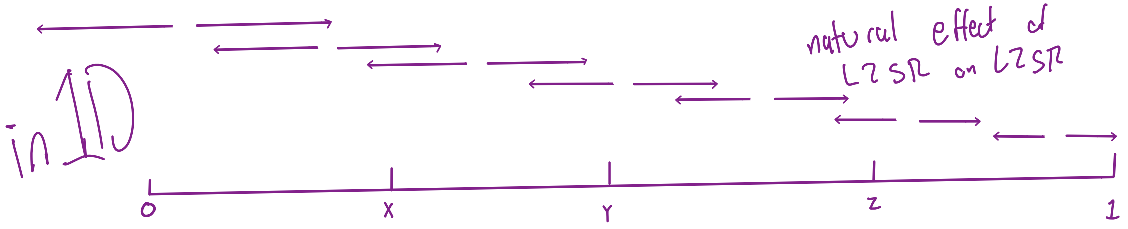
L2SR

- shape: consistently uncorrelated with reflexivity
- scale: increasingly negatively correlated with volatility
  - 1st derivative: as L2SR increases (decreases), volatility decreases (increases)
    - if L2SR increases -> expansion
    - if L2SR decreases -> compression
  - 2nd derivative: the more L2SR increases (decreases), the more expansion (compression)
    - same increase of L2SR @ higher L2SR -> less L2SR volatility -> more expansion
    - same decrease of L2SR @ lower L2SR -> more L2SR volatility -> more compression

similar to the case of debt level there is no direct relation between the L2SR levels being optimized around by the model and the L2SRS at which the volatility of the L2SR changes. the selection of L2SR levels for Beanstalk to optimize around may take into account expected changes in the volatility of the L2SR, but the actual scale of the axis is hard to determine in practice.

the L2SR is not reflexive, so the 2D version is still flat.





X - the L2SR level below which the L2SR is highly volatile.

likely somewhere between 40% and 60%.

L2SR* should be somewhere near X, conservatively set slightly above where X is estimated to be or aggressively set slightly below.

L2SR^Lower should be set at some amount less than L2SR* (and X) such that below L2SR^Lower there is very little liquidity remaining so Beanstalk no longer wants to encourage conversions from LP to Bean within the Silo in an attempt to save liquidity.

Y - the L2SR level below which the L2SR is increasingly more volatile and above which the L2SR is increasingly less volatile.

likely somewhere between 55% and 80%.

there is an argument to be made L2SR* should be close to Y, but doing so is quite conservative. a more aggressive approach is to place L2SR* closer to X.

Z - the L2SR level above which the L2SR is not volatile.

likely somewhere between 65% and 80%.

Beanstalk has experienced extended periods out beyond Z. in practice in such instances the Silo ends up playing a larger role in peg maintenance.

Direction, Acceleration

1 axis at a time

Direction and Acceleration of PRICE

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

Direction:
key points:

- V - optimal value for PRICE. in a perfect world the force acting on P would always move it towards V.
- V' - the value where the first derivative of the axis with respect to gravity is 0

key idea: the continuous slope away from V' makes it possible to see that P is always naturally moving away from V'.

- if P > V' the directional force of P on P is positive (*i.e.*, away from V)
- if P < V' the direction force of P on P is negative (*i.e.*, away from V)
- The directional force of P on P is away from ideal equilibrium for all P except V < P < V', which is negligible and impossible to measure in practice. .
- **Therefore, it can be understood that P is always naturally moving away from V.**

Acceleration:
key points:

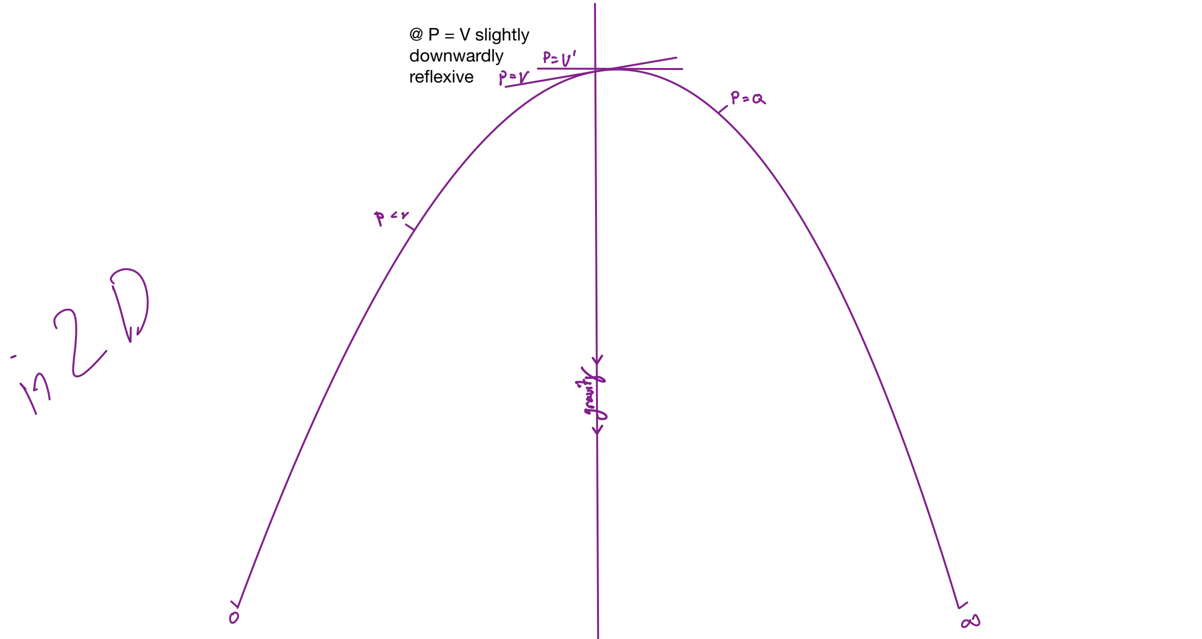
- V - optimal value for PRICE. in a perfect world the force acting on P would always move it towards V.
- V' - the value where the first derivative of the axis with respect to gravity is 0

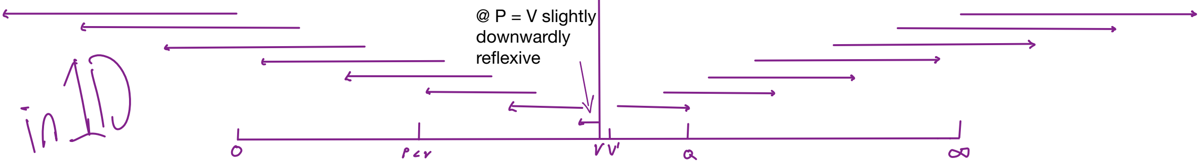
key idea: the continuous concavity with global maximum at V' makes it possible to see that P is always naturally accelerating away from V'

- if P > V' the directional force of P on P is increasing as P increases. therefore, P is naturally accelerating away from V if P > V'.
- if P < V' the directional force of P on P is increasing as P decreases. therefore, P is naturally accelerating away from V if P < V.
- P is accelerating away from V at all P except V < P < V', which is negligible and impossible to measure in practice.
- **Therefore, it can be understood that P is always naturally accelerating away from V.**

future work:

- some time based measurement of acceleration (*e.g.*, change in deltaB over the previous X Seasons compared to over the previous Season)


@ P = V slightly downwardly reflexive


@ P = V slightly downwardly reflexive

Direction and Acceleration of DEBT LEVEL

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

Direction:

key points:

- B - the value where the first derivative of the axis with respect to gravity is 0 **is unknown**

key idea: the lack of understanding of where A, B, or C are in practice makes understanding the direction of DEBT LEVEL with respect to itself very difficult.

theory:

- if D^R > B the directional force of D^R on D^R is positive (*i.e.*, away from B)
- if D^R < B the direction force of D^R on D^R is negative (*i.e.*, away from B)
- The directional force of D^R on D^R is away from B for all D^R.
- Therefore, it can be understood that D^R is always naturally moving away from B.

Acceleration:

key points:

- B - the value where the first derivative of the axis with respect to gravity is 0 **is unknown**

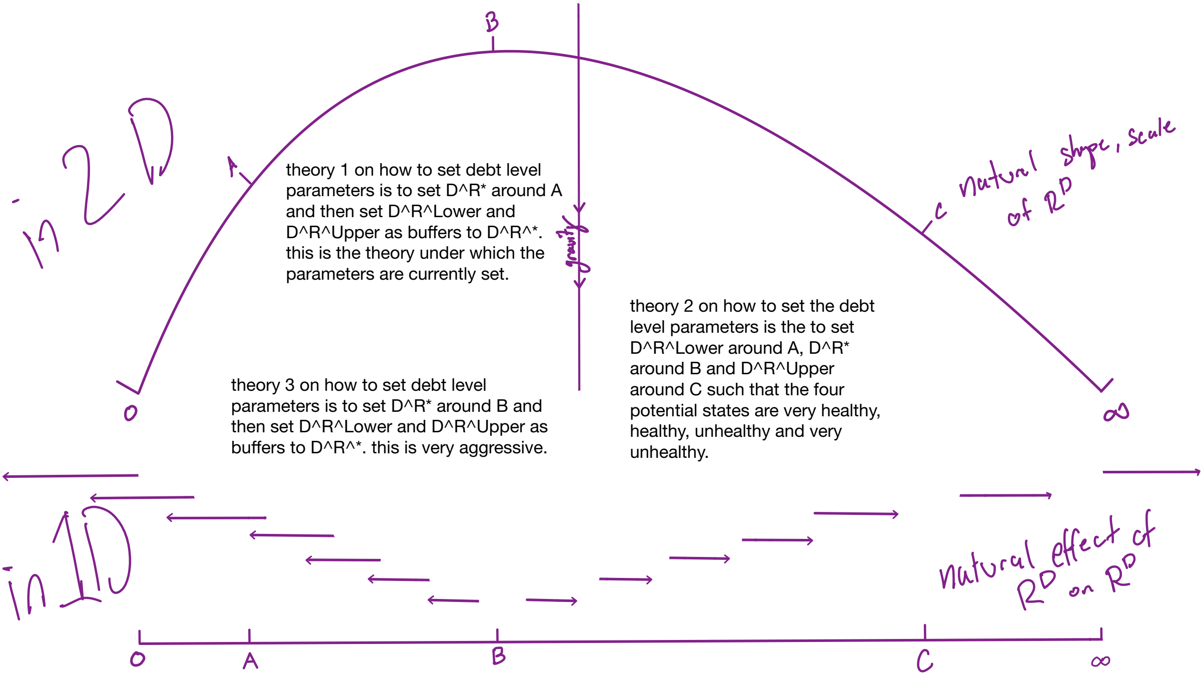
key idea: the lack of understanding of where A, B, or C are in practice makes understanding the acceleration of DEBT LEVEL with respect to itself very difficult.

theory: the continuous concavity with global maximum at B makes it possible to see that D^R is always naturally accelerating away from B

- if D^R > B the directional force of D^R on D^R is increasing as D^R increases. therefore, D^R is naturally accelerating away from B if D^R > **V'**.
- if D^R < B the directional force of D^R on D^R is increasing as D^R decreases. therefore, D^R is naturally accelerating away from B if D^R < B.
- D^R is accelerating away from B at all D^R.
- Therefore, it can be understood that D^R is always naturally accelerating away from B.

future work:

- study of where are A, B and C in practice. how can they be estimated in real time?



theory 1 on how to set debt level parameters is to set D^R* around A and then set D^R^Lower and D^R^Upper as buffers to D^R^*. this is the theory under which the parameters are currently set.

theory 2 on how to set the debt level parameters is the to set D^R^Lower around A, D^R* around B and D^R^Upper around C such that the four potential states are very healthy, healthy, unhealthy and very unhealthy.

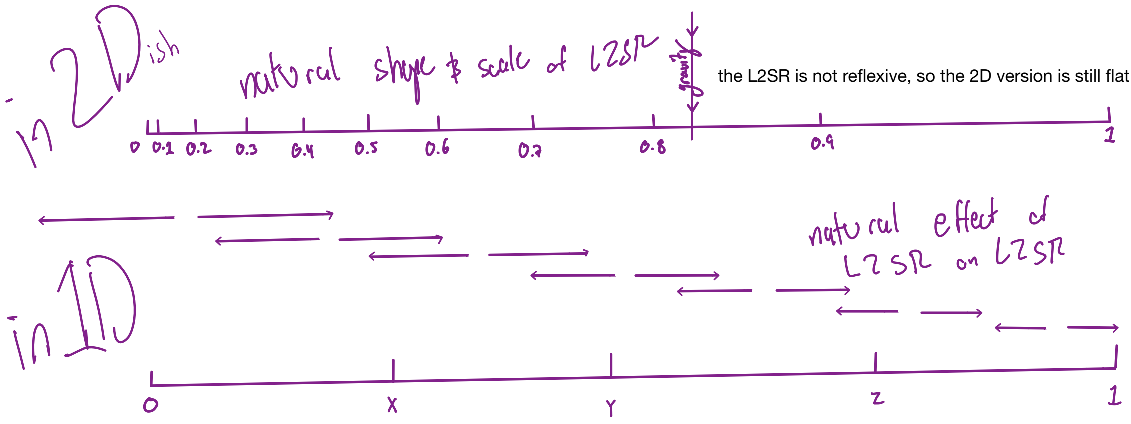
theory 3 on how to set debt level parameters is to set D^R* around B and then set D^R^Lower and D^R^Upper as buffers to D^R^*. this is very aggressive.

Direction and Acceleration of L2SR

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

Direction:
key idea: there is no directional force in terms of the L2SR acting on itself

Acceleration:
key idea: the lack of directional force implies a lack of natural acceleration as well.



the L2SR is not reflexive, so the 2D version is still flat.

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

question 1.4 has been answered for each axis individually.

so what?

understanding the shape, scale, direction and acceleration in one dimension is a helpful foundation to understand the complex interplay between each of the dimensions with one another, both 2 and a time and eventually all 3 at once.

Shape, Scale,

Direction, Acceleration

2 axes at a time

Price α Debt level

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

The following questions can be used to better understand the interplay between PRICE and DEBT LEVEL:

(1.4.a) how does PRICE act differently upon itself at various DEBT LEVELS?

(1.4.b) how does DEBT LEVEL act differently upon itself at various PRICES?

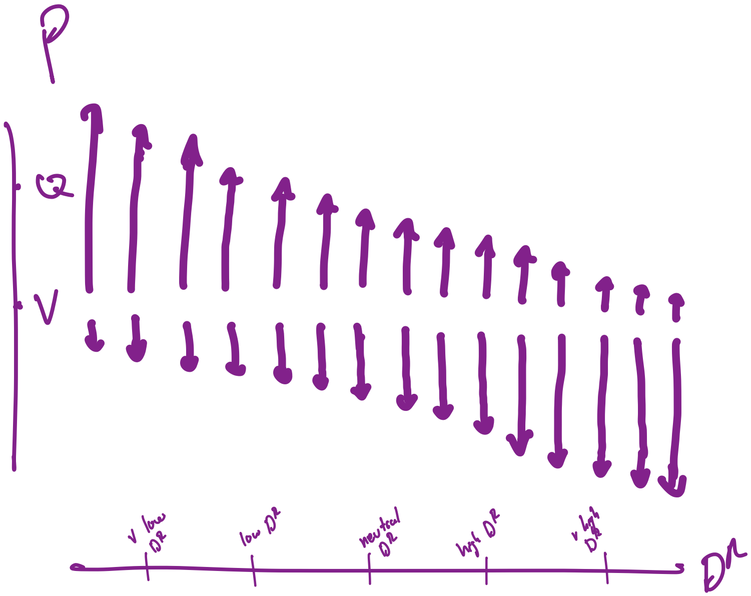
(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL?

(1.4.d) given the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL, what can Beanstalk infer about its state in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

(1.4.a) how does PRICE act differently upon itself at various DEBT LEVELS?

<u>Effect of D^R on P</u>

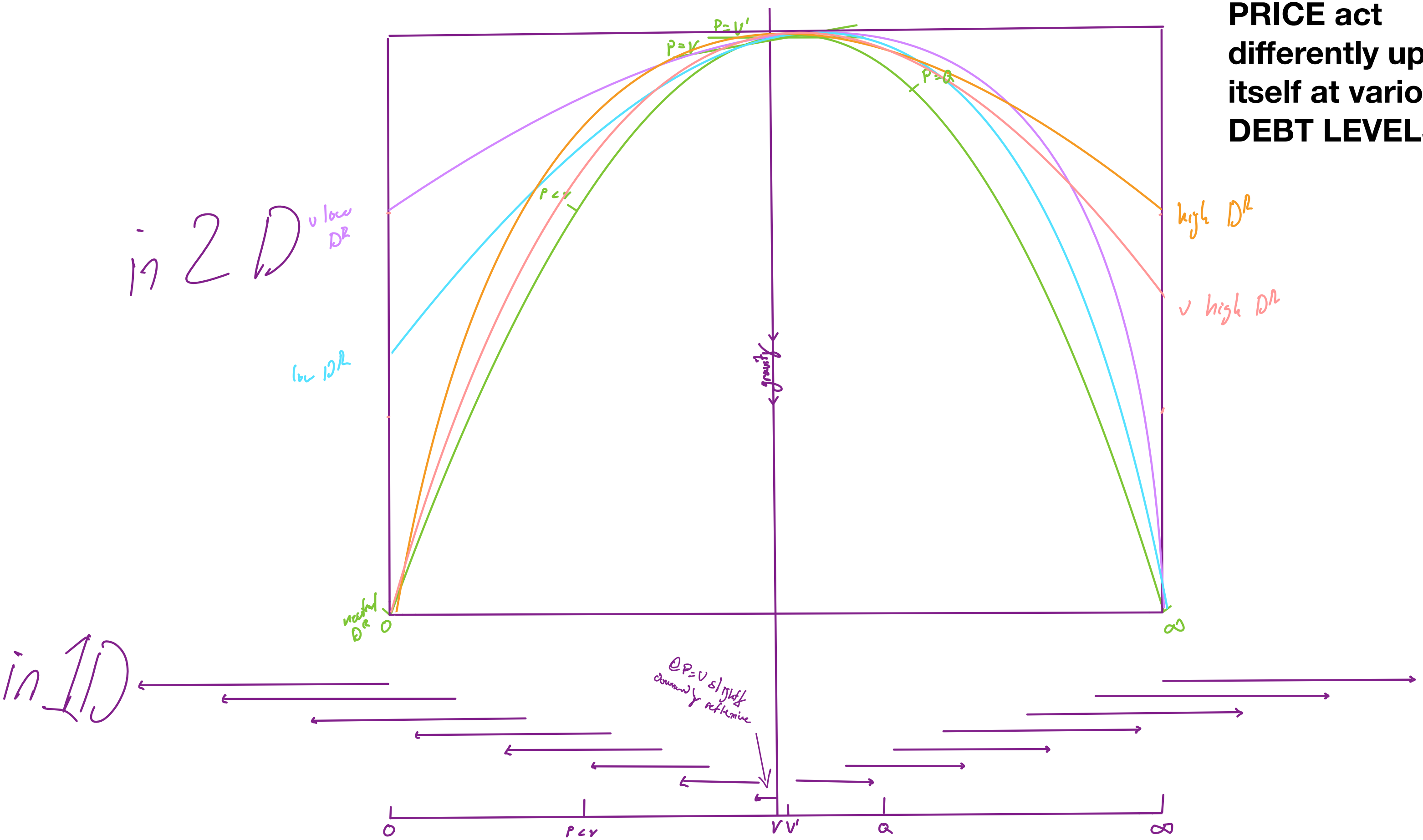
- as D^R increases P becomes less reflexive when P > V’ and more reflexive when P < V’ because at higher Debt Levels it is easier for the Bean price to decrease than increase.
- as D^R decreases P becomes more reflexive when P > V’ and less reflexive when P < V’ because at higher Debt Levels it is easier for the Bean price to i crease than decrease.



natural effect of D^R on P

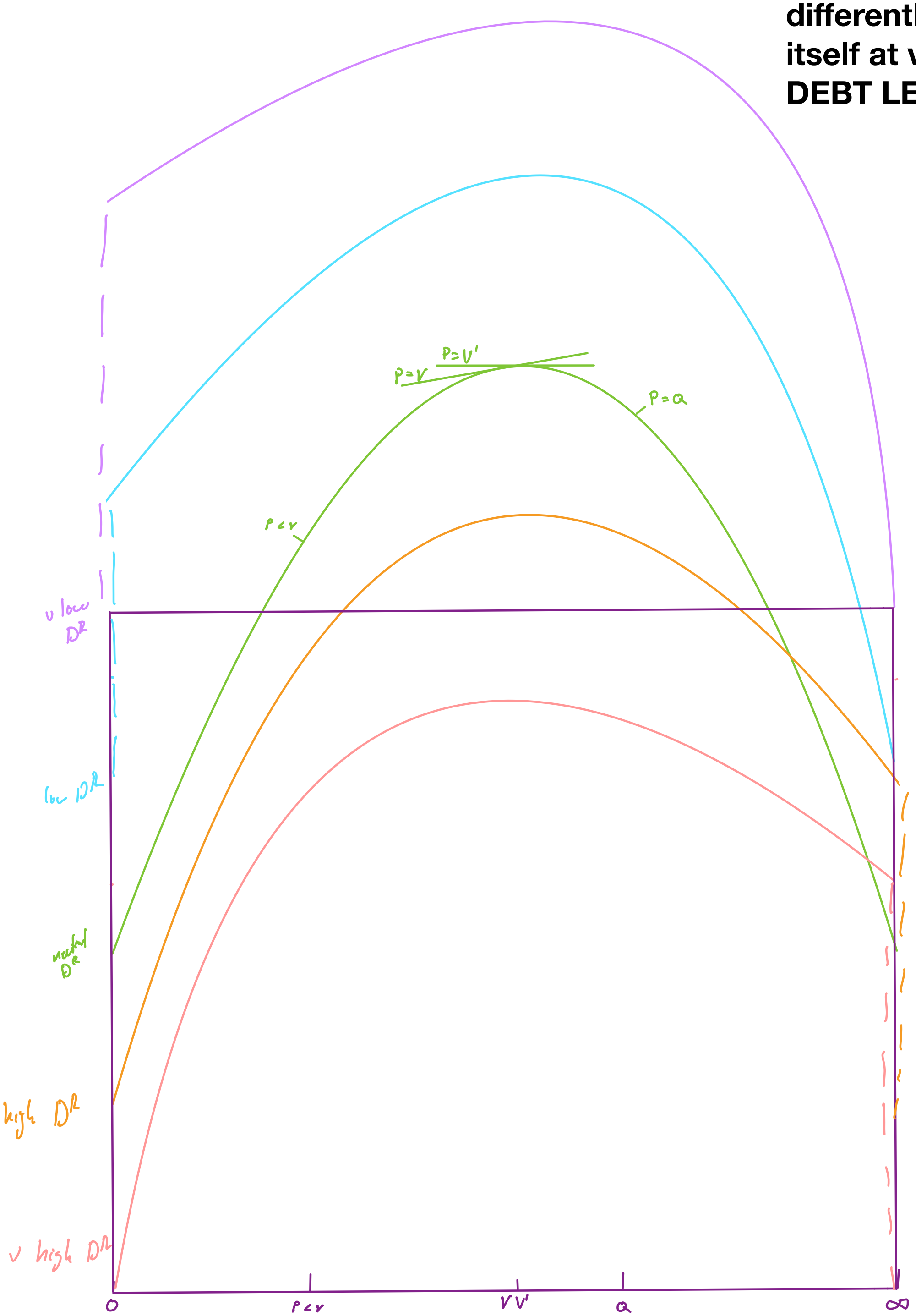
Natural Shape, Scale of P at various D^R

(1.4.a) how does PRICE act differently upon itself at various DEBT LEVELS?



(1.4.a) how does PRICE act differently upon itself at various DEBT LEVELS?

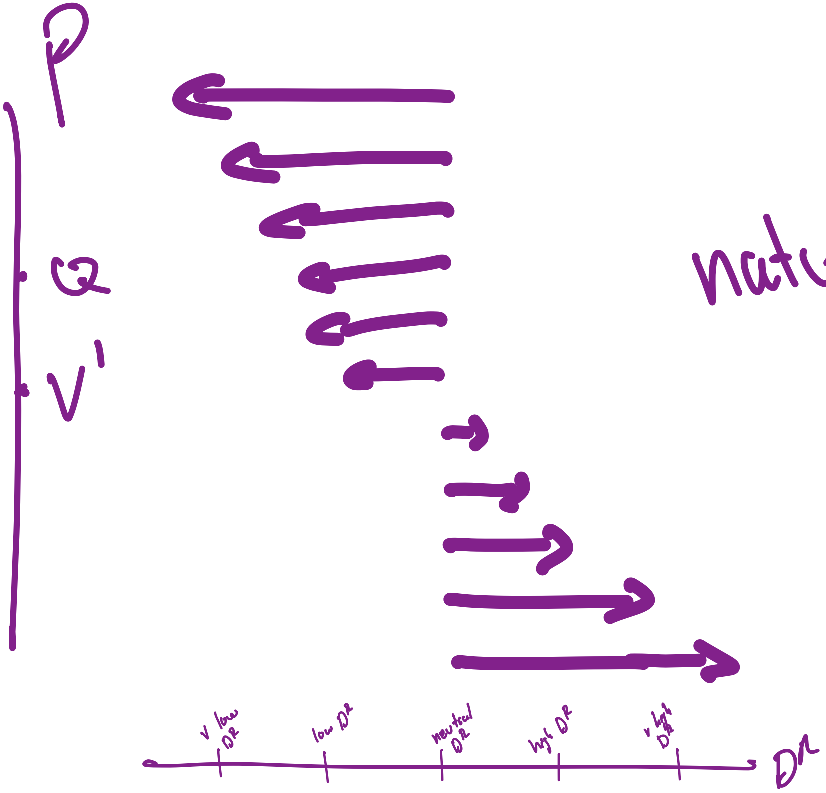
Natural Shape, Scale of P at various D^R



(1.4.b) how does DEBT LEVEL act differently upon itself at various PRICES?

Effect of P on D^R

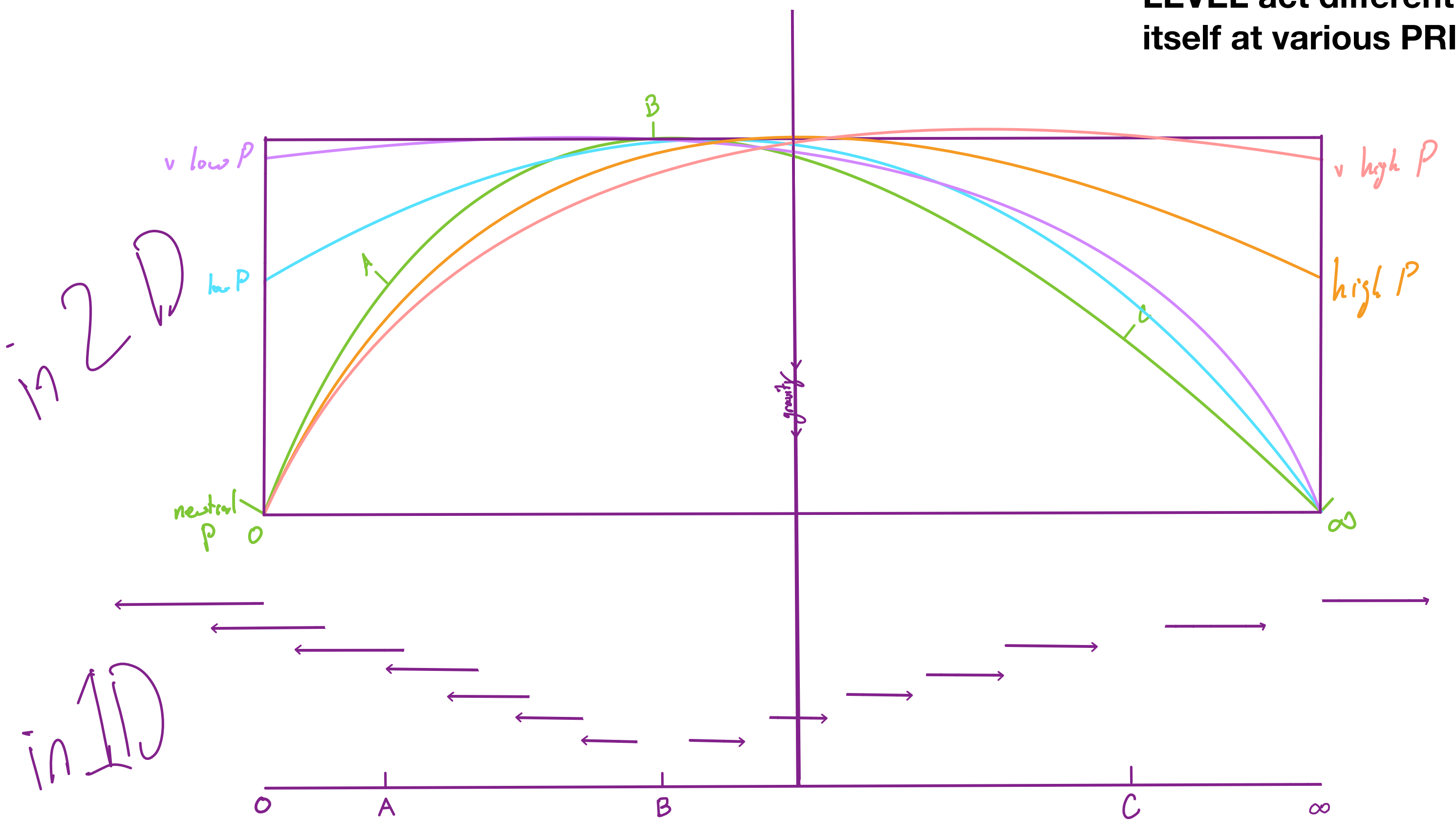
- the higher P is above V’, the faster D^R decreases because more Beans are being minted
- the lower P is below V’, the faster D^R can increase because more Soil is being minted



natural effect of P on D^R

Natural Shape, Scale of D^R at various P

(1.4.b) how does DEBT LEVEL act differently upon itself at various PRICES?

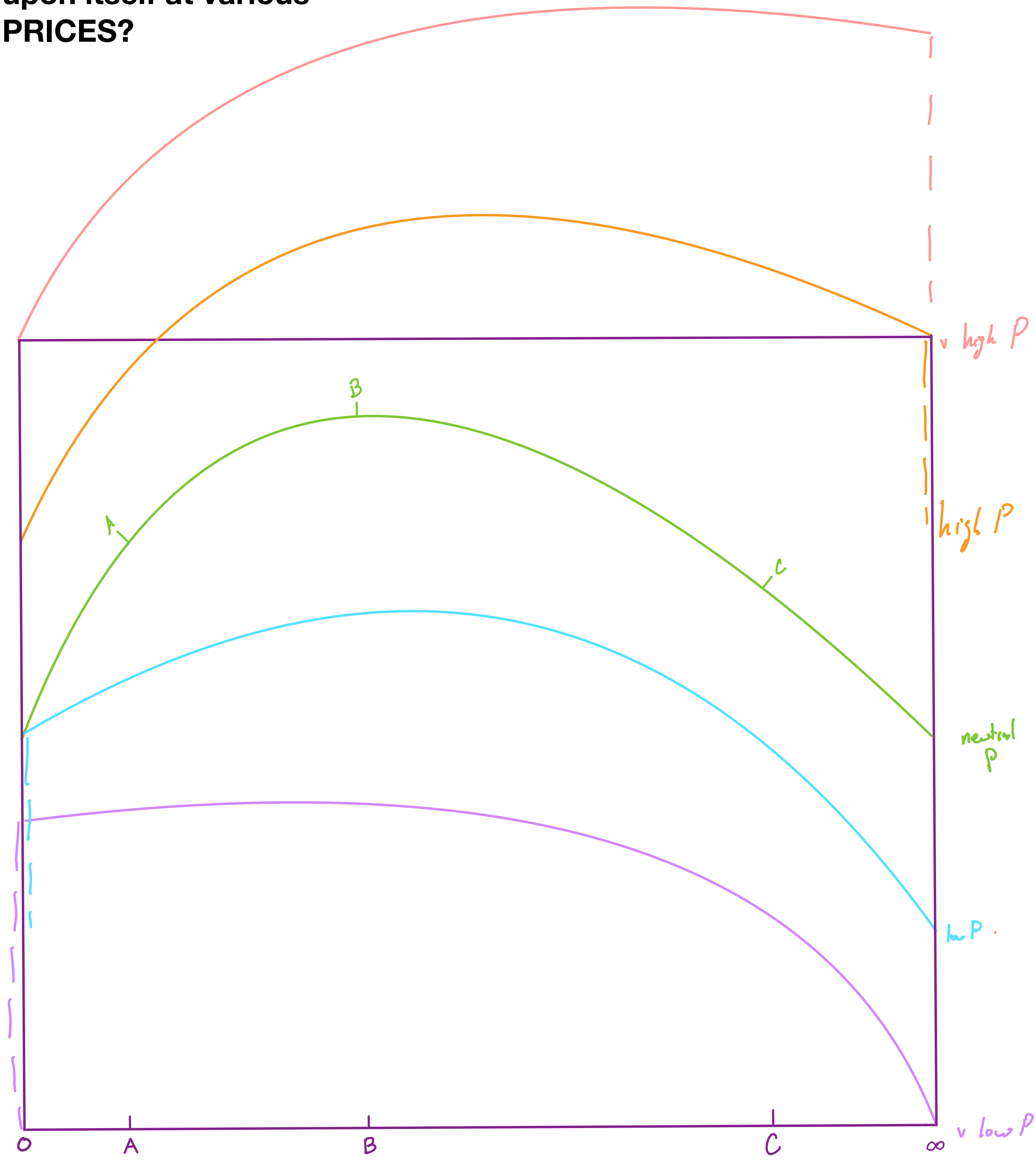


Open Question:

To what extend to A, B and C change at various P?

(1.4.b) how does DEBT LEVEL act differently upon itself at various PRICES?

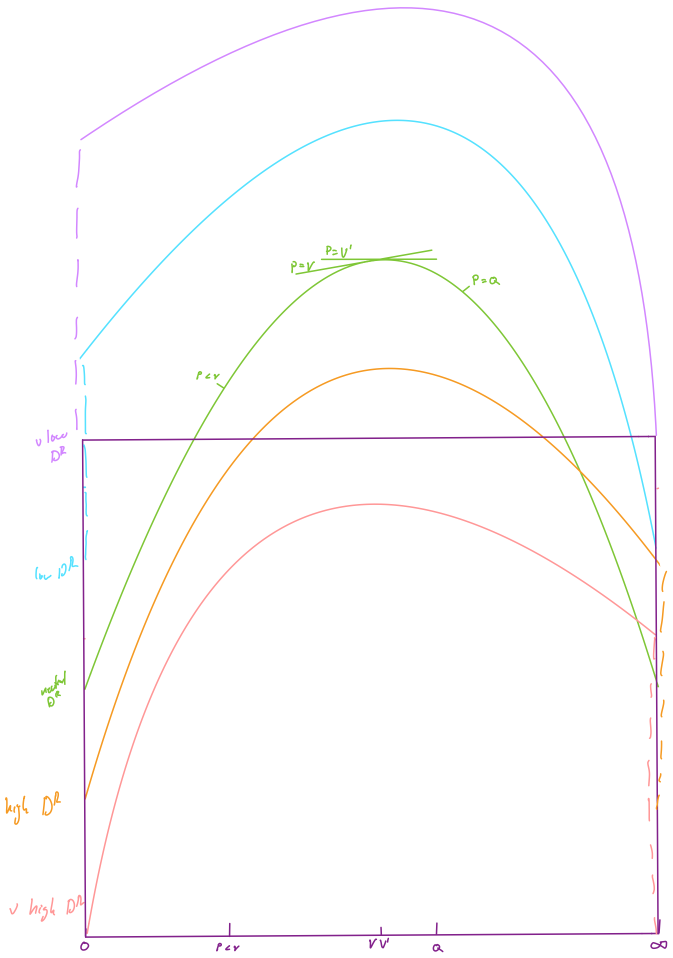
Natural Shape, Scale of D^R at various P



(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL?

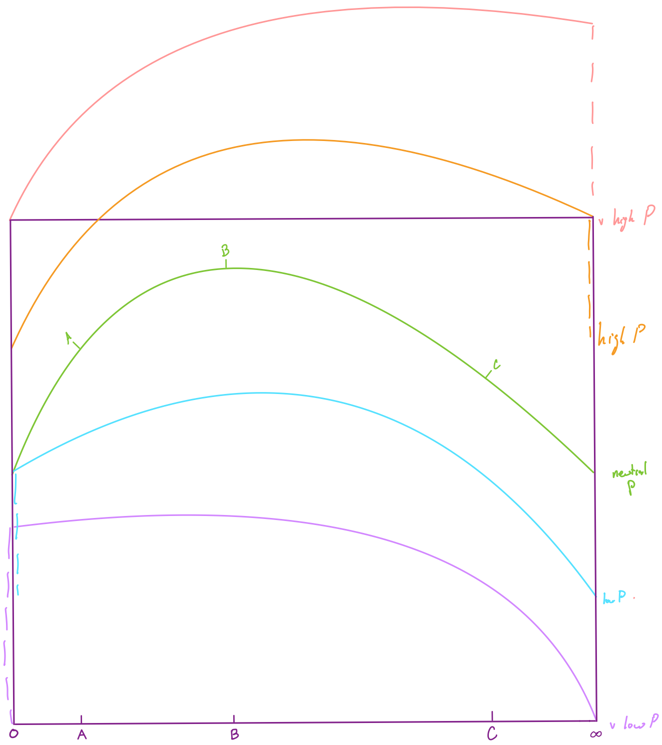
key idea: stitch the shapes and scales of the 2 axes together into a single space

Natural Shape, Scale of P at various D^R



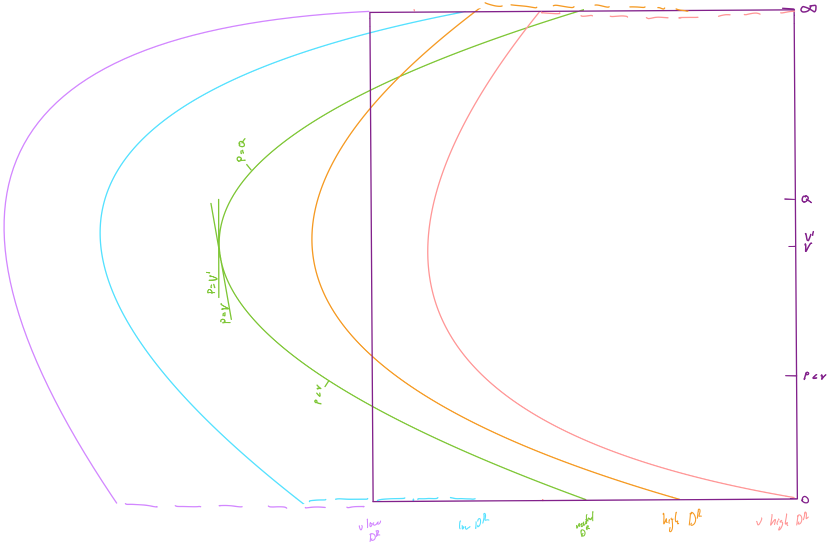
×

Natural Shape, Scale of D^R at various P



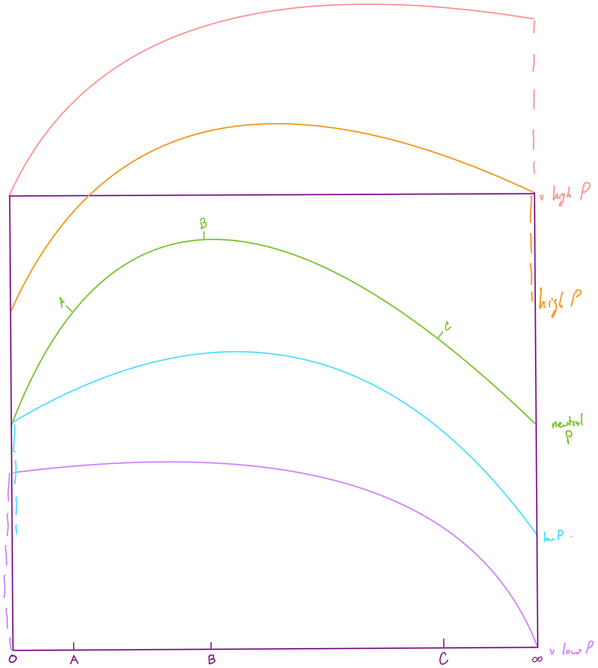
(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL?

Natural Shape, Scale of P at various D^R



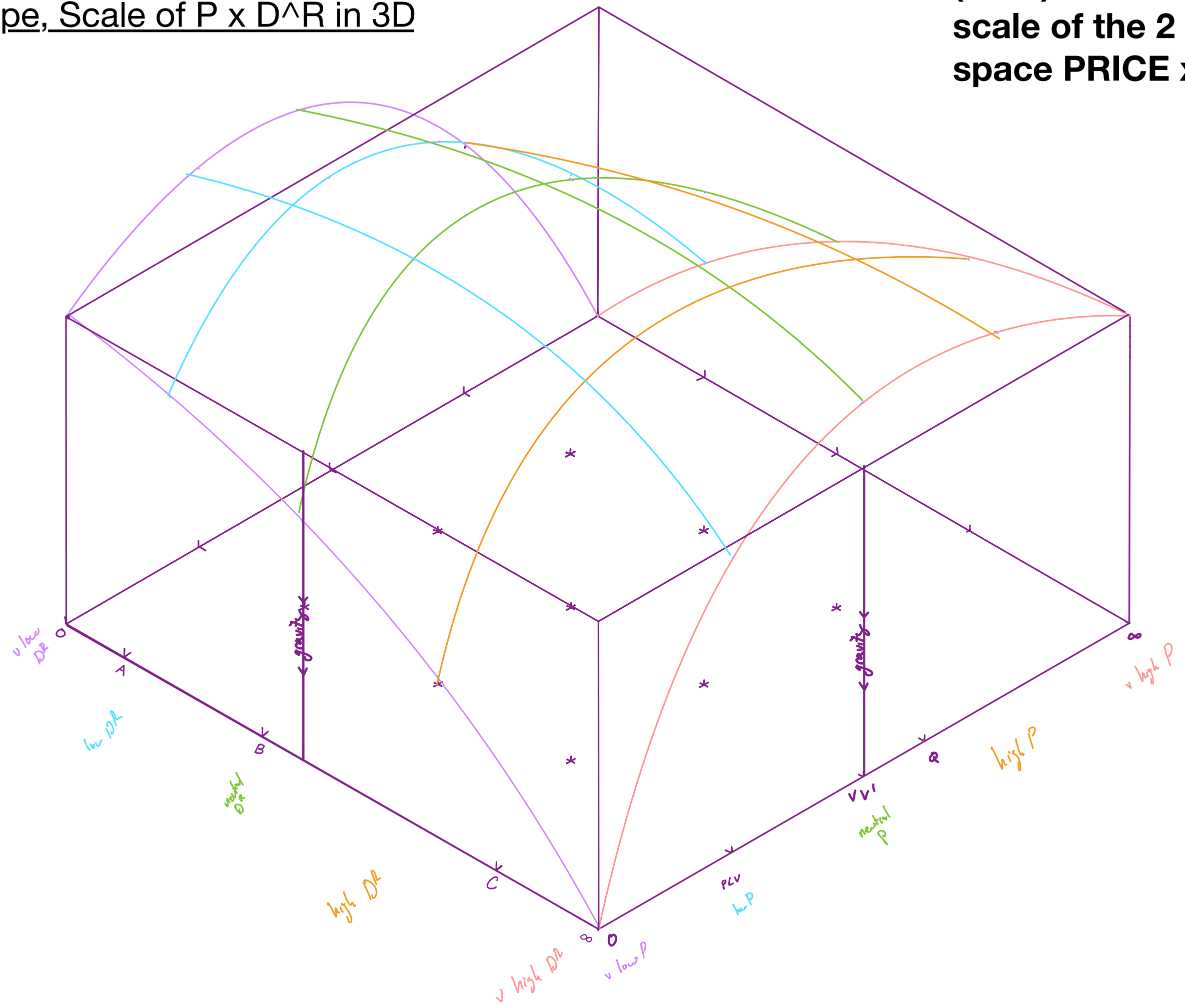
X

Natural Shape, Scale of D^R at various P



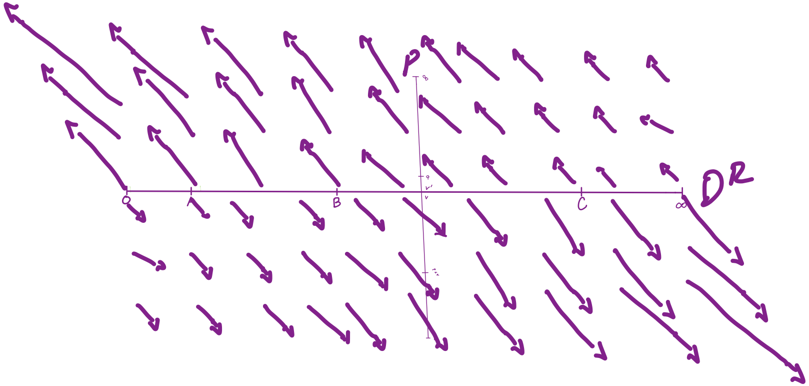
Natural Shape, Scale of P x D^R in 3D

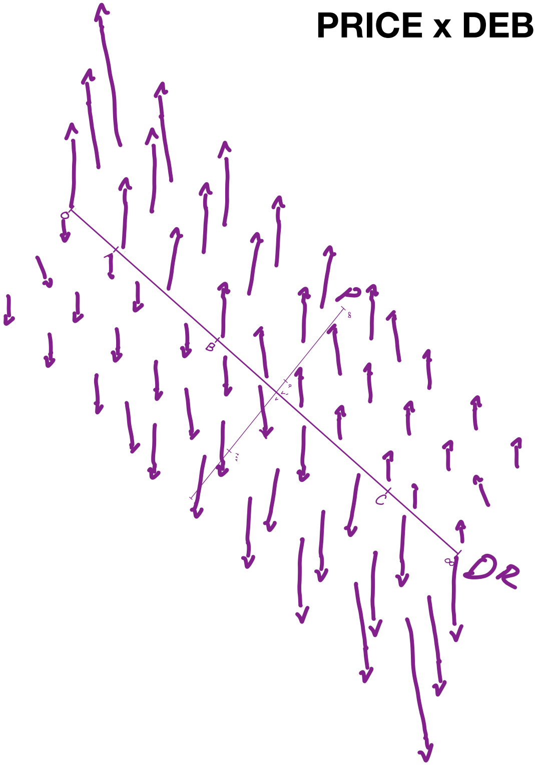
(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL?



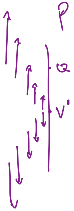
Natural Shape, Scale of P x D^R in 2D

(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL?

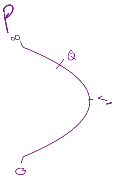




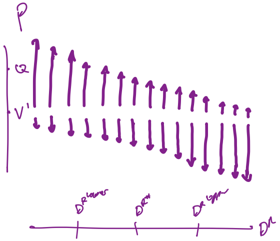
Direction in 2 Dimensions Cheat Sheet: Price x Debt Level



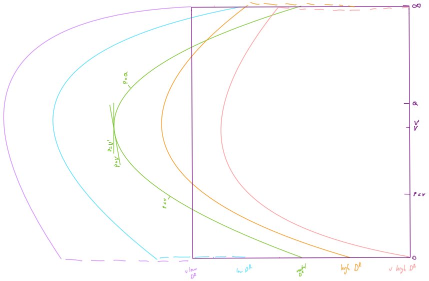
natural effect of P on P @ various P



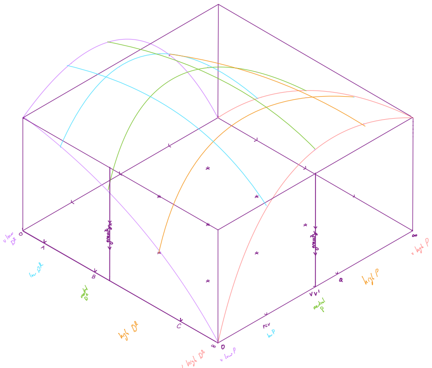
natural shape of P,



natural effect of D^R on P



natural shape of P @ various D^R



natural shape, scale of P x D^R in 3D

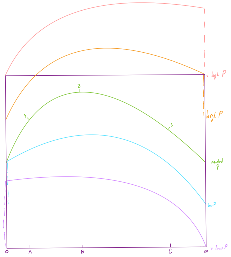
price

- highly reflexive
  - as P increases further past V', P trends upwards faster
  - as P decreases further past V', P trends downwards faster
- change in directional effect @ V'

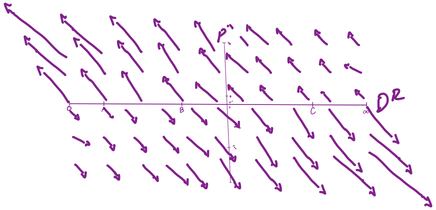
D^R on P

- as D^R increases P becomes less reflexive when P > V' and more reflexive when P < V'
- as D^R decreases P becomes more reflexive when P > V' and less reflexive when P < V'

natural shape, scale of D^R @ various P



Natural Shape, Scale of P x D^R in 2D



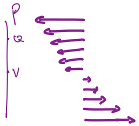
P on D^R

- the higher P is above V', the faster D^R
- the lower P is below V', the faster D^R can increase

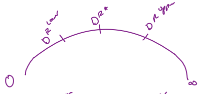
debt level

- moderately reflexive
  - higher debt level requires higher Temperature which increases debt level faster
  - lower debt level requires lower Temperature which increases debt level slower
- turning point is not necessarily around D^R^*

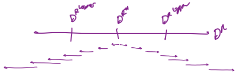
natural effect of P on D^R



natural shape of D^R



natural effect of D^R on D^R @ various D^R



Shape and Scale of PRICE x DEBT LEVEL

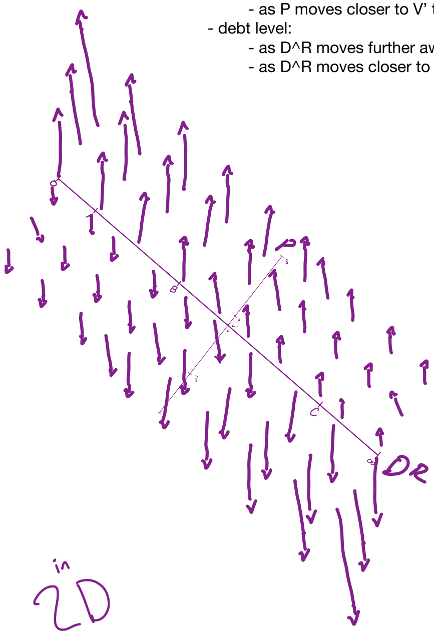
(1.4.d) given the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL, what can Beanstalk infer about its state in terms of derivatives of position (i.e., direction, acceleration, etc.)?

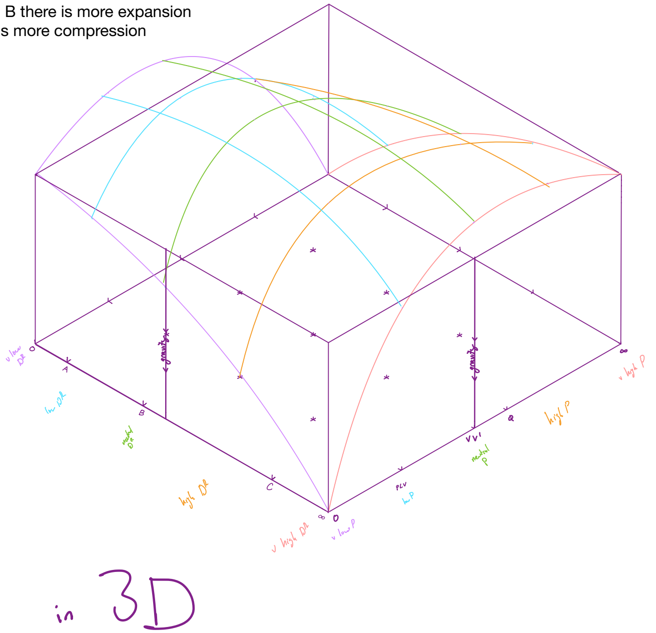
Shape: highly reflexive, global maximum at (V', B), max concavity at (v low P, v high D^R) and (v high P, v low D^R)

- 1st derivative: negative slope wrt P if P > V', positive slope if P < V'. positive slope wrt D^R if P > V, negative slope if P < V.
  - price:
    - if P > V' there is upward price pressure. D^R is inversely correlated with upward price pressure.
    - if P < V' there is downward price pressure. D^R is inversely correlated with downward price pressure.
  - debt level:
    - if P > V there is downward pressure on D^R. P is correlated with downward D^R pressure.
    - if P < V there is upward pressure on D^R. P is inversely correlated with upward D^R pressure.
- 2nd derivative: concave throughout. global maximum at (neutral P, neutral D^R) OR (V, B).
  - price:
    - if P > V', D^R is inversely correlated with the rate of increase in upwards price pressure.
    - if P < V', D^R is correlated with the rate of increase in downwards price pressure.
  - debt level:
    - if P > V, P is correlated with the rate of increase in downward D^R pressure.
    - if P < V, P is inversely correlated with the rate of increase in upward D^R pressure.
- 3rd derivative: gravitational "holes" at (v low P, v high D^R) and (v high P, v low D^R)

Scale: relationship between volatility of price and debt level is highly influenced in practice by the demand for Soil. When there is no demand for Soil they are unrelated. When there is excess demand for Soil they are highly related because Beanstalk can limit downside volatility in Price at the cost of volatility in debt level via Soil issuance.

- 1st derivative: expansion if far from (V', B), compression if close to (V', B).
  - price:
    - if P far from V' there is expansion
    - if P is close to V' there is compression
  - debt level:
    - if D^R far from B there is expansion
    - if D^R is close to B there is compression
- 2nd derivative : more compression closer to (V', B), more expansion further from (V', B).
  - price:
    - as P moves further away from V' there is more expansion
    - as P moves closer to V' there is more compression
  - debt level:
    - as D^R moves further away from B there is more expansion
    - as D^R moves closer to B there is more compression





intuitively, the difficulty of balancing a ball on this surface is far from a simple task. nonetheless, this is Beanstalk's current peg maintenance task (*i.e.*, before Seed Gauge).

while it may seem like a good thing to be in the corner with high price and low debt level, it requires a lot of momentum to get out of the hole. therefore, when Beanstalk's price finally comes down in such a scenario its momentum is likely to carry it into a major debt cycle with low prices, getting stuck in the other corner. therefore, it is best to avoid either hole.

better understanding the shape of this space can help answer questions like 'why doesn't Beanstalk prioritize peg maintenance over paying premiums for debt issuance? Clearly avoiding extremely high levels of debt, beyond which there is no return, is a priority over short term peg maintenance.

the relationship between price and debt level becomes clear by looking at this space: the debt level can only decrease when P > V and only increase when P < V.

note, the actual locations of A, B, C, D^R^Lower, D^R^*, D^R^Upper and Q along these curves are unknown in practice. this makes Beanstalk's peg maintenance task more complex.

Direction and Acceleration of PRICE x DEBT LEVEL

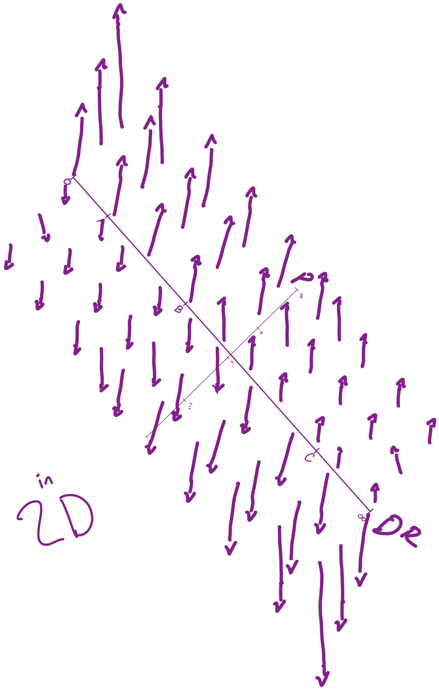
Direction: **D^R is increasing when P < V and decreasing when P > V. Price is overwhelmingly important compared to debt level in its effect on the shape of space wrt both price and debt level.**

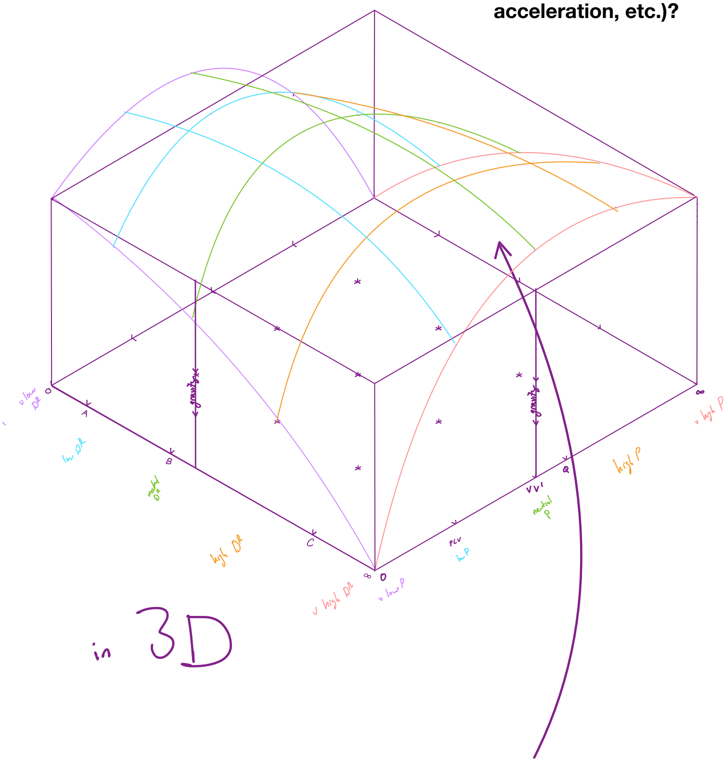
- if P > V’ the directional force of P x D^R is positive wrt P
- if P > V the directional force of P x D^R is negative wrt D^R
- if P < V’ the direction force of P x D^R is negative wrt P
- if P < V the directional force of P x D^R is positive wrt D^R

Acceleration: **Beanstalk naturally accelerates towards (v low P, v high D^R) if P < V’ and towards (v high P, v low D^R) if P > V’**

- if P > V’ the directional force of P x D^R on P is increasing in magnitude as P increases.
- if P < V’ the directional force of P x D^R on P is increasing in magnitude as P decreases.
- if P > V’ the directional force of P x D^R on D^R is increasing in magnitude as P increases.
- if P < V’ the directional force of P x D^R on D^R is increasing in magnitude as P decreases

(1.4.d) given the shape and scale of the 2 dimensional space PRICE x DEBT LEVEL, what can Beanstalk infer about its state in terms of derivatives of position (i.e., direction, acceleration, etc.)?





Ideal equilibrium lies somewhere near this line. In practice, properly setting D^R^Lower, D^R and D^R^Upper is very difficult.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

using the most optimal manipulation resistant source for position allows Beanstalk to respond to its state in the most efficient manner possible.

<u>time</u>
the current state of Beanstalk is evaluated relative to its ideal state exclusively from data over the previous Season.

while a more sophisticated peg maintenance model could account for historical state (i.e., state over more time than just the past Season), as will continue to become clear in this document, the current peg maintenance mechanism is already quite complex. given its current complexity and the questionable benefits of accounting for more than one Season of data, at this time it does not seem prudent to expand the inputs to the peg maintenance mechanism beyond data from the previous Season.

however, given the modularity of the inputs to measuring state it is possible to easily update over what time period or how one more more dimensions of position are measured.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

PRICE

Beanstalk uses deltaB as a proxy for PRICE, as they are always correlated. deltaB is calculated using a time-weighted average comparison between the ratio of the bean/eth basin well and the usd/eth chainlink price over the course of the previous Season.

With the additional classification of PRICE as > Q, there is the need to calculate the actual average price relative to Q. When deltaB is positive, P is calculated using the same time-weighted values (*i.e.*, over the course of the previous Season) used to calculate deltaB.

Therefore, Beanstalk evaluates price discretely as greater than V, less than V, or greater than Q.

The greater than Q case will go into effect upon implementation of the Seed Gauge.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

<u>DEBT LEVEL</u>

Beanstalk uses the Pod Rate as a proxy for DEBT LEVEL. Both the supply of Beans and outstanding Pods is calculated at the time of the Sunrise. Neither value requires explicit manipulation resistance because both values change exclusively downstream of manipulation resistant values.

Beanstalk evaluates D^R discretely as very low, low, high and very high depending on its position with respect to D^R^Lower, D^R* and D^R^Upper.

It is very difficult to properly set D^R^Lower, D^R* and D^R^Upper given the lack of understanding of the locations of A, B and C in practice.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

PRICE x DEBT LEVEL

The position of Beanstalk in the 2 dimensional space of PRICE x DEBT LEVEL is the combination of the position in PRICE and position in DEBT LEVEL

This leaves 2 cases with respect to price and 4 cases with respect to debt level for a total of 8 potential positions with respect to price and debt level in the current peg maintenance model.

Note in the Seed Gauge implementation a 3rd case is added to PRICE of $P > Q$.

(1.5) how should Beanstalk evaluate its position, *direction* and acceleration at the beginning of each Season in practice?

Ideal Equilibrium

Direction is much more useful with respect some sort of optimal point within the PRICE x DEBT LEVEL space.

Optimal point in space is (V, D^R*). V because it is the primary point of Beanstalk to oscillate P across V along the PRICE axis. D^R* because that is the arbitrarily defined optimal D^R along the DEBT LEVEL axis.

Beanstalk is defined to be in ideal equilibrium in space when (1) P is regularly oscillating across V, (2) the debt level is at D^R* and (3) demand for Soil is steady.

In practice maintaining ideal equilibrium is impossible and Beanstalk must always respond to its position direction and acceleration with respect to ideal equilibrium.

(1.5) how should Beanstalk evaluate its position, *direction* and acceleration at the beginning of each Season in practice?

Beanstalk is either moving away from ideal equilibrium (and into one of the two holes in the space) or towards ideal equilibrium (and out of one of the two holes in the space).

This is due to the strong relationship between P and D^R. One could argue Beanstalk is always moving towards one of the two holes and they would be correct.

A future peg maintenance model may use relative position, direction and acceleration relative to the two holes instead of relative to ideal equilibrium.

This leaves 2 cases with respect to direction, 2 cases with respect to price and 4 cases with respect to debt level for a total of 16 potential positions and directions with respect to price and debt level in the current peg maintenance model.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

The natural state of Beanstalk is always accelerating towards one of the two holes. however, it is possible for Beanstalk to measure its health in real time in the market via demand for Soil.

When demand for Soil is increasing Beanstalk is experiencing force towards the (v high P, v low D^R) hole. When demand for Soil is decreasing Beanstalk is experiencing force towards the (v low P, v high D^R) hole. When demand for Soil is steady Beanstalk is experiencing only the natural force and its momentum.

Demand for Soil is considered increasing, steady or decreasing.

By combining direction with the force derived from the measurement of demand for soil, Beanstalk can get a more accurate read on its second derivative with respect to ideal equilibrium than just always accelerating.

This leaves 3 cases with respect to acceleration, 2 cases with respect to price and 4 cases with respect to debt level for a total of 24 potential positions, directions and accelerations with respect to price and debt level in the current peg maintenance model.

While PRICE is being implemented with 3 cases as soon as the Seed Gauge goes live, it currently has only 2 cases. for the purposes of treating the current peg maintenance system, the 2 case version is considered here

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

When the maximum amount of Beans are sown in multiple consecutive Seasons, Beanstalk uses the time it took for all the Beans to be sown (*i.e.*, for demand to reach 0) to determine the change in demand for Soil.

Measuring changing demand for Soil is one of the most difficult tasks to do in a manipulation resistant fashion because if there is a known threshold at which point all the Soil is considered to be sown anyone can sow just slightly less than the threshold making it unlikely the threshold is reached because there is not enough Soil remaining to justify the gas cost.

One potential future solution for this is to have Beanstalk HIDE the exact threshold at which point all the Soil would be considered sown until the end of each Season, and to generate a slightly different one each time. Unclear how much that would help or the range of randomness necessary to meaningfully help.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

Basic intuition for determining state with respect to ideal equilibrium:

- When P > V, D^R is decreasing. When P < V, D^R can only increase
- Direction is towards ideal equilibrium if D^R is moving towards D^R*
- Demand for Soil is used for a proxy for health of the Soil market. This represents Beanstalk's ability to slow down or speed up the natural acceleration to (v low P, v high D^R) or (v high P, v high D^R) via issuance of debt.
- Acceleration is determined by whether the change in demand for Soil from Season to Season is:
  - directionally aligned with the natural acceleration wrt D^R —> accelerating
  - constant —> steady
  - directionally aligned against the natural acceleration wrt D^R —> decelerating

we have answered all the questions necessary to properly be able to classify state for the current implementation of Beanstalk’s peg maintenance model, which only operates in these 2 dimensions.

before we move on to other pairs of dimensions, lets pause to understand how Beanstalk currently classifies and responds to its state in PRICE x DEBT LEVEL.

Current Beanstalk

Price x Debt Level

Peg Maintenance

Response to State

(2) how should Beanstalk respond to its state at the beginning of each Season in an attempt to return to its ideal state?

Responding to State

Some key questions to answer in determining what the proper response to the Beanstalk state is are:

(2.1) What is the natural flow of state given the current position?

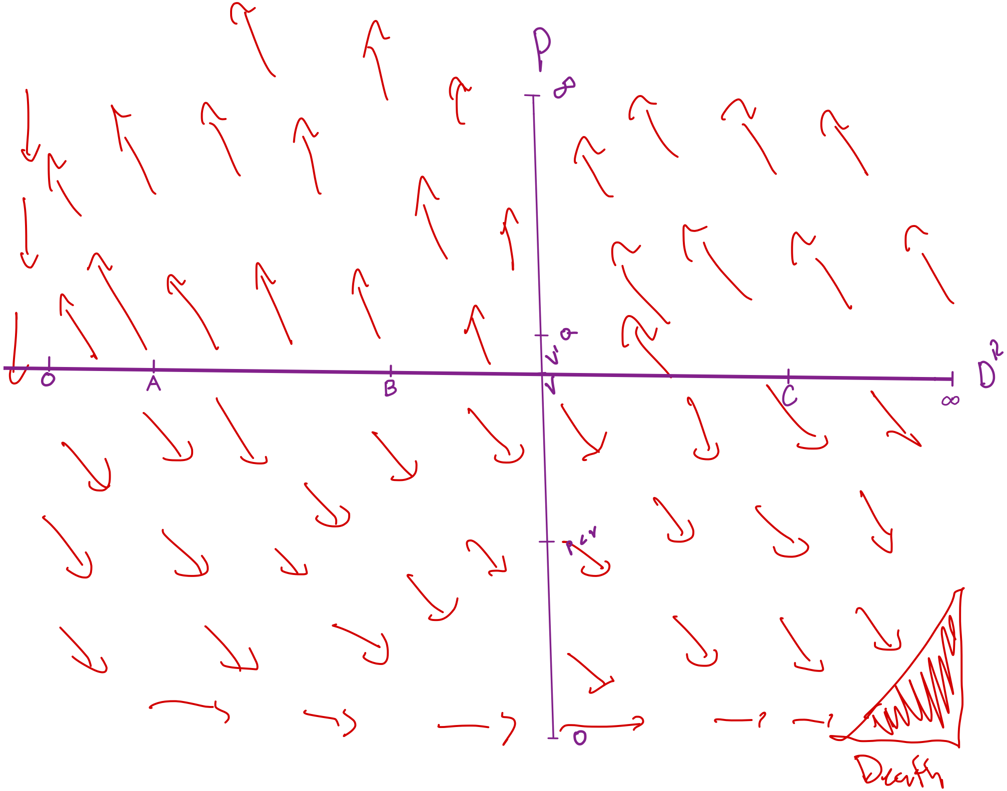
(2.2) What tools does Beanstalk have available to perform peg maintenance?

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

(2.4) How should Beanstalk use these tools in practice?

(2.1) What is the natural flow of state given the current position?

<u>PRICE x DEBT LEVEL Natural Flow (*i.e.*, the Problem</u>)



Natural flow: given shape and scale of PRICE x DEBT LEVEL what is the natural tendency of the system?

Any debt based stablecoin system is going to tend to pump (*i.e.*, high P, low D^R). The higher the pump, the greater the dump (*i.e.*, low P, high D^R).

The natural tendency to buy when P > V and dump when P < V is addressed by the Silo. further discussion of the Silo's Stalk and Seed system will occur upon analysis of the future seed gauge and generalized convert systems later in this document.

(2.2) What tools does Beanstalk have available to perform peg maintenance?

PRICE x DEBT LEVEL Peg Maintenance

The 4 peg maintenance tools currently available to Beanstalk are to (1) mint Beans, (2) mint Soil, (3) change the Maximum Temperature and (4) sell newly minted Beans directly on the market.

Mint Beans and Soil
Because price is the dominant factor in determining the force acting upon Beanstalk, and Beanstalk as a stablecoin protocol is trying to create stability of P around V, responding primarily to P is the first step in the peg maintenance model.

The time-weighted average deltaB is a good source of the marginal amount of Beans that needed to be bought or sold on average for the time-weighted average price to be equal to V and a good proxy for PRICE wrt V.

When P > V (*i.e.*, deltaB was positive on average over the previous Season), Beanstalk can *mint deltaB Beans* in an attempt to increase the marginal supply of Beans on the market and lower the average price to V in the next Season.

When P < V (*i.e.*, deltaB was negative on average over the previous Season), Beanstalk can *mint -deltaB Soil* in attempt to borrow Beans from the market in an attempt to decrease the marginal supply and raise the average price to V in the next Season.

In order to measure Demand for Soil every Season Beanstalk mints Soil every Season, even if P > V. In such cases Beanstalk mints only enough Soil such that the maximum Pods issues that Season is the same amount of Pods that were Harvested at the beginning of the Season. Therefore, the relationship that D^R can only decrease when P > V is preserved.

(2.2) What tools does Beanstalk have available to perform peg maintenance?

PRICE x DEBT LEVEL Peg Maintenance Pt 2

Change the Maximum Temperature
Based on the shape of the space, it is clear that when setting the Maximum Temperature the most important thing is to not issue too much debt such that you send Beanstalk into a hole/death spiral.

Given the efficiency of price discovery in the Morning Auction, there is little reason to raise or lower the Maximum Temperature aggressively. However, the Maximum Temperature should still be set at a reasonable rate to create the most optimal market conditions for Beanstalk as a borrower.

Beanstalk changes the Maximum Temperature based on its position, direction and acceleration relative to ideal equilibrium (i.e., (V, D^R*)).

However, given the complexity of accurately estimating the shape and scale of the space in practice, determining the amount to change the Maximum Temperature is currently more art than science.

Basic principles in changing Maximum Temperature:

- Change should be small because the potential for major change creates less efficient market for Soil
- Should err on the side of being too low instead of too high because Beanstalk would rather deviate its peg than overissue debt and enter a death spiral.

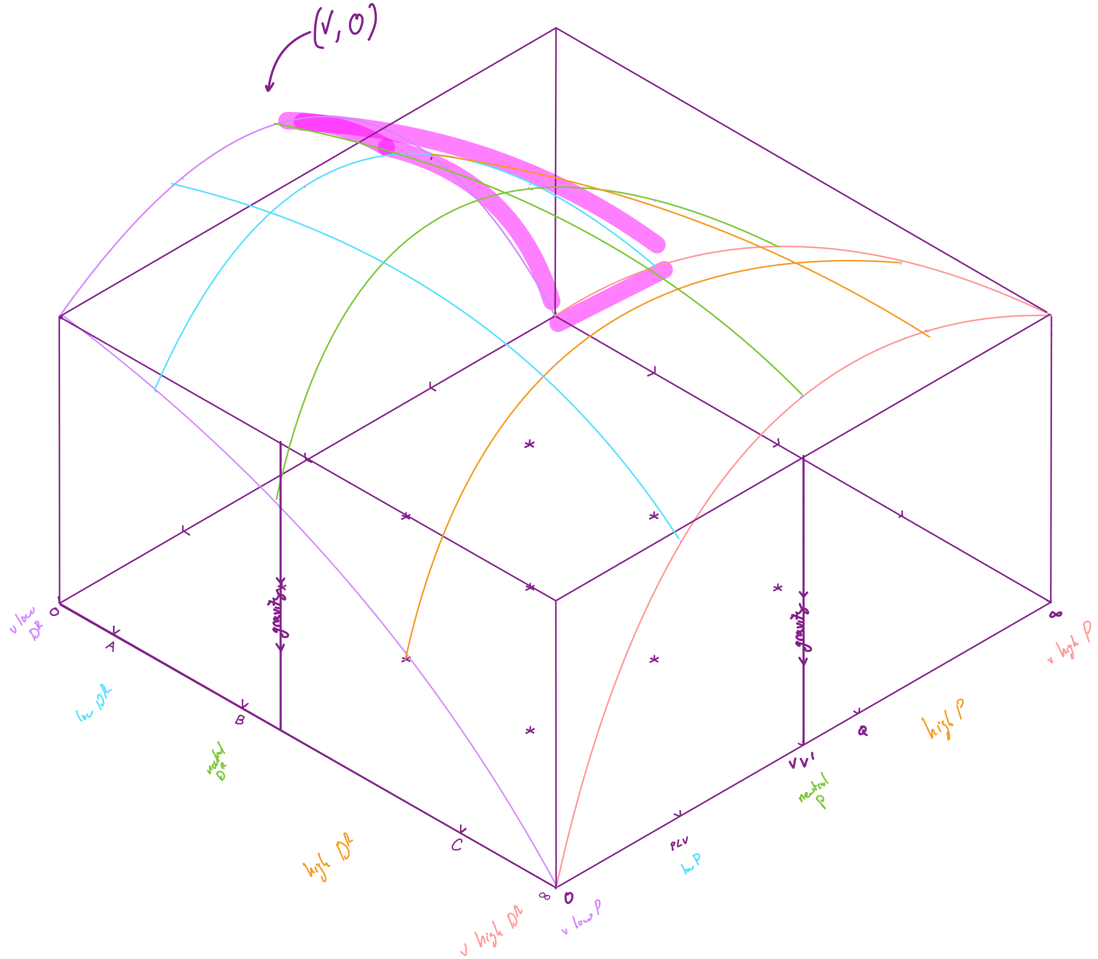
(2.2) What tools does Beanstalk have available to perform peg maintenance?

Flood

the flood is useful when Beanstalk is at risk of overheating and getting stuck in the v high P, v low D^R corner.

a flood returns the position of Beanstalk to (V, 0) and applies downward force wrt P through the minting and sale of new Beans to return the Bean price to V and the complete repayment of all outstanding debt. however, this push can only be applied once per debt cycle. repeated floods can still keep P close to V and discourage inorganic demand. it should also reset the Maximum Temperature.

in practice, flood uses all 3 tools at once to limits the momentum Beanstalk will enter a debt cycle into.



(2.3) How do these tools affect Beanstalk’s position along various axes in theory?

As a design principle, any time Beanstalk mints Beans it pays off debt, such that attempts to decrease PRICE through minting are always coupled with a decrease in DEBT LEVEL.

As a debt based stablecoin, Beanstalk can increase deltaB by issuing debt, such that attempts to increase PRICE through the Field are always coupled with an increase in DEBT LEVEL.

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

Beanstalk can always decrease price for decreased debt level through the issuance of newly minted Beans.

In the case where there is demand for Soil at some price, Beanstalk can always increase price for increased debt level through the issuance of Pods.

Raising the Temperature means Beanstalk is willing to issue more debt to raise the deltaB (and therefore Price) by the same amount.

Lowering the Temperature means Beanstalk is willing to issue less debt debt to raise the deltaB (and therefore Price) by the same amount.

The exact rate of tradeoff between PRICE AND DEBT LEVEL is unclear in practice.

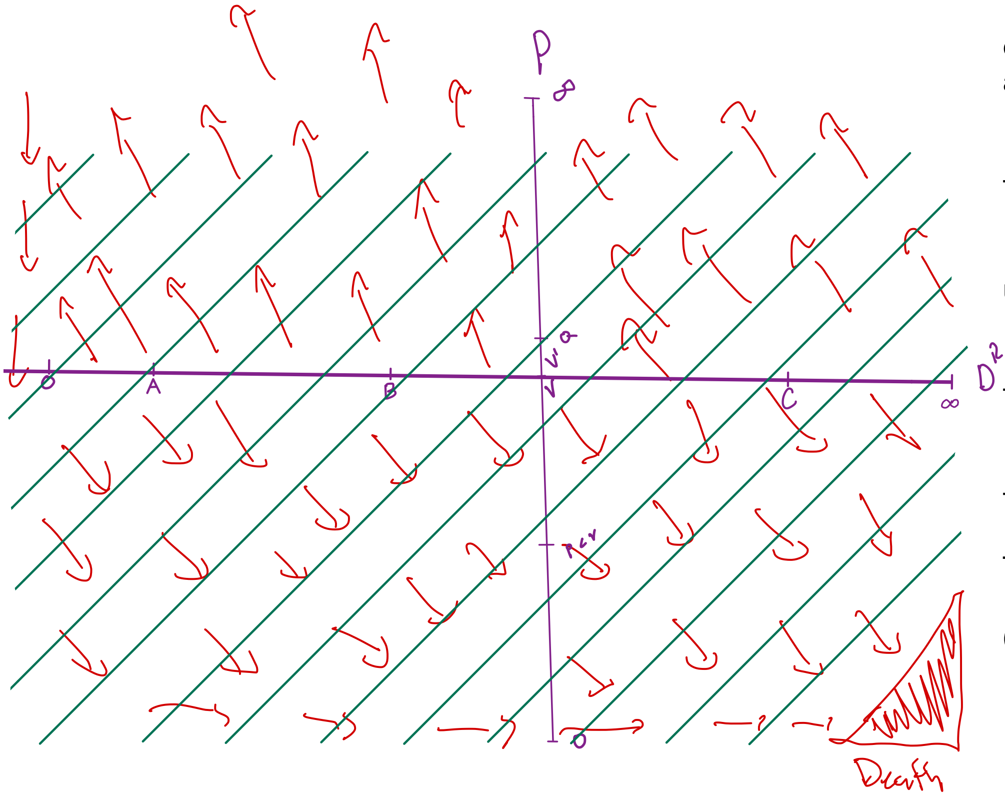
The question then becomes under what circumstances should Beanstalk encourage the tradeoff and in which direction?

How Beanstalk changes the Maximum Temperature is its answer.

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

PRICE x DEBT LEVEL Position Tradeoff

▨ estimated directional tradeoff between PRICE and DEBT LEVEL



Minting Beans is applying force down and to the left along the green lines.

Minting Soil in an environment where there is demand for it is applying force up and to the right along the green lines.

Minting more (less) Beans increases (decreases) the downward force along both axes.

Minting more (less) Soil increases (decreases the upward force along both axes.

Decreasing the Maximum Temperature increases the slope of the green lines.

Increasing the Maximum Temperature decreases the slope of the green lines.

The larger the amount of increase (decrease) in the Maximum Temperature, the larger the increase (decrease) in slope.

(2.4) How should Beanstalk use these tools in practice?

Natural Questions:

Q: Should Beanstalk mint more Beans at higher or lower debt levels?

A: Minting extra is likely to kill inorganic demand entirely but may also lead to a dip below peg which would require extra debt issuance and therefore have the reverse effect than intended. Doesn't make sense at higher levels, could make sense at lower levels. Minting less Beans than deltaB is likely to spur inorganic demand, which should not be encouraged at any time.

Q: Should Beanstalk be more or less aggressive at raising the Temperature when it is close to a potential death spiral.

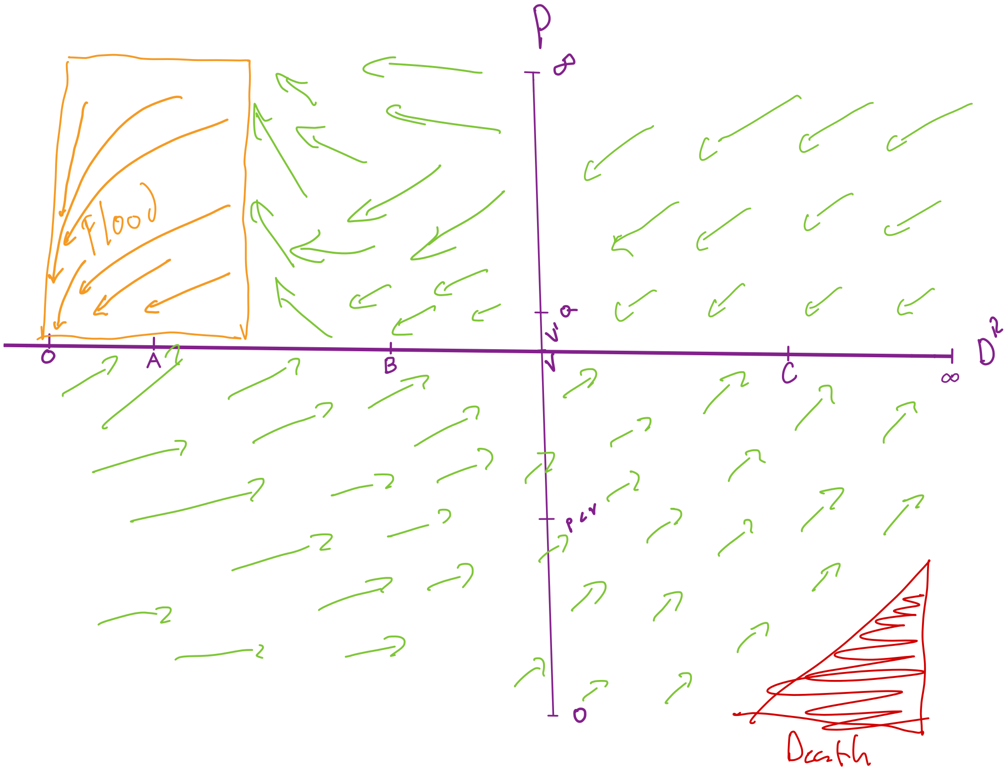
A: I would argue it should be it shouldn't be more aggressive at raising the Maximum Temperature because it is better to wait out the storm than issue too much debt at too high of an interest rate. Slowly raising the interest rate is a more long term oriented solution. In fact, it may even be beneficial to be more aggressive with Maximum Temperature changes with lower debt levels.

Q: In a perfect world, how would Beanstalk use the Field and the Flood to affect the position of Beanstalk across the PRICE x DEBT LEVEL space to maximize peg maintenance?

A: See next graphic.

(2.4) How should Beanstalk use these tools in practice?

PRICE x DEBT LEVEL effect of efficient Field and Flood



Whenever P < V there is upward price pressure of P through the issuance of debt. promptly issuing debt to repeg for a reasonable interest rate is the simplest way for Beanstalk to maintain long term stability. However, doing so requires a consistently active market for primary issuance of new Pods.

Above peg there is a tendency towards repayment of debt until Beanstalk enters a Flood.

The Flood returns the state of Beanstalk to P = V and D^R = 0.

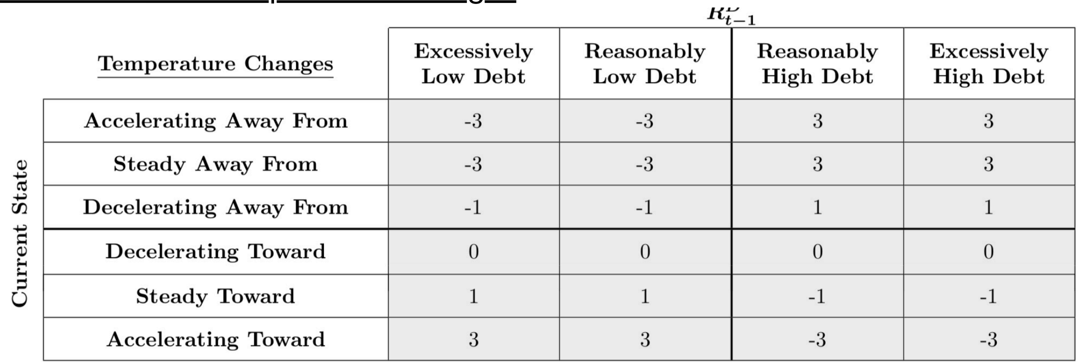
(2.4) How should Beanstalk use these tools in practice?

Peg Maintenance Response Summary

1. Mint deltaB Beans if time-weighted average deltaB is positive.
2. Mint negative deltaB Soil if time-weighted average deltaB is negative and enough Soil to keep issue the most amount of Pods without increasing the outstanding Pods over the course of the Season if the time-weighted average deltaB is positive.
3. Change Maximum Temperature based on position, direction and acceleration wrt ideal equilibrium.

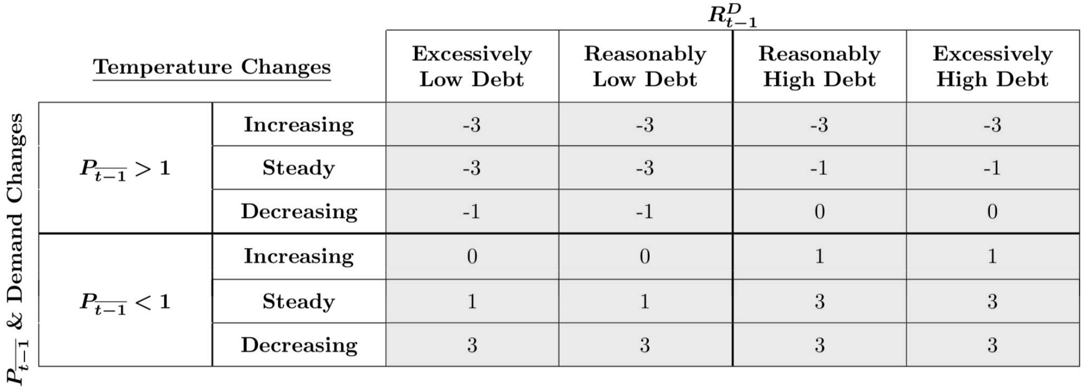
(2.4) How should Beanstalk use these tools in practice?

Beanstalk Maximum Temperature Changes



	Temperature Changes	R^D_{t-1} Excessively Low Debt	Reasonably Low Debt	Reasonably High Debt	Excessively High Debt
Current State	Accelerating Away From	-3	-3	3	3
	Steady Away From	-3	-3	3	3
	Decelerating Away From	-1	-1	1	1
	Decelerating Toward	0	0	0	0
	Steady Toward	1	1	-1	-1
	Accelerating Toward	3	3	-3	-3

Figure 12: Maximum Temperature Changes From Current State and R^D_{t-1}



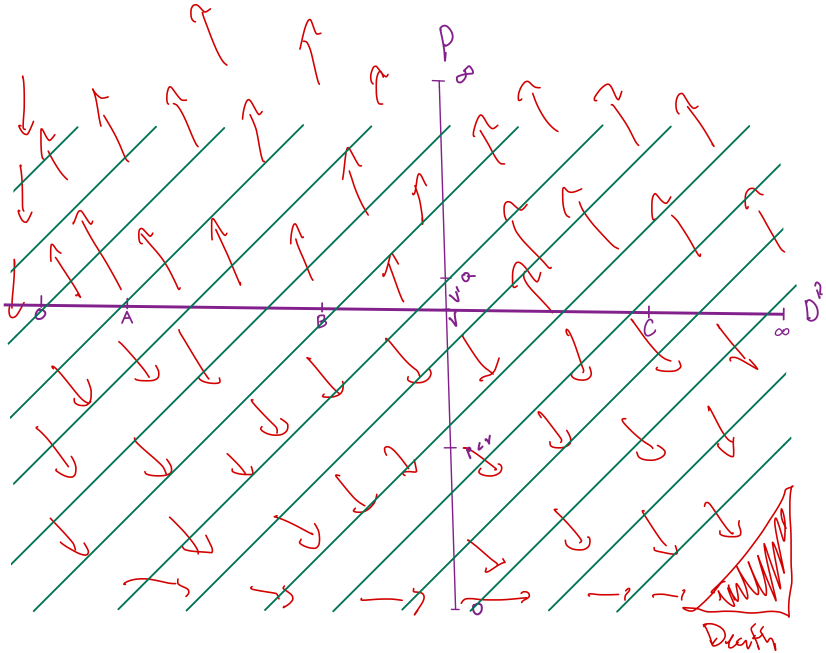
		Temperature Changes	R^D_{t-1} Excessively Low Debt	Reasonably Low Debt	Reasonably High Debt	Excessively High Debt
$P_{\overline{t-1}}$ & Demand Changes	$P_{\overline{t-1}} > 1$	Increasing	-3	-3	-3	-3
		Steady	-3	-3	-1	-1
		Decreasing	-1	-1	0	0
	$P_{\overline{t-1}} < 1$	Increasing	0	0	1	1
		Steady	1	1	3	3
		Decreasing	3	3	3	3

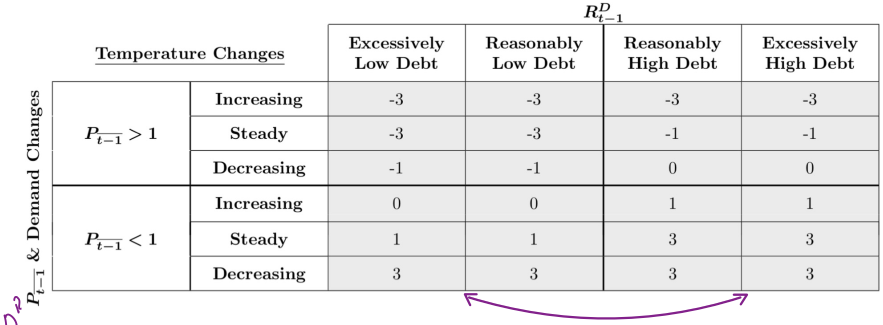
Figure 13: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1}

(2.4) How should Beanstalk use these tools in practice?

Beanstalk Maximum Temperature Changes

▨ estimated directional tradeoff between PRICE and DEBT LEVEL





$P_{\overline{t-1}}$ & Demand Changes	Temperature Changes	R^D_{t-1} Excessively Low Debt	Reasonably Low Debt	Reasonably High Debt	Excessively High Debt
$P_{\overline{t-1}} > 1$	Increasing	-3	-3	-3	-3
	Steady	-3	-3	-1	-1
	Decreasing	-1	-1	0	0
$P_{\overline{t-1}} < 1$	Increasing	0	0	1	1
	Steady	1	1	3	3
	Decreasing	3	3	3	3

Figure 13: Maximum Temperature Changes From $P_{\overline{t-1}}$, Demand for Soil Changes and R^D_{t-1}

This study would argue that it is better to raise the Maximum Temperature faster with lower debt levels to quickly repeg at the cost of more debt issuance, and raise the Maximum Temperature more slowly with higher debt levels due to the desire to protect the long term health of the system at the cost of downside price deviations.

Future Work within the Current PRICE x DEBT LEVEL Model

- Research into proper way to estimate A, B and C;
- Research into optimal way to set D^R^Lower, D^R*, D^R^Upper wrt A, B and C;
- Research into measuring the acceleration of price as some function of deltaB over time;
- Research into measuring demand for Soil in increasingly manipulation resistant fashion via some threshold;
- Research into some natural opposite of Q along the PRICE dimension;
- Research into a “mini flood” where price is brought back down immediately to Q when it is above it at Sunrise, but no additional harvest beyond the minting needed to bring P to Q. unclear who should get the extra mint in terms of debt holders vs stalk holders;
- Research into measuring position, direction and acceleration over more than just one Season over one or more axes;
- Research into a future peg maintenance model that could use relative position, direction and acceleration relative to the two holes instead of relative to ideal equilibrium;

auto compounding flood

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

(1.5) how should Beanstalk evaluate its position, direction and acceleration at the beginning of each Season in practice?

With a deeper understanding of the PRICE x DEBT LEVEL space and the current Beanstalk peg maintenance mechanism, let’s expand the scope of our inquiry to the other two combinations of 2 dimensions and the new Seed Gauge system:

- PRICE x L2SR
- DEBT LEVEL x L2SR

Price x L2SR

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

The following questions can be used to better understand the interplay between PRICE and L2SR:

(1.4.a) how does PRICE act differently upon itself at various L2SR?

(1.4.b) how does L2SR act differently upon itself at various PRICES?

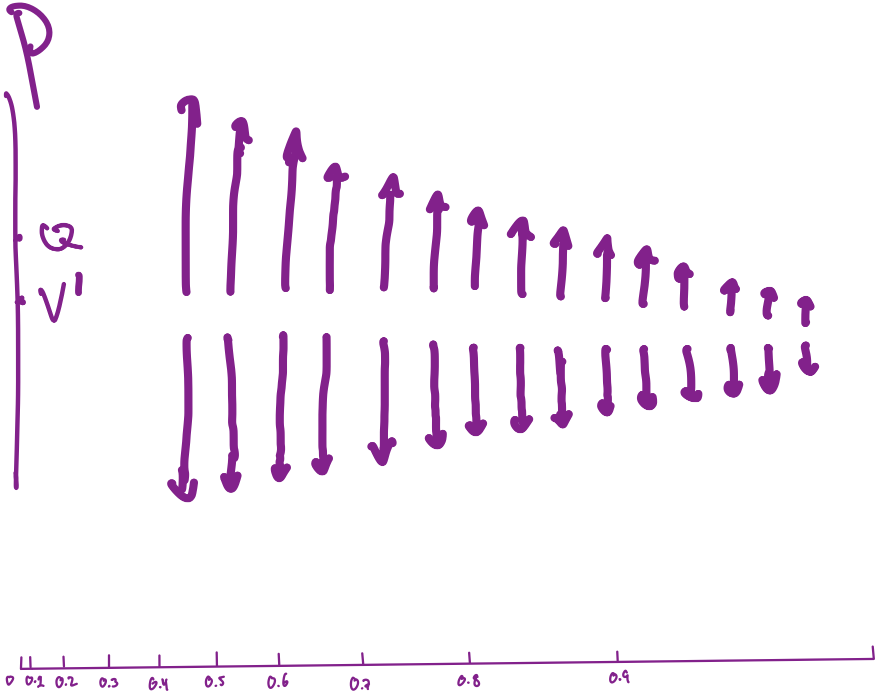
(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x L2SR?

(1.4.d) given the shape and scale of the 2 dimensional space PRICE x L2SR, what can Beanstalk infer about its state in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

(1.4.a) how does PRICE act differently upon itself at various L2SR?

Effect of L2SR on P

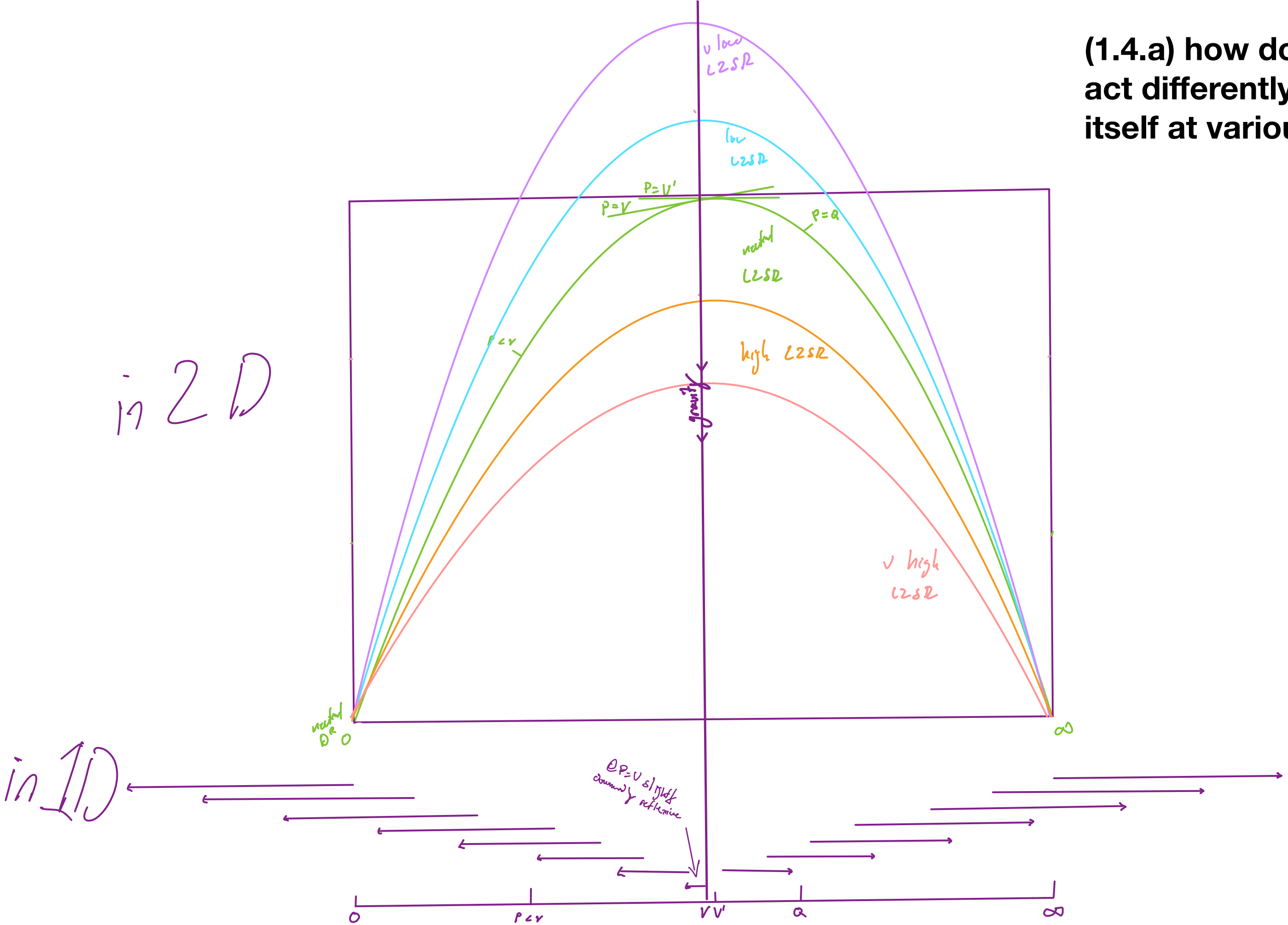
- as L2SR increases P becomes less reflexive because there is more liquidity.
- as L2SR decreases P becomes more reflexive because there is less liquidity.



natural effect of L2SR on P

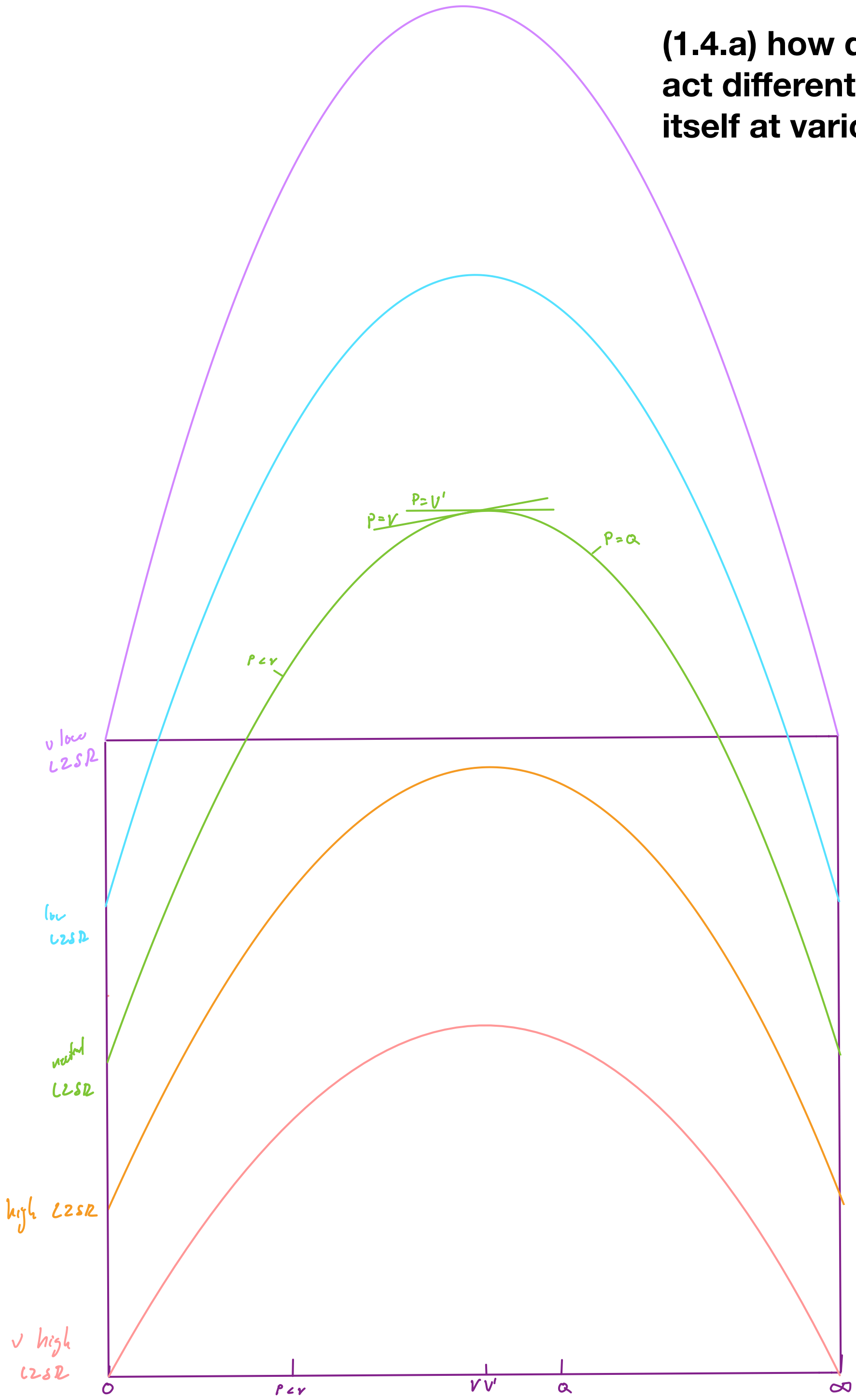
Natural Shape, Scale of P at various L2SR

(1.4.a) how does PRICE act differently upon itself at various L2SR?



Natural Shape, Scale of P at various L2SR

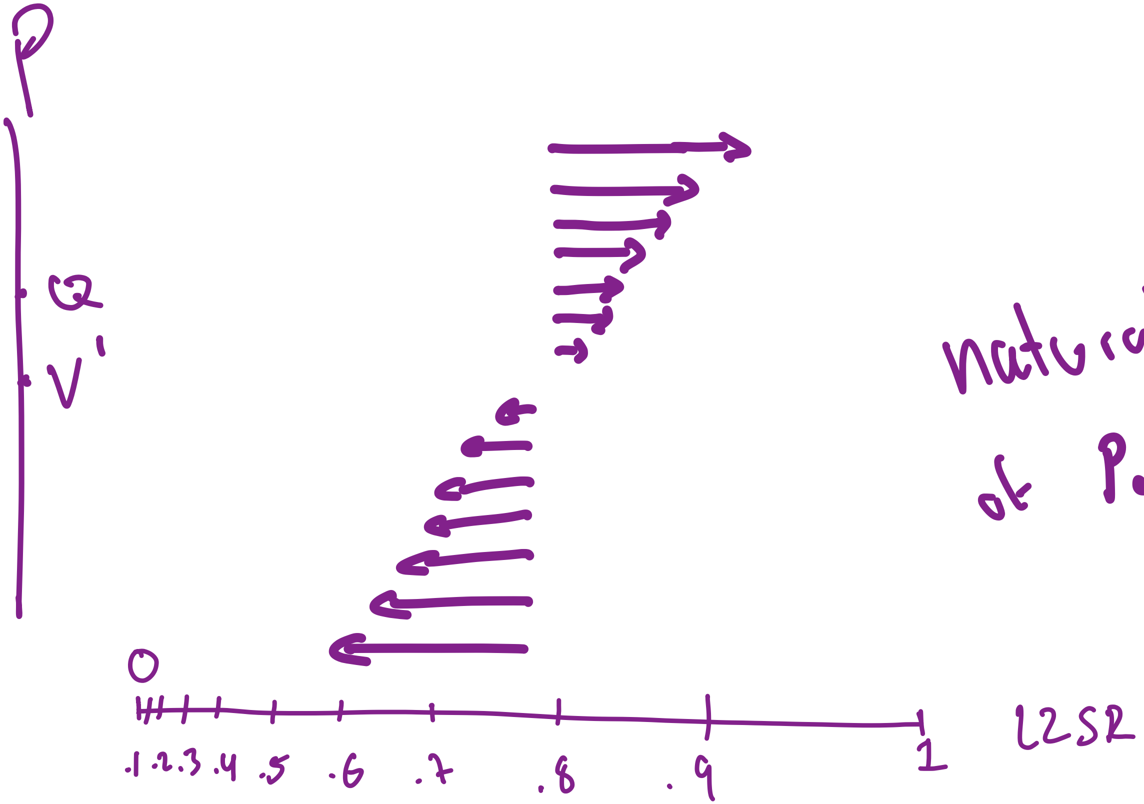
(1.4.a) how does PRICE act differently upon itself at various L2SR?



(1.4.b) how does L2SR act differently upon itself at various PRICES?

Effect of P on L2SR

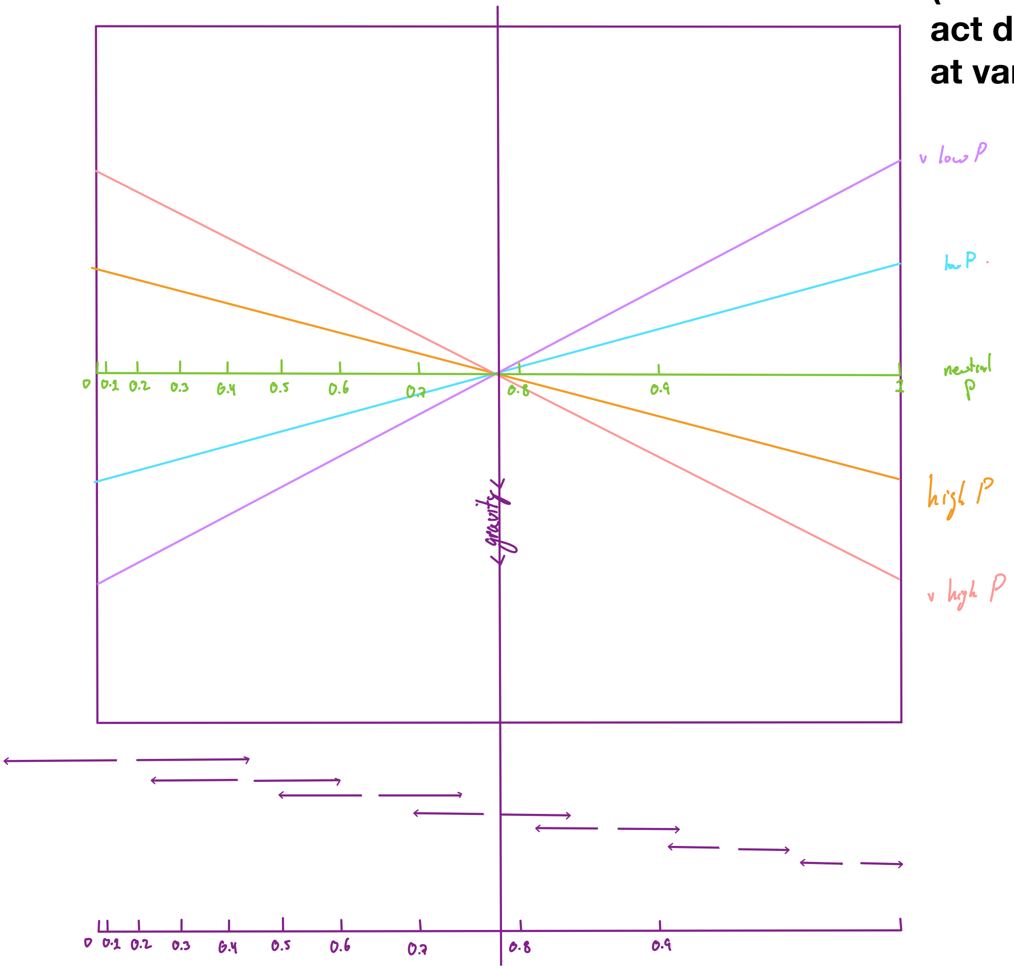
- the higher P is above V’, the faster L2SR increases as more people convert Bean to LP
- the lower P is below V’, the faster L2SR decreases as more people convert LP to Bean



natural effect of P on L2SR

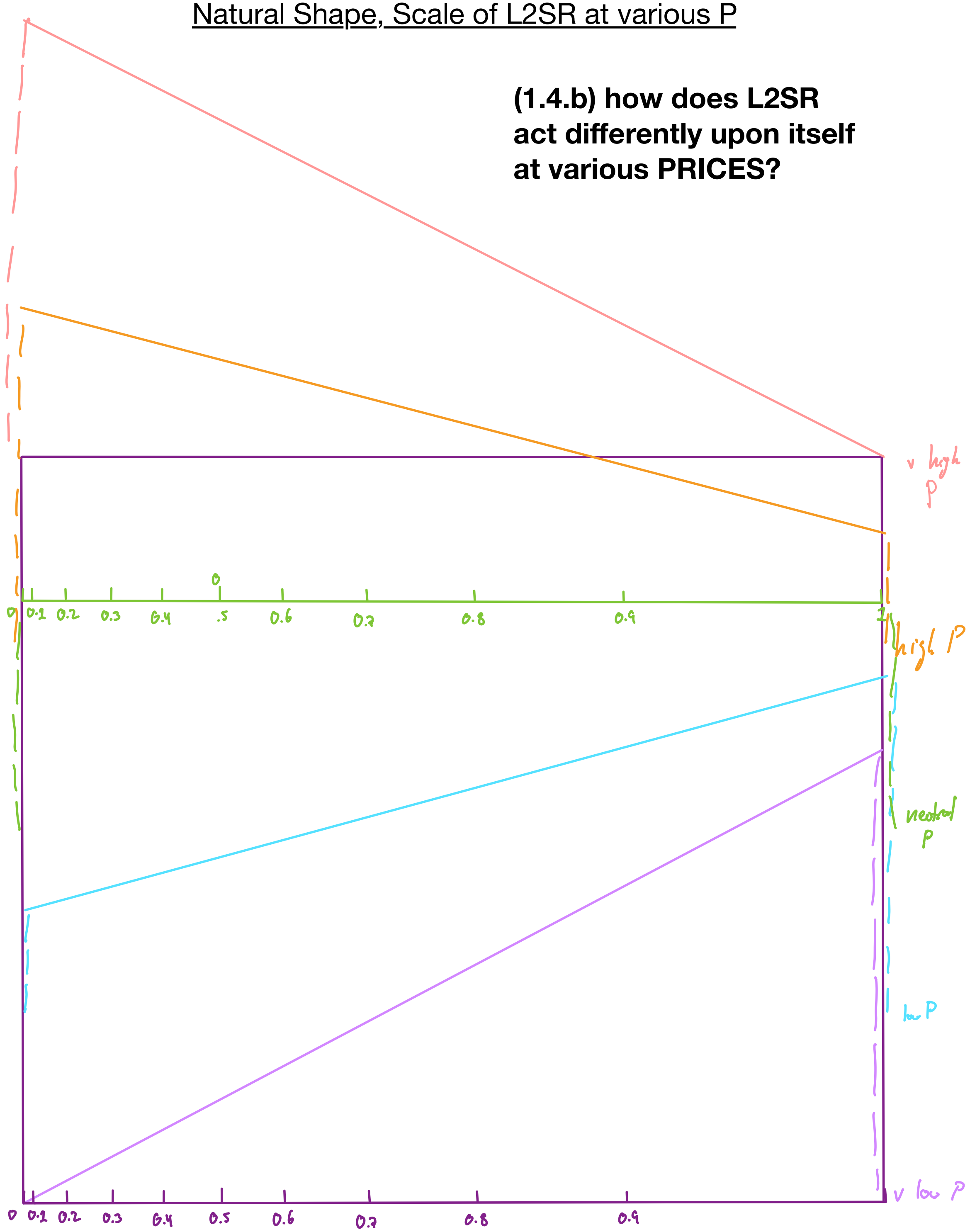
Natural Shape, Scale of L2SR at various P

(1.4.b) how does L2SR act differently upon itself at various PRICES?



Natural Shape, Scale of L2SR at various P

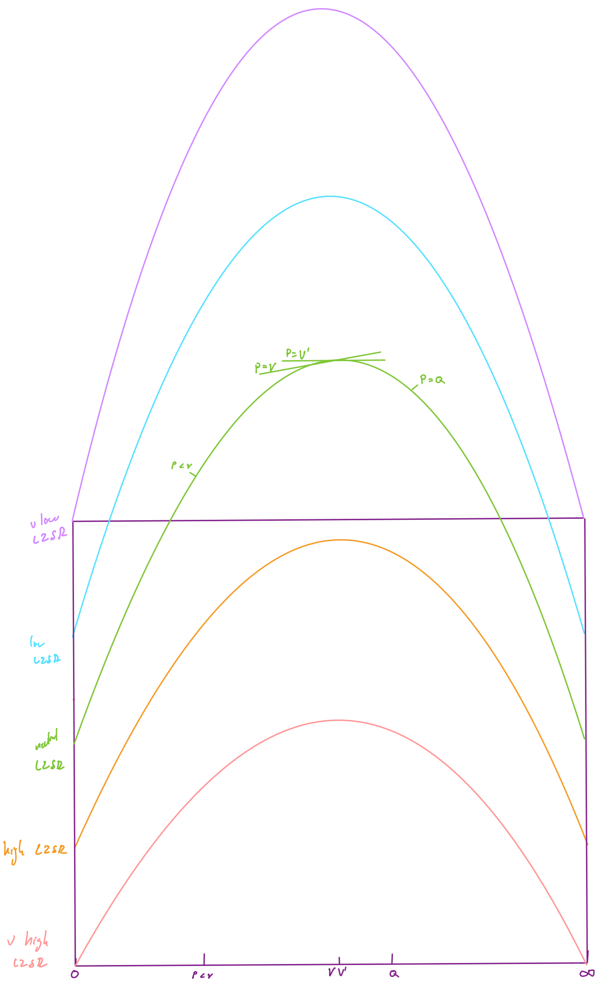
(1.4.b) how does L2SR act differently upon itself at various PRICES?



(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x L2SR?

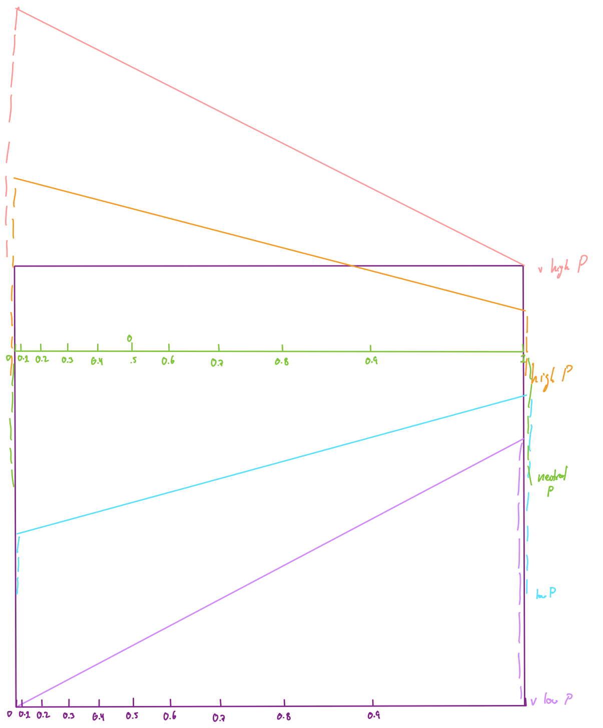
key idea: stitch the shapes and scales of the 2 axes together into a single space

Natural Shape, Scale of P at various L2SR



X

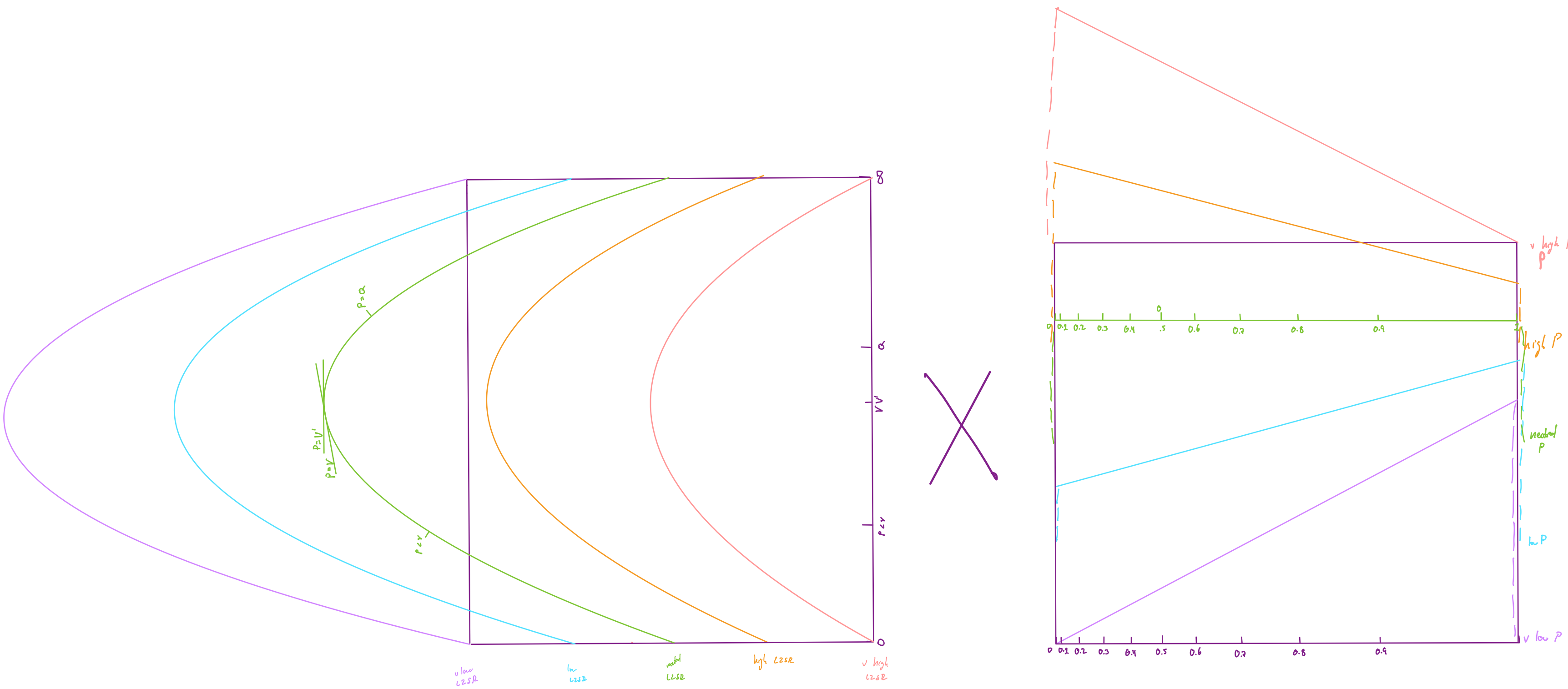
Natural Shape, Scale of L2SR at various P



(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x L2SR?

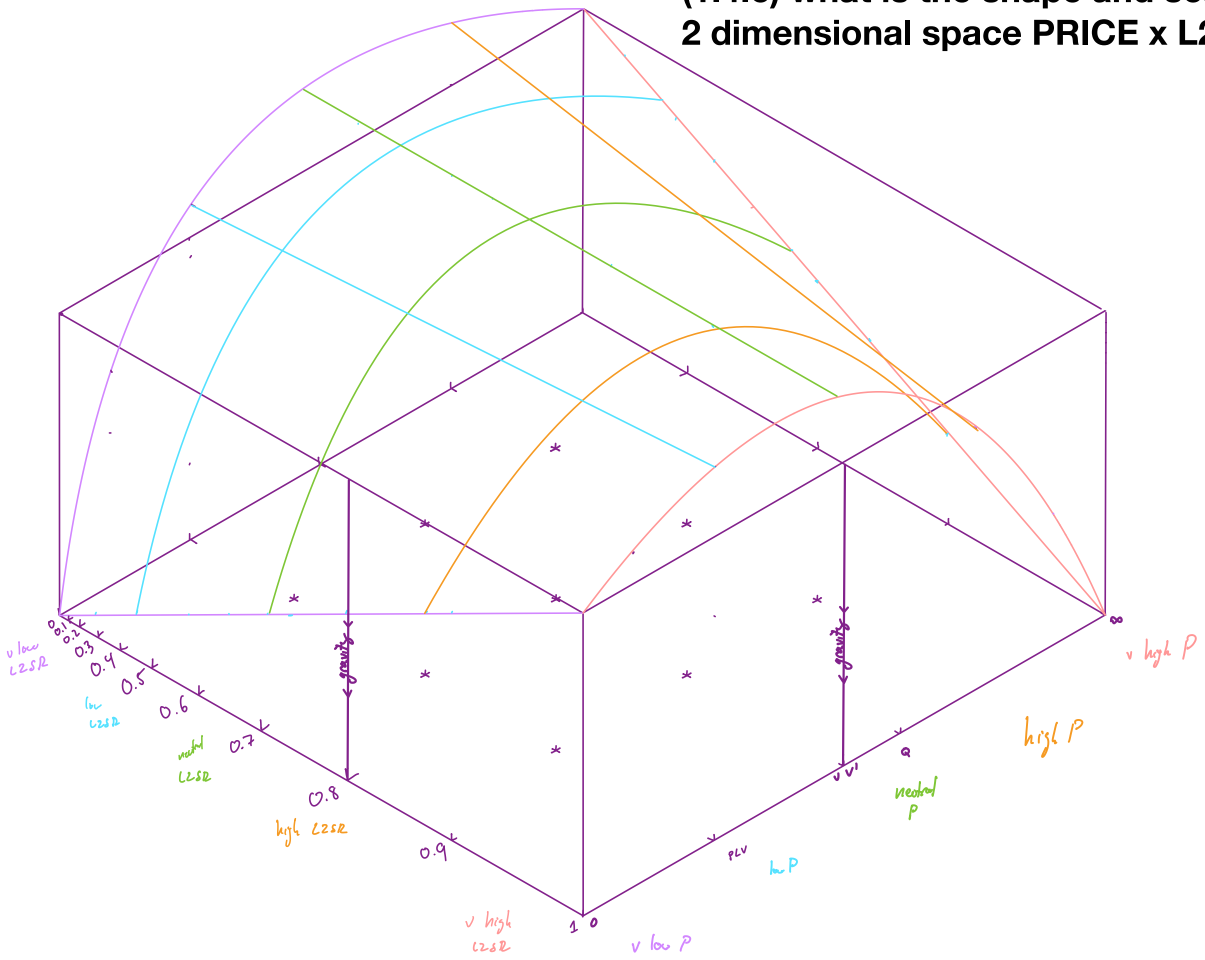
Natural Shape, Scale of P at various L2SR

Natural Shape, Scale of L2SR at various P



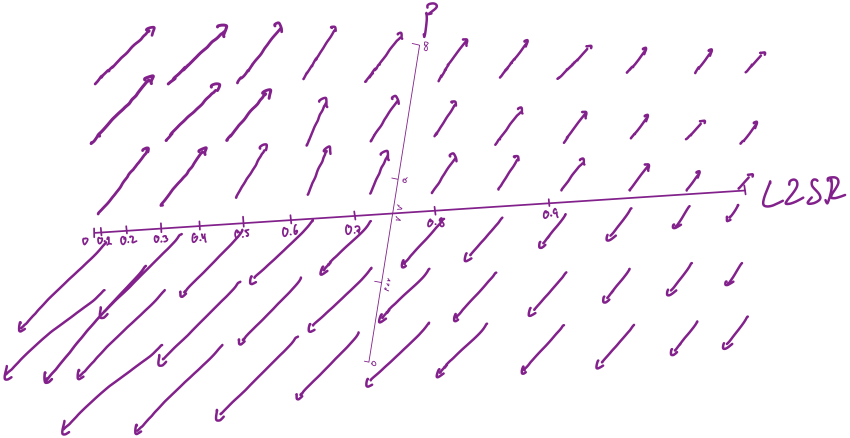
Natural Shape, Scale of P x L2SR in 3D

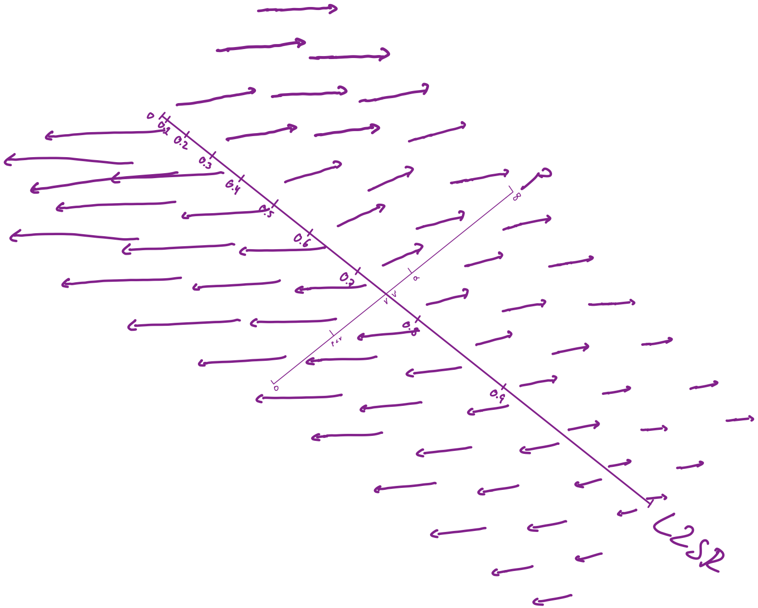
(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x L2SR?



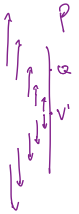
Natural Shape, Scale of P x L2SR IN 2D

(1.4.c) what is the shape and scale of the 2 dimensional space PRICE x L2SR?

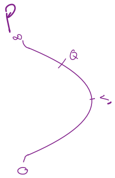




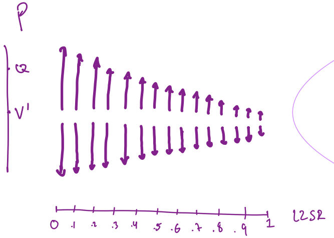
Direction in 2 Dimensions Cheat Sheet: Price x L2SR



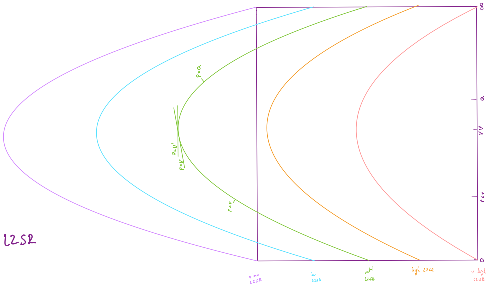
natural effect of P on P @ various P



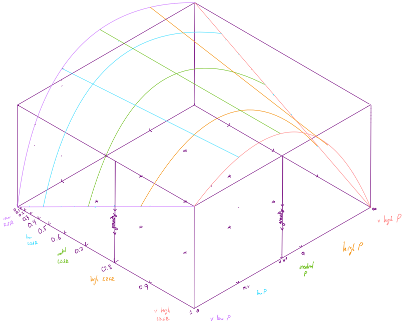
natural shape of P



natural effect of L2SR on P



natural shape of P @ various L2SR



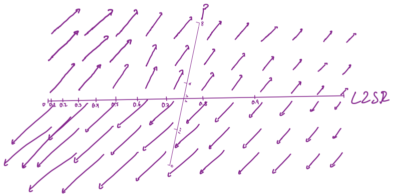
natural shape, scale of P x L2SR in 3D

price

- highly reflexive
  - as P increases past V', P trends upwards faster
  - as P decreases past V', P trends downward faster
- change in directional effect @ V'

L2SR on P

- as L2SR increases P becomes less reflexive
- as L2SR decreases P bevcomes more reflexive



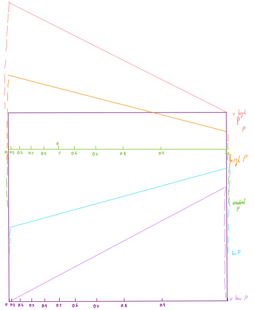
natural shape of P x L2SR in 2D

P on L2SR

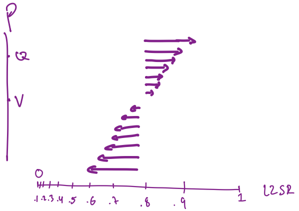
- when higher P above V the faster the L2SR increases because of Converts down
- when lower P below V the faster the L2SR decreases because of Converts up

L2SR

- negatively correlated with volatility
  - as L2SR increases, L2SR voltatility decreases
  - as L2SR decreases, L2SR volatility increases

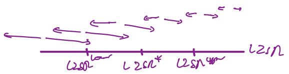


natural shape of L2SR @ various P w/ scaled L2SR



natural effect of P on L2SR

natural shape of L2SR (flat and squishes closer to 0)



natural effect of L2SR on L2SR

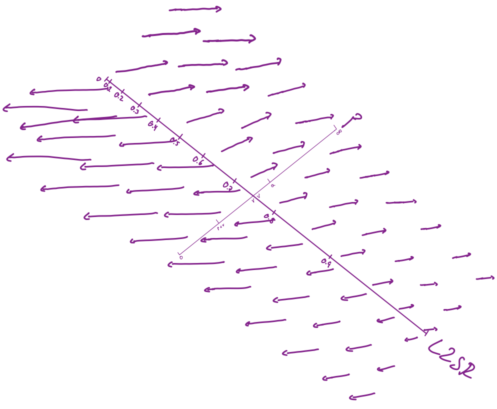
Shape and Scale of PRICE x L2SR

Shape: highly reflexive, global maximum along P = V’, max concavity at (v low P, v low L2SR) and (v high P, v high L2SR)

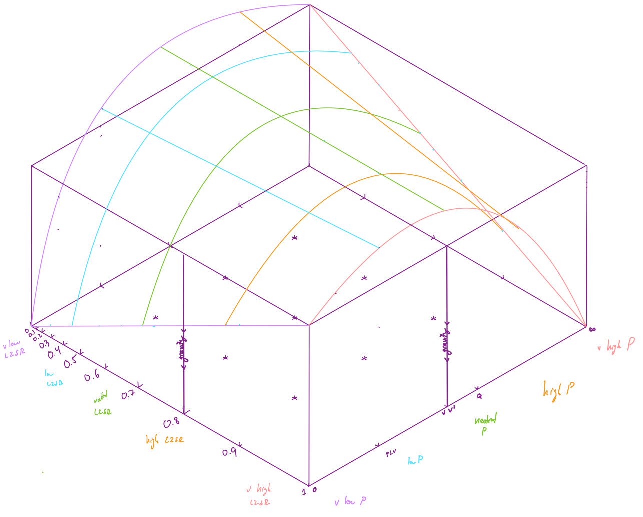
- 1st derivative: negative slope wrt P if P > V’, positive slope if P < V’. negative slope wrt L2SR if P > V’, positive slope if P < V’.
    - price:
        - if P > V’ there is upward price pressure. L2SR is correlated with upward price pressure.
        - if P < V’ there is downward price pressure. L2SR is inversely correlated with downward price pressure.
    - L2SR:
        - if P > V’ there is upward pressure on L2SR. P is correlated with upward L2SR pressure.
        - if P < V’ there is downward pressure on L2SR. P is inversely correlated with downward L2SR pressure.
- 2nd derivative: concave throughout. global maximum at (neutral P, neutral L2SR) OR (V, W) (*i.e.*, somewhere on the line P = V).
    - price:
        - if P > V’, L2SR is uncorrelated with the rate of increase in upwards price pressure.
        - if P < V’, L2SR is uncorrelated with the rate of increase in downwards price pressure.
    - L2SR:
        - if P > V’, P is correlated with the rate of increase in upward L2SR pressure.
        - if P < V’, P is inversely correlated with the rate of increase in downwards L2SR pressure.
- 3rd derivative: gravitational “holes” at (v low P, v low L2SR) and (v high P, v high L2SR)

Scale: relationship between volatility of price and liquidity levels is highly influenced in practice by the willingness to convert.

- 1st derivative: expansion if far from (V’,W), compression if close to (V’, W). as L2SR increases (decreases), volatility decreases (increases).
    - price:
        - if P far from V’ there is expansion
        - if P is close to V’ there is compression
    - L2SR:
        - if L2SR increases -> expansion
        - if L2SR decreases -> compression
- 2nd derivative : more compression closer to (V’, W), more expansion further from (V’, W). the more L2SR increases (decreases), the more expansion (compression)
    - price:
        - as P moves further away from V’ there is more expansion
        - as P moves closer to V’ there is more compression
    - L2SR:
        - same increase of L2SR @ higher L2SR -> less L2SR volatility -> more expansion
        - same decrease of L2SR @ lower L2SR -> more L2SR volatility -> more compression



in 2D



in 3D

intuitively, the difficulty of balancing a ball on this surface is far from a simple task. nonetheless, this is one of 3 cuts over which Beanstalk has to optimize under the new peg maintenance task (*i.e.*, post Seed Gauge).

while it may seem like a good thing to be in the corner with v high price and v high L2SR, it requires a lot of momentum to get out of the hole. therefore, when Beanstalk’s price finally comes down in such a scenario its momentum is likely to carry it into a major debt cycle with low prices and decreasing liquidity levels, getting stuck in the other corner. therefore, it is best to avoid either hole.

better understanding the shape of this space can help answer questions like ‘why doesn’t Beanstalk prioritize soft peg maintenance over paying premiums for hard peg maintenance?’ Clearly, avoiding extremely low levels of liquidity, beyond which there is no return, is a priority over short term peg maintenance.

the relationship between price and L2SR becomes clear by looking at this space: the L2SR is much more likely to increase when P > V and much more likely to decrease when P < V.

note, the actual scale of the L2SR along these curves is unknown in practice. this makes Beanstalk’s peg maintenance task more complex.

Direction and Acceleration of PRICE x L2SR

Direction:

key idea: **L2SR is naturally increasing when P > V and decreasing when P < V. Price is overwhelmingly important compared to L2SR in its effect on the shape of space wrt both price and L2SR.**

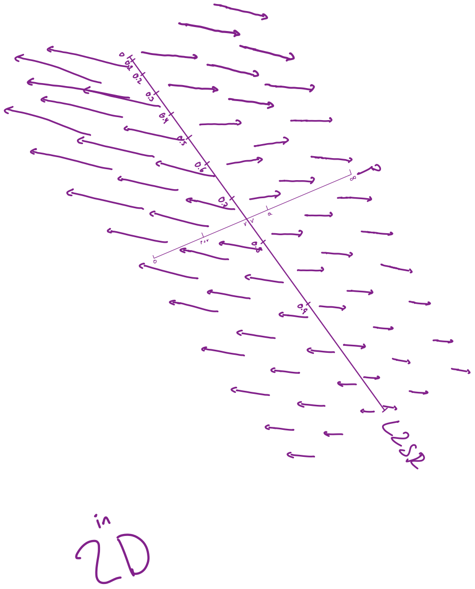
- if P > V' the directional force of P x L2SR is positive wrt P
- if P > V' the directional force of P x L2SR is positive wrt L2SR
- if P < V' the direction force of P x L2SR is negative wrt P
- if P < V' the directional force of P x L2SR is negative wrt L2SR

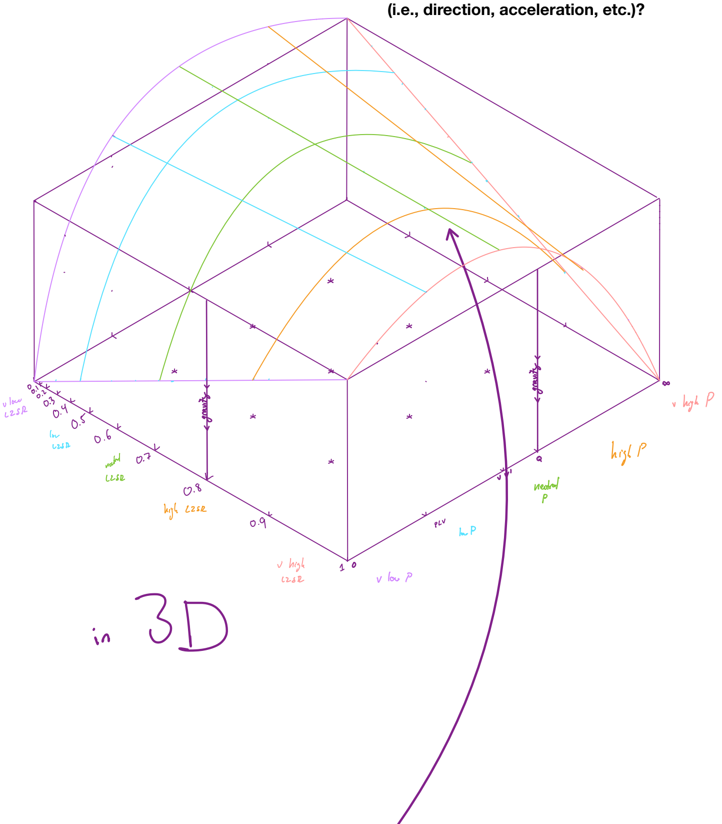
Acceleration:

key idea: **Beanstalk naturally accelerates towards (v low P, v low L2SR) if P < V and towards (v high P, v high L2SR) if P > V'**

- if P > V' the directional force of P x L2SR on P is increasing in magnitude as P increases.
- if P < V' the directional force of P x L2SR on P is increasing in magnitude as P decreases.
- if P > V' the directional force of P x L2SR on L2SR is increasing in magnitude as P increases.
- if P < V' the directional force of P x L2SR on L2SR is increasing in magnitude as P decreases.

(1.4.d) given the shape and scale of the 2 dimensional space PRICE x L2SR, what can Beanstalk infer about its state in terms of derivatives of position (i.e., direction, acceleration, etc.)?





Ideal equilibrium lies somewhere near this line. In practice, properly setting L2SR^Lower, L2SR* and L2SR^Upper is very difficult.

Before we can evaluate the interplay between the 3 axes and the peg maintenance system that will go live upon the implementation of the Seed Gauge system we must first understand the DEBT LEVEL x L2SR space.

Debt Level

x

L2SR

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

The following questions can be used to better understand the interplay between DEBT LEVEL and L2SR:

(1.4.a) how does DEBT LEVEL act differently upon itself at various L2SR?

(1.4.b) how does L2SR act differently upon itself at various DEBT LEVEL?

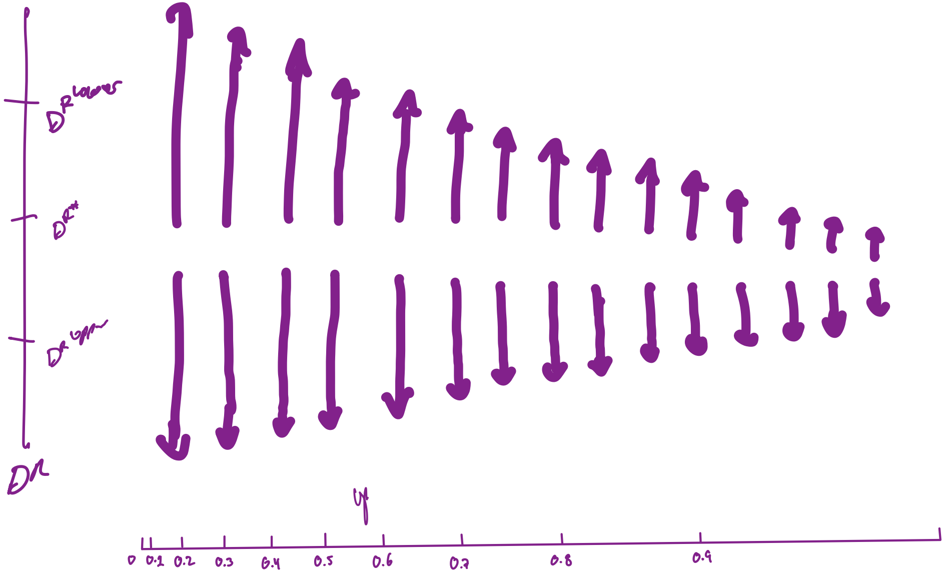
(1.4.c) what is the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR?

(1.4.d) given the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR, what can Beanstalk infer about its state in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

(1.4.a) how does DEBT LEVEL act differently upon itself at various L2SR?

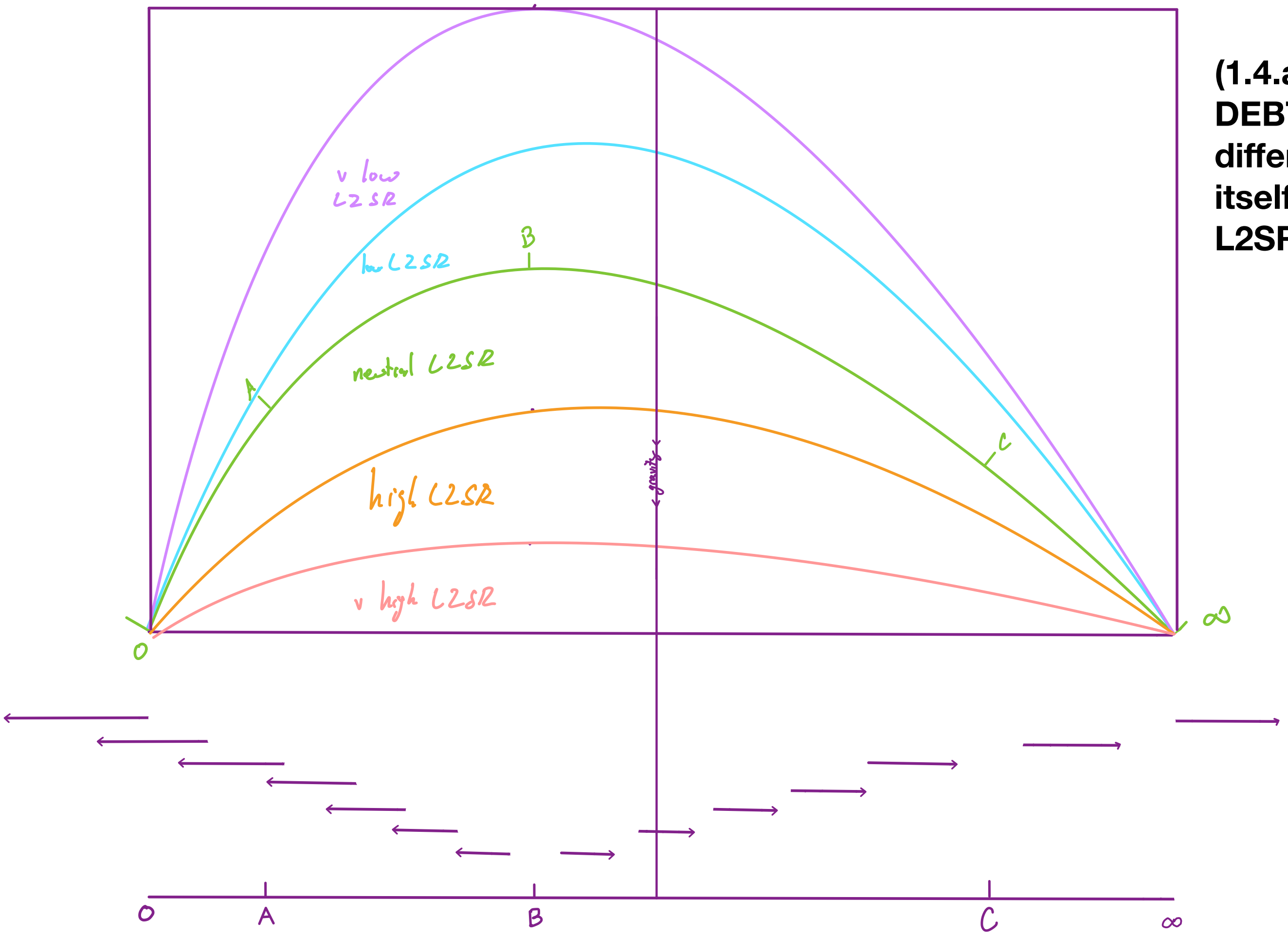
<u>Effect of L2SR on D^R</u>

- as L2SR increases D^R becomes less reflexive due to increased conversions preventing the need to issue as many Beans for a given inflow.
- as L2SR decreases D^R becomes more reflexive due to decreased conversions creating more need to issue as much Soil for a given outflow.



natural effect of L2SR on D^R

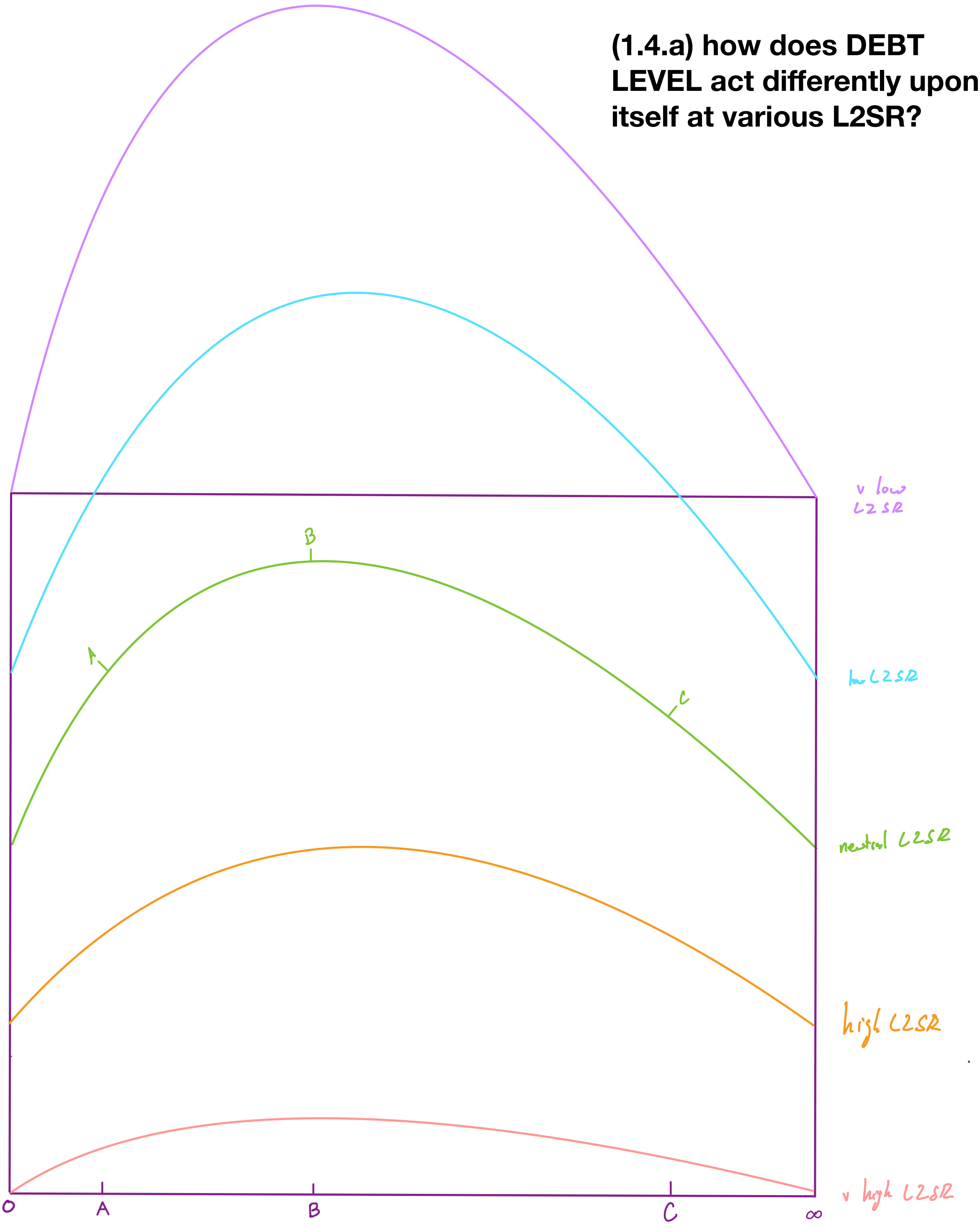
Natural Shape, Scale of D^R at various L2SR



(1.4.a) how does DEBT LEVEL act differently upon itself at various L2SR?

Natural Shape, Scale of D^R at various L2SR

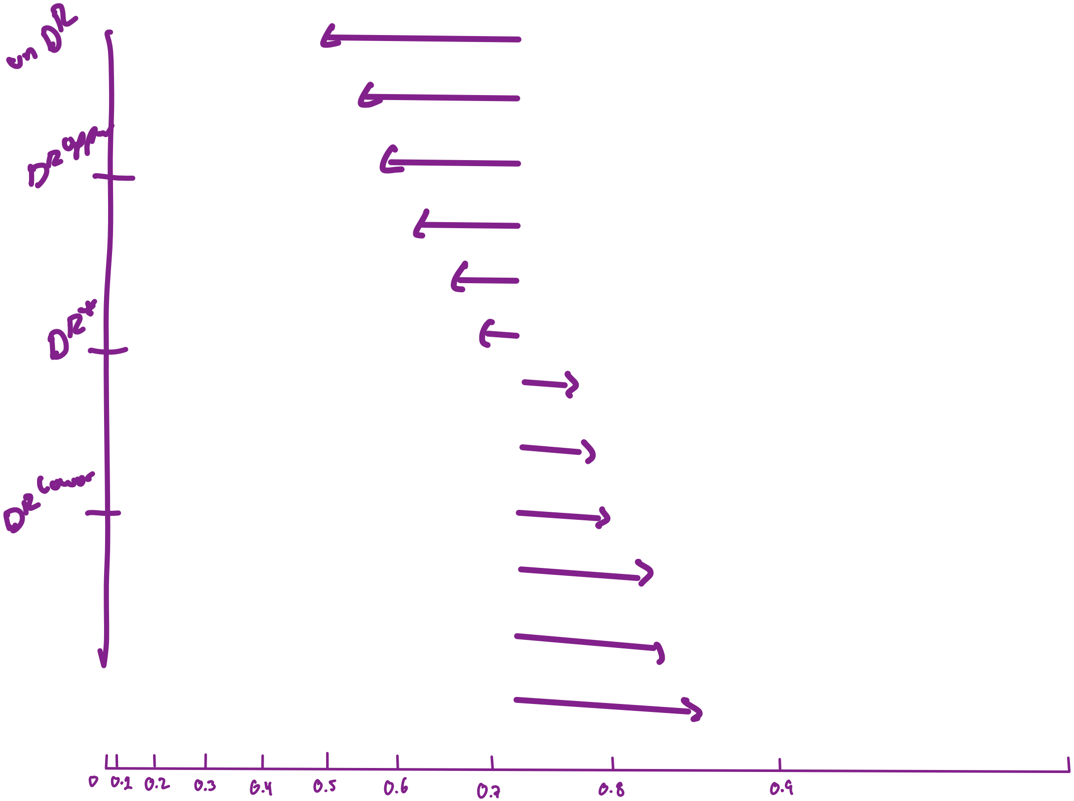
(1.4.a) how does DEBT LEVEL act differently upon itself at various L2SR?



(1.4.b) how does L2SR act differently upon itself at various DEBT LEVEL?

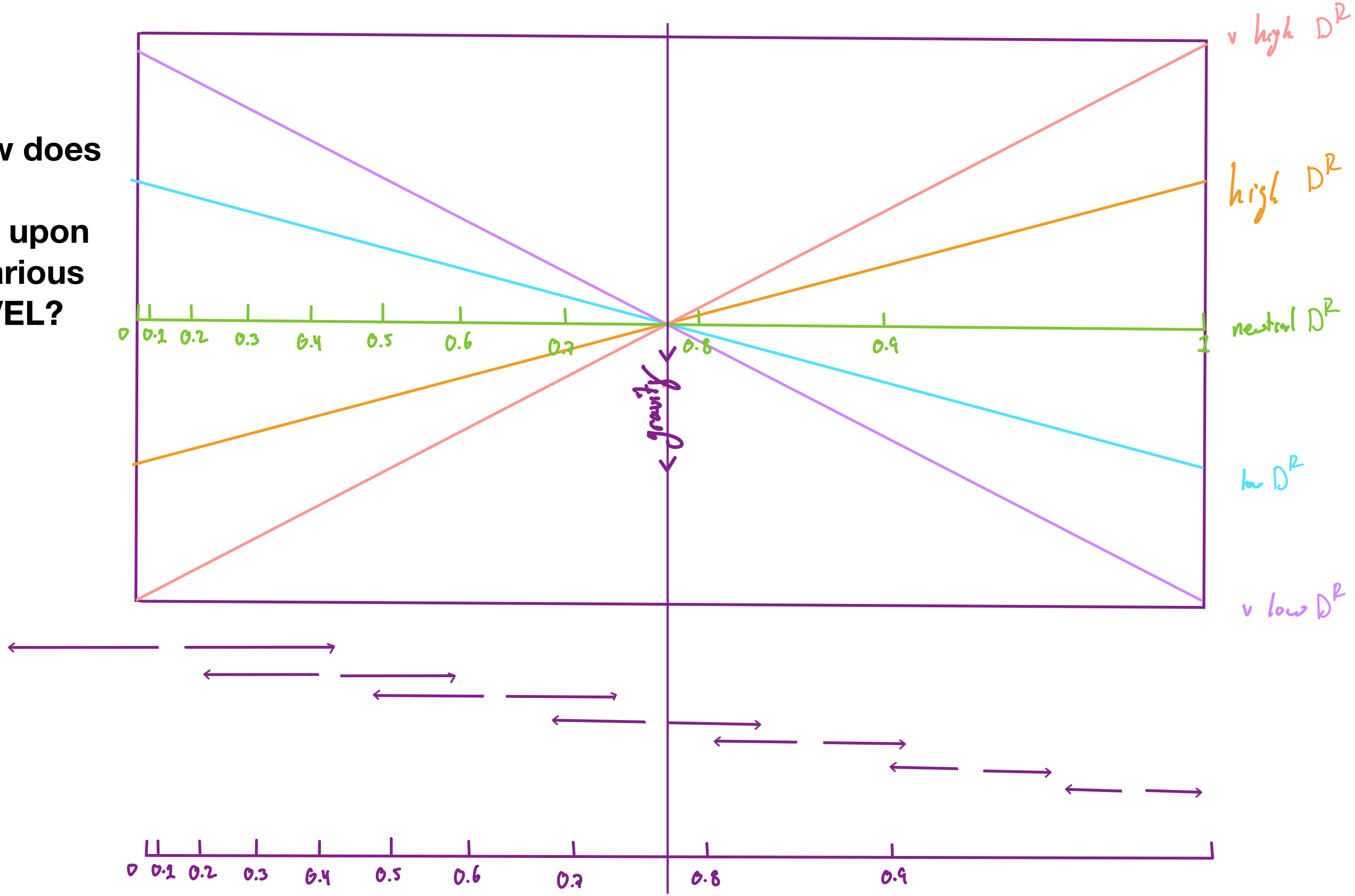
Effect of D^R on L2SR

- the higher D^R is the easier it is for the L2SR to decrease and the harder it is for the L2SR to increase.
- the lower D^R is the harder it is for the L2SR to decrease and the easier it is for the L2SR to increase.



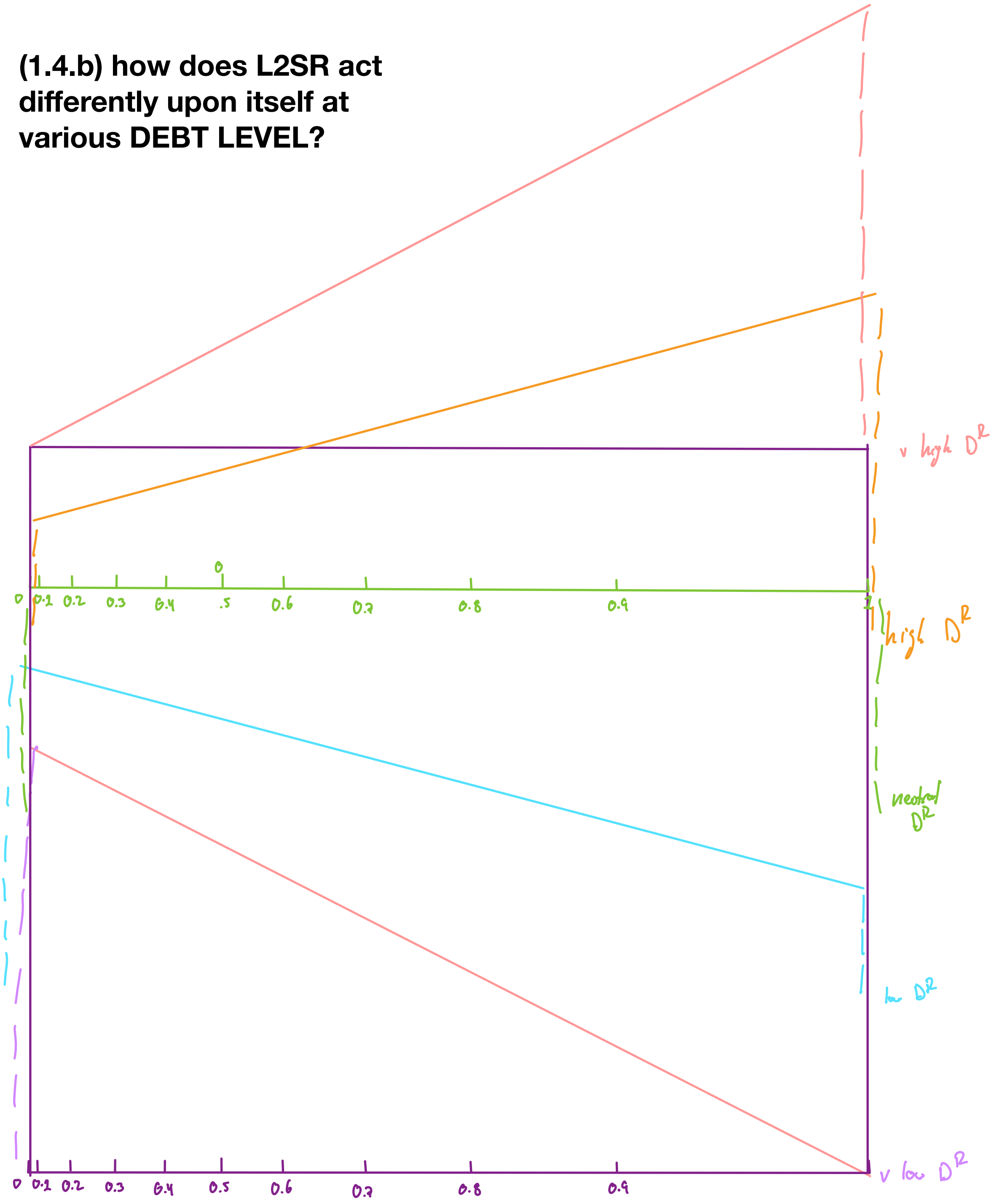
Natural Shape, Scale of L2SR at various D^R

(1.4.b) how does L2SR act differently upon itself at various DEBT LEVEL?



Natural Shape, Scale of L2SR at various D^R

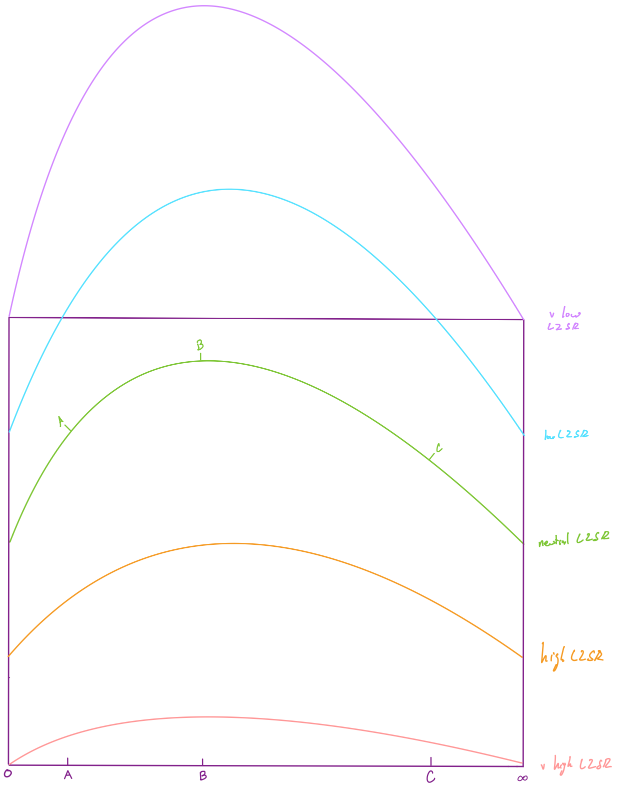
(1.4.b) how does L2SR act differently upon itself at various DEBT LEVEL?



(1.4.c) what is the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR?

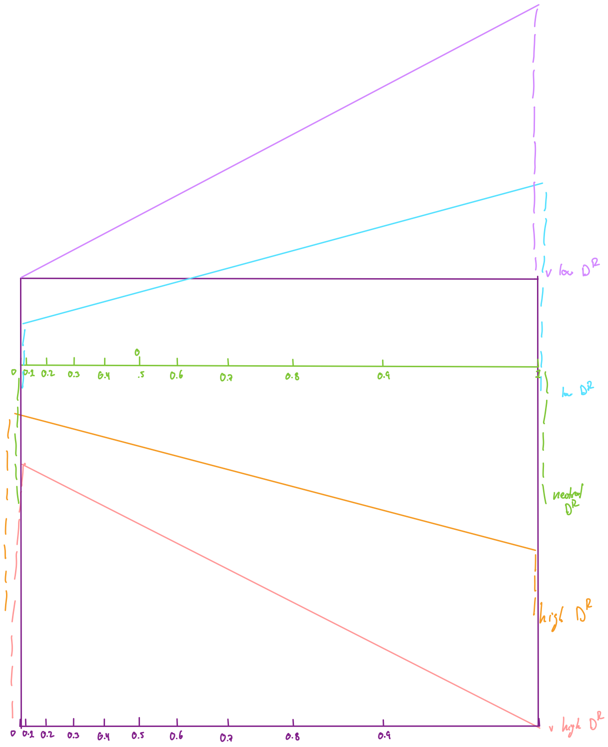
key idea: stitch the shapes and scales of the 2 axes together into a single space

Natural Shape, Scale of D^R at various L2SR



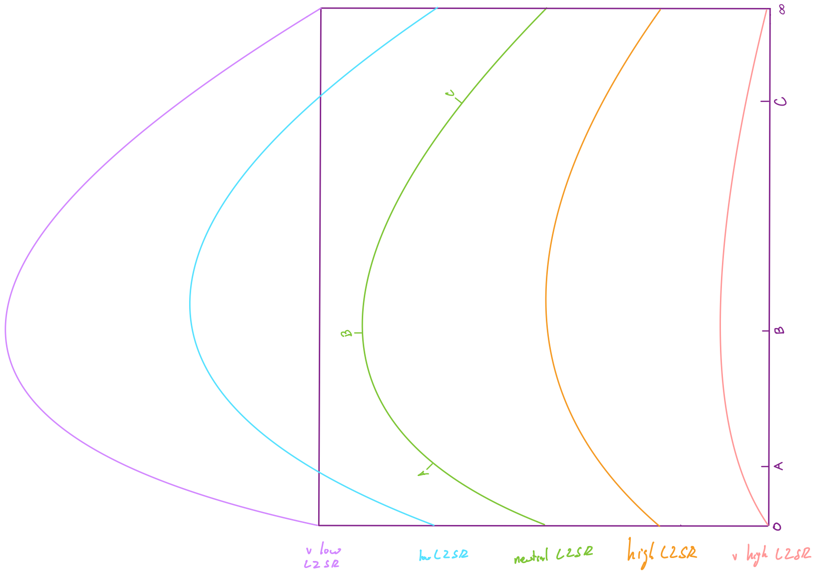
×

Natural Shape, Scale of L2SR at various D^R



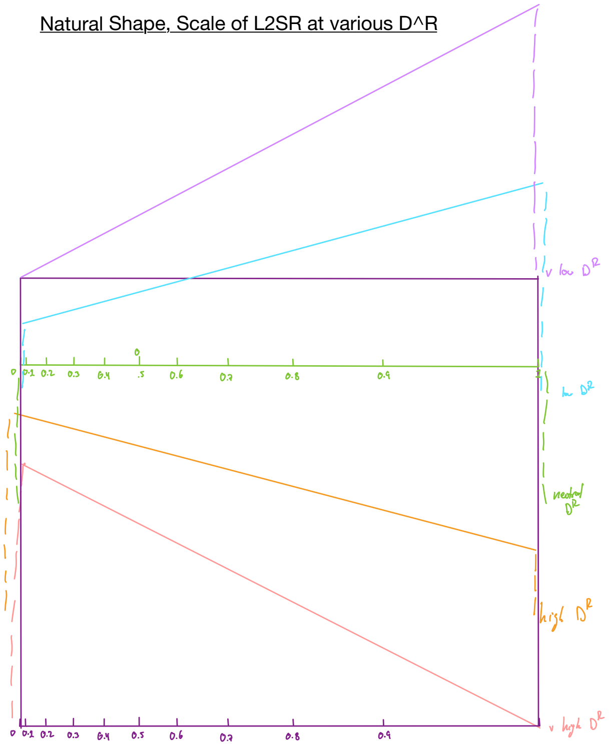
(1.4.c) what is the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR?

Natural Shape, Scale of D^R at various L2SR



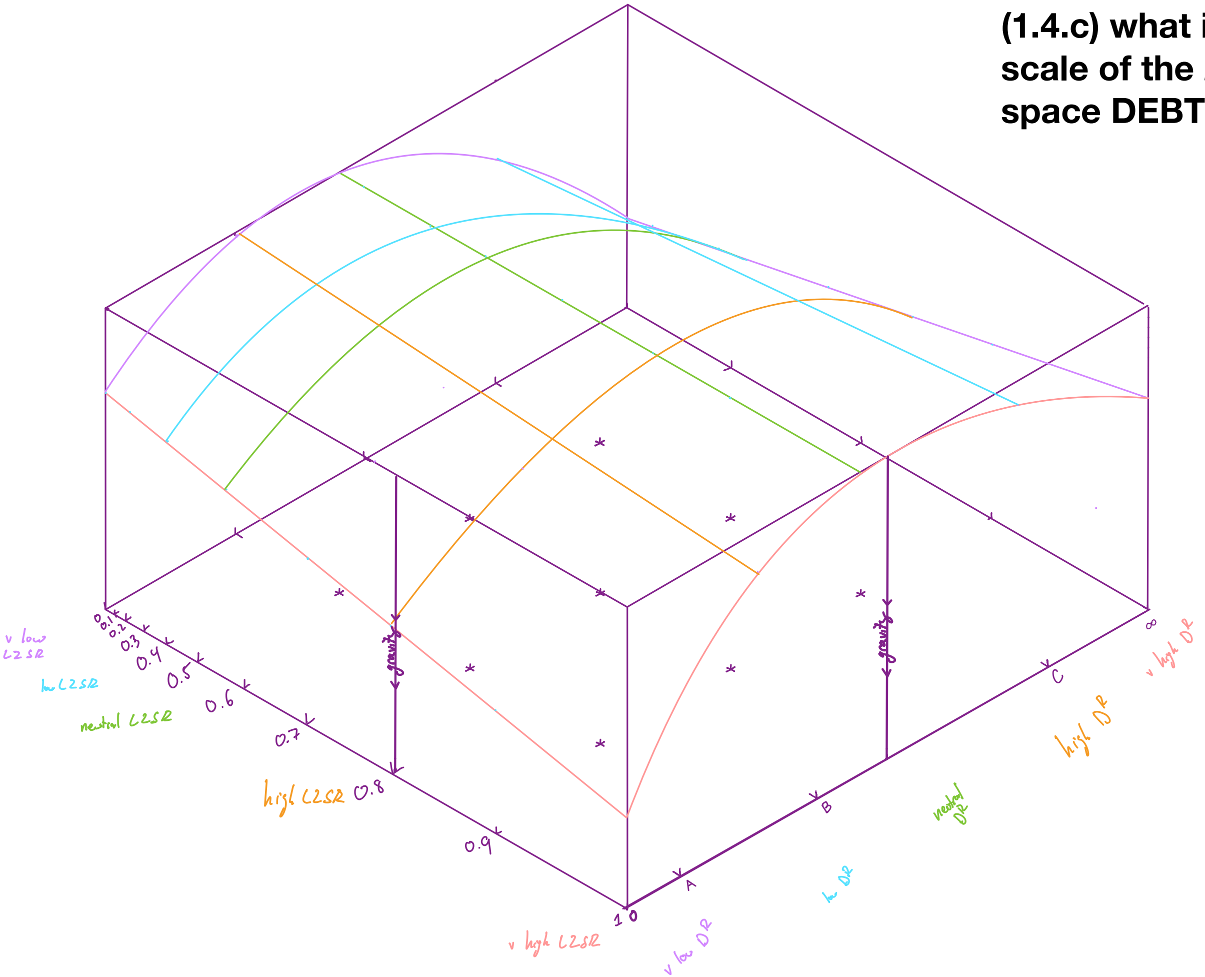
×

Natural Shape, Scale of L2SR at various D^R

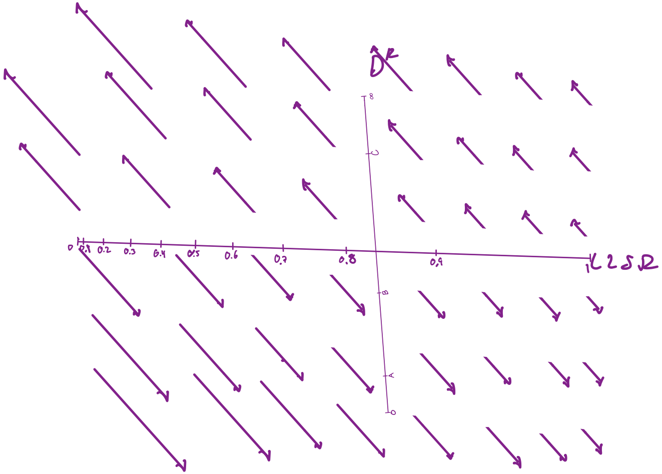


Natural Shape, Scale of D^R x L2SR in 3D

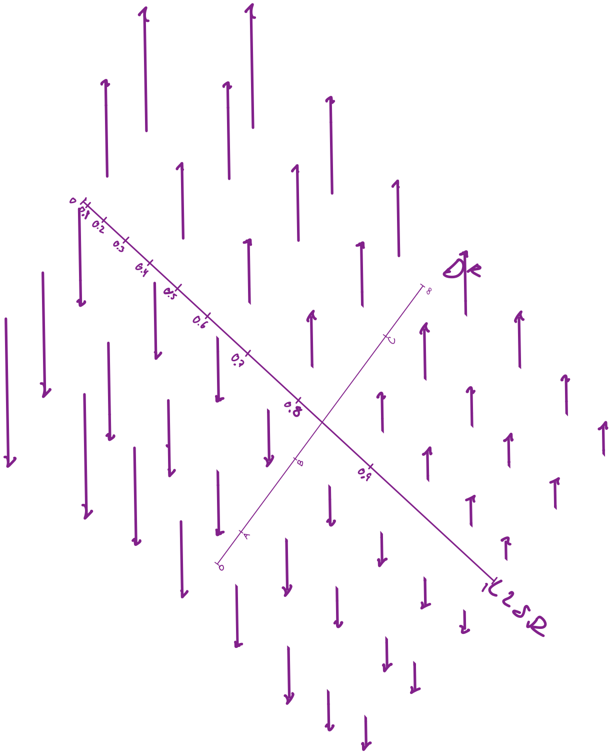
(1.4.c) what is the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR?



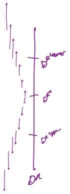
Natural Shape, Scale of P x L2SR in 2D



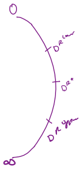
(1.4.c) what is the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR?



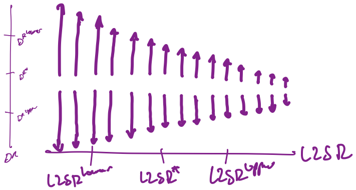
Direction in 2 Dimensions Cheat Sheet: Debt Level x L2SR



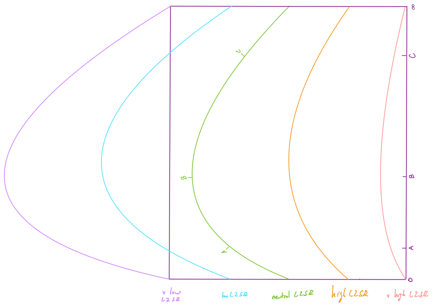
natural effect of D^R on D^R at various D^R



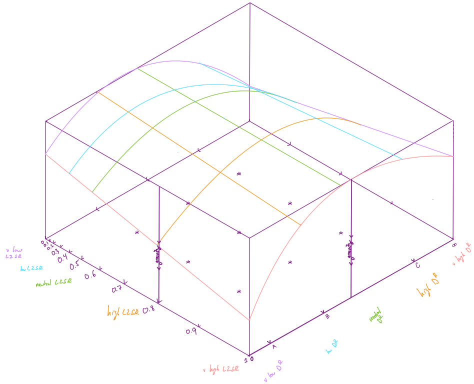
natural shape of D^R



natural effect of L2SR on D^R



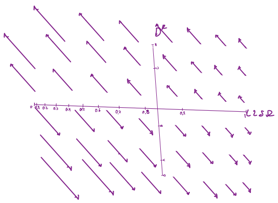
natural shape, scale of D^R at various L2SR



natural shape, scale of D^R x L2SR in 3D

debt level
- moderately reflexive
  - higher debt level requires higher Temperature which increases debt level faster
  - lower debt level requires lower Temperature which increases debt level slower
- turning point is not necessarily around D^R^*

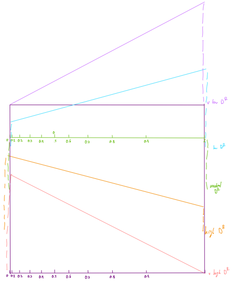
L2SR on D^R
- as L2SR increases, D^R becomes less reflexive
- as L2SR decreases, D^R becomes less reflexive



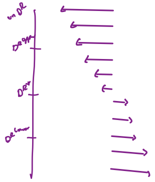
natural shape of D^R x L2SR in 2D

D^R on L2SR
- as debt level increases it becomes more difficult for the L2SR to increase and easier to decrease
- as debt level decreases it becomes easier for the L2SR to increase and more difficult to decrease

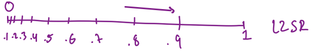
L2SR
- negative correlated with volatility
  - as L2SR increases, L2SR volatility decreases
  - as L2SR decreases, L2SR volatility increases



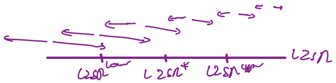
natural shape, scale of L2SR at various D^R



natural effect of D^R on L2SR



natural shape of L2SR (flat and squishes to 0)



natural effect of L2SR on L2SR

Shape and Scale of DEBT LEVEL x L2SR

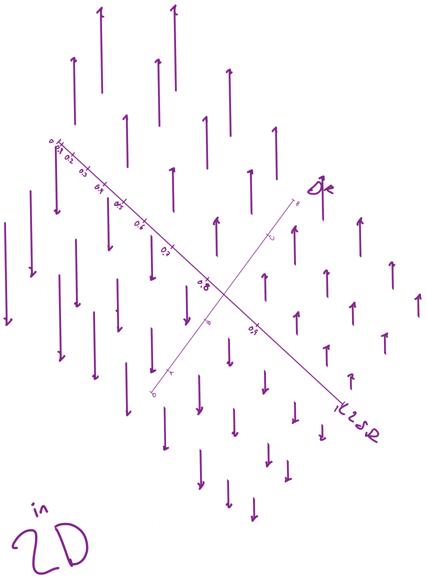
Shape: reflexive wrt D^R, global maximum somewhere along D^R = B, max concavity at (v high D^R, v low L2SR) and (v low D^R, v high L2SR)

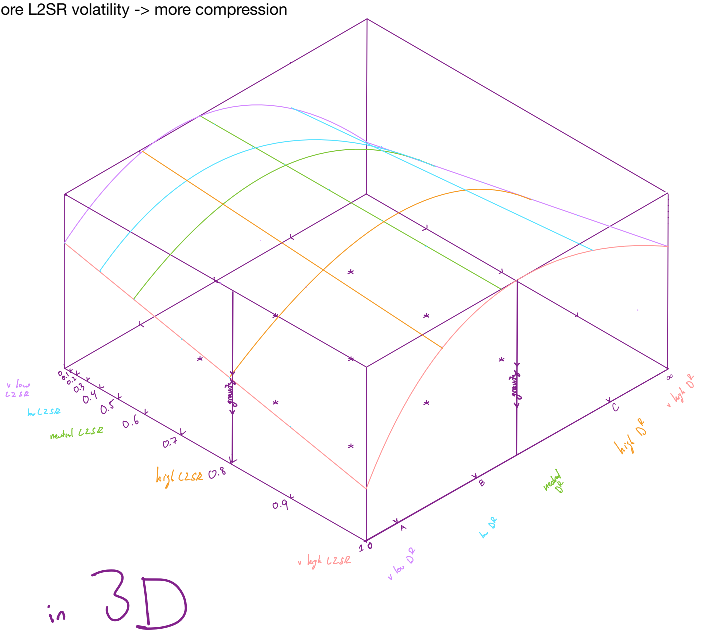
- 1st derivative: negative slope wrt D^R if D^R > B, positive slope if D^R < B. negative slope wrt L2SR if D^R < B, positive slope if D^R > B.
    - debt level:
        - if D^R > B there is upward D^R pressure. L2SR is inversely correlated with upward D^R pressure.
        - if D^R < B there is downward D^R pressure. L2SR is correlated with downward D^R pressure.
    - L2SR:
        - if D^R > B there is downward pressure on L2SR. D^R is correlated with downward L2SR pressure.
        - if D^R < B there is upward pressure on L2SR. D^R is inversely correlated with upward L2SR pressure.
- 2nd derivative: concave throughout. global maximum at (neutral D^R, neutral L2SR) OR (B, W) (*i.e.*, somewhere on the line D^R = B).
    - debt level:
        - if D^R > B, L2SR is inversely correlated with the rate of increase in upward D^R pressure.
        - if D^R < B, L2SR is inversely correlated with the rate of increase in downward D^R pressure.
    - L2SR:
        - if D^R > B, D^R is positively correlated with the rate of increase in downward L2SR pressure.
        - if D^R < B, D^R is inversely correlated with the rate of increase in upward L2SR pressure.
- 3rd derivative: gravitational "holes" at (v high D^R, v low L2SR) and (v low D^R, v high L2SR)

(1.4.d) given the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR, what can Beanstalk infer about its state in terms of derivatives of position (i.e., direction, acceleration, etc.)?

Scale: relationship between the volatility of debt level and liquidity level is highly influenced in practice by the relative demand for Soil vs Converting.

- 1st derivative: expansion if far from B, compression if close to B. as L2SR increases (decreases), volatility decreases (increases).
    - debt level:
        - if D^R far from B there is expansion
        - if D^R is close to B there is compression
    - L2SR:
        - if L2SR increases -> expansion
        - if L2SR decreases -> compression
- 2nd derivative : more compression closer to B, more expansion further from B. the more L2SR increases (decreases), the more expansion (compression)
    - debt level:
        - as D^R moves further away from B there is more expansion
        - as D^R moves closer to B there is more compression
    - L2SR:
        - same increase of L2SR @ higher L2SR -> less L2SR volatility -> more expansion
        - same decrease of L2SR @ lower L2SR -> more L2SR volatility -> more compression





intuitively, the difficulty of balancing a ball on this surface is far from a simple task. nonetheless, this is one of 3 cuts over which Beanstalk has to optimize under the new peg maintenance task (*i.e.*, post Seed Gauge).

in this case it is a sign of health for Beanstalk to fall into the corner with low D^R and high L2SR. However, because it requires a lot of momentum to get out of the hole Beanstalk should be careful that when price finally comes down in its momentum to carry it into a major debt cycle with low prices and decreasing liquidity levels is limited as much as possible.

better understanding the shape of this space can help answer questions like 'when should Beanstalk prefer to issue debt over encourage Converts or vice versa?'. Clearly, avoiding extremely high levels of debt coupled with low levels of liquidity, beyond which there is no return, is a priority over short term peg maintenance.

the relationship between debt level and L2SR becomes clear by looking at this space: the L2SR is more likely to increase when D^R > B and more likely to decrease when D^R < B.

note, the actual scale of the debt level and L2SR along these curves is unknown in practice. this makes Beanstalk's peg maintenance task more complex.

Direction and Acceleration of DEBT LEVEL x L2SR

Direction:

key idea: **L2SR is naturally more likely to be decreasing when D^R > B and increasing when D^R < B. Debt level is important compared to L2SR in its effect on the shape of space wrt both debt level and L2SR.**

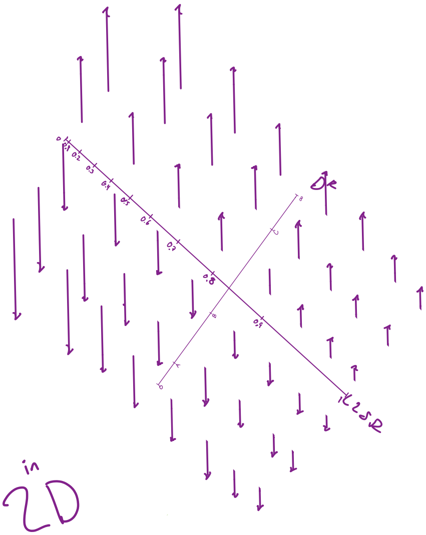
- if D^R > B the directional force of D^R x L2SR is positive wrt D^R
- if D^R > B the directional force of D^R x L2SR is negative wrt L2SR
- if D^R < B the direction force of D^R x L2SR is negative wrt D^R
- if D^R < B the directional force of D^R x L2SR is positive wrt L2SR

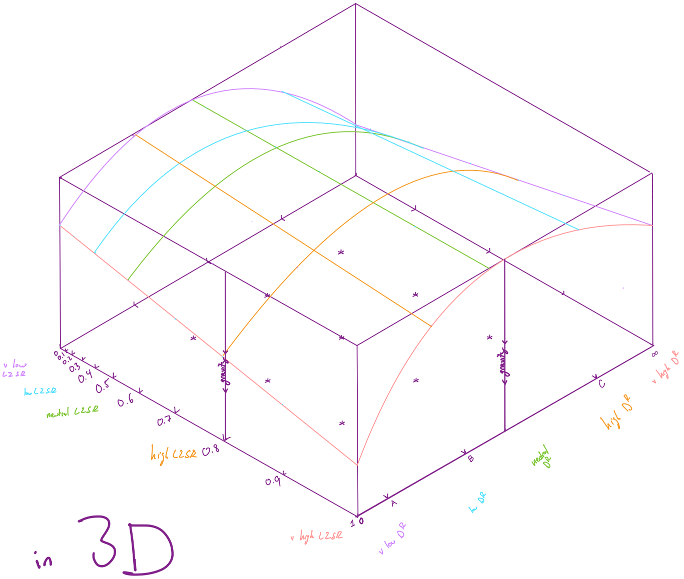
Acceleration:

key idea: **Beanstalk naturally accelerates towards (v high D^R, v low L2SR) if D^R > B and towards (v low D^R, v high L2SR) if D^R < B**

- if D^R > B the directional force of D^R x L2SR on D^R is increasing in magnitude as D^R increases.
- if D^R < B the directional force of D^R x L2SR on D^R is increasing in magnitude as D^R decreases.
- if D^R > B the directional force of D^R x L2SR on L2SR is increasing in magnitude as D^R increases.
- if D^R < B the directional force of D^R x L2SR on L2SR is increasing in magnitude as D^R decreases.

(1.4.d) given the shape and scale of the 2 dimensional space DEBT LEVEL x L2SR, what can Beanstalk infer about its state in terms of derivatives of position (i.e., direction, acceleration, etc.)?





Ideal equilibrium lies somewhere on this curve. In practice, properly setting L2SR^Lower, L2SR*, L2SR^Upper, D^R^Lower, D^R* and D^R^Upper is very difficult.

Now that we have evaluated the interplay between each pair of axes, we can better evaluate the interplay between the 3 axes and the peg maintenance system that will go live upon the implementation of the Seed Gauge system.

Beanstalk in

3D

(1.4) given the shape and scale of each axis, what can Beanstalk infer about the state of Beanstalk in terms of derivatives of position (i.e., direction, acceleration, etc.)?

The following questions can be used to better understand the interplay between PRICE, DEBT LEVEL and LIQUIDITY LEVEL:

(1.4.a) how does PRICE act differently upon itself at various DEBT LEVELS and LIQUIDITY LEVELS?

(1.4.b) how does DEBT LEVEL act differently upon itself at various PRICES and LIQUIDITY LEVELS?

(1.4.c) how does LIQUIDITY LEVEL act differently upon itself at various PRICES and DEBT LEVELS?

(1.4.d) what is the shape and scale of the 3 dimensional space PRICE x DEBT LEVEL x LIQUIDITY LEVEL?

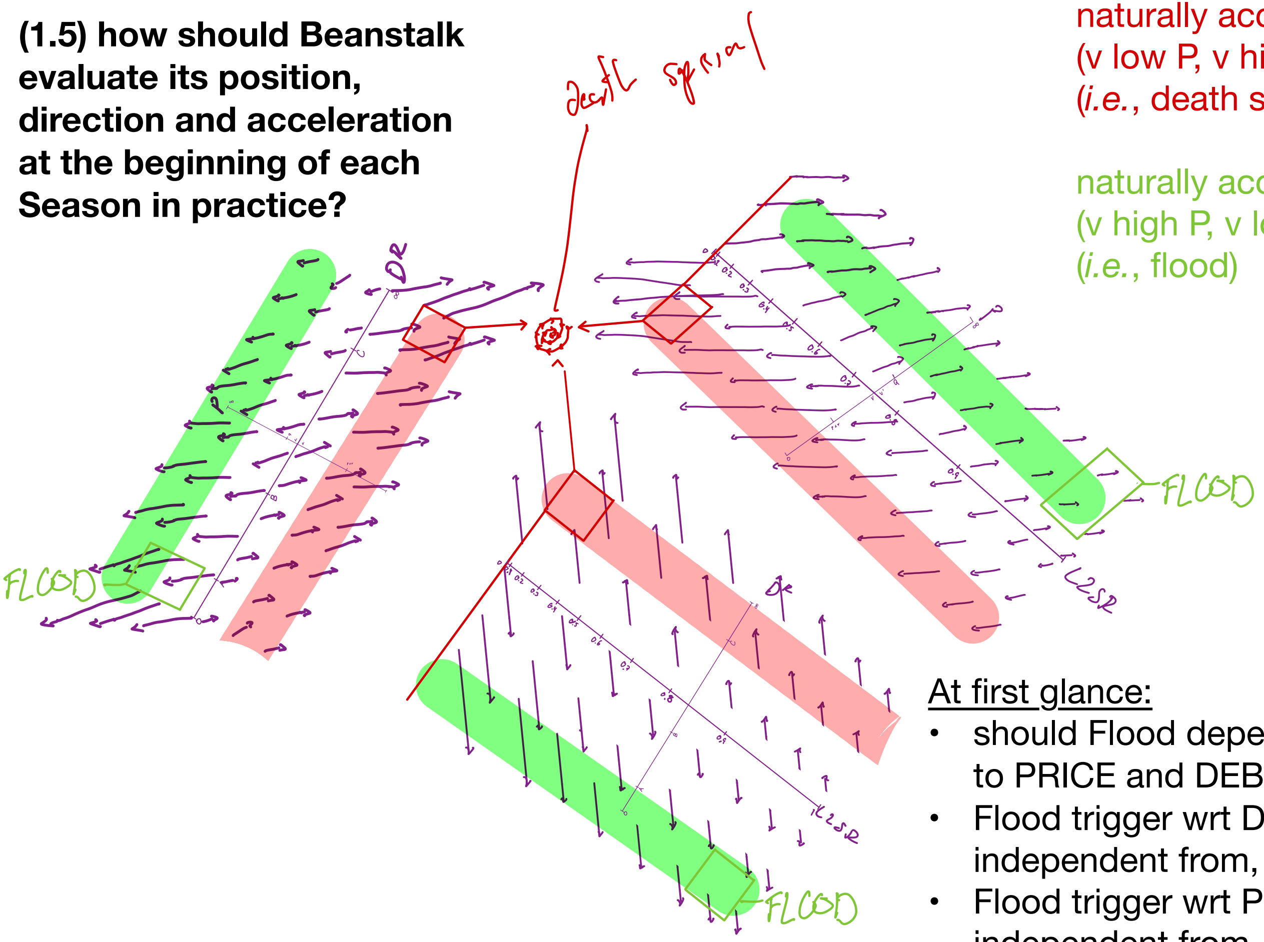
(1.4.e) given the shape and scale of the 3 dimensional space PRICE x DEBT LEVEL x LIQUIDITY LEVEL, what can Beanstalk infer about its state in terms of derivatives of position (*i.e.*, direction, acceleration, etc.)?

these questions are presented as an exercise for anyone to think deeply on the shape and scale of the 3 dimensional space in 4 dimensions.

(1.5) how should Beanstalk evaluate its position, direction and acceleration at the beginning of each Season in practice?

naturally accelerating to (v low P, v high D^R, v low L2SR), (*i.e.*, death spiral) 🟥

naturally accelerating to (v high P, v low D^R, v high L2SR), (*i.e.*, flood) 🟩



At first glance:

- should Flood depending on L2SR in addition to PRICE and DEBT LEVEL? **Probably**
- Flood trigger wrt DEBT LEVEL should be independent from, and lower than, D^R^Lower
- Flood trigger wrt PRICE should be independent from, and higher than, V, and independent from Q. Maybe should be = to Q in practice.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

It may seem like Beanstalk is ALWAYS accelerating towards jubilee or a death spiral.

In some sense, this is TRUE. Algorithmic stablecoins are naturally reflexive, and this is clearly reflected is the shape of space in which Beanstalk operates.

This is a scary picture at first, and speaks to the complexity of the problem Beanstalk is attempting to solve.

However, using some basic reasoning and previous experience with Beanstalk, we can understand the picture a lot more deeply.

Let's build on the work we did for peg maintenance within PRICE x DEBT LEVEL and try to come up with good rules for classifying state within the additional dimension of LIQUIDITY LEVEL which is introduced in the upcoming Seed Gauge system.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

LIQUIDITY LEVEL

Beanstalk uses the L2SR as a proxy for LIQUIDITY LEVEL. Both the supply of Beans and amount of liquidity is calculated at the time of the Sunrise. The Bean supply does not require explicit manipulation resistance because it changes exclusively downstream of manipulation resistant values. The liquidity calculated in V as the product of the time-weighted SMA of the non Bean asset in whitelisted liquidity pools and the time-weighted average of the price of the non Bean asset over the previous Season.

Beanstalk evaluates L2SR discretely as very low, low, high and very high depending on its position with respect to L2SR^Lower, L2SR* and L2SR^Upper.

It is very difficult to properly set L2SR^Lower, L2SR* and L2SR^Upper given the lack of understanding of the actual L2SR scale in practice.

A future modification to the implementation of this calculation could implement a liquidity whitelist to modularly include liquidity from given pools at arbitrary weights from 0 to 1 in the sum.

(1.5) how should Beanstalk evaluate its *position*, direction and acceleration at the beginning of each Season in practice?

PRICE x DEBT LEVEL x LIQUIDITY LEVEL

The position of Beanstalk in the 3 dimensional space of PRICE x DEBT LEVEL x LIQUIDITY LEVEL is the combination of the position in PRICE and position in DEBT LEVEL

This leaves 3 cases with respect to price, 4 cases with respect to debt level and 4 cases with respect to liquidity level for a total of 48 potential positions with respect to price, debt level and liquidity level in the peg maintenance model in place upon the implementation of the Seed Gauge system.

Note in the Seed Gauge implementation a 3rd case is added to PRICE of P > Q.

(1.5) how should Beanstalk evaluate its position, *direction* and acceleration at the beginning of each Season in practice?

Ideal Equilibrium

Direction is much more useful with respect some sort of optimal point within the PRICE x DEBT LEVEL x LIQUIDITY LEVEL space.

Optimal point in space is (V, D^R*, L2SR*). V because it is the primary point of Beanstalk to oscillate P across V along the PRICE axis. D^R* because that is the arbitrarily defined optimal D^R along the DEBT LEVEL axis. L2SR* because that is the arbitrarily defined optimal L2SR along the LIQUIDITY LEVEL axis.

Beanstalk is defined to be in ideal equilibrium in space when (1) P is regularly oscillating across V, (2) the debt level is at D^R*, (3) the liquidity level is at L2SR* and (4) demand for Soil is steady.

In practice maintaining ideal equilibrium is impossible and Beanstalk must always respond to its position direction and acceleration with respect to ideal equilibrium.

(1.5) how should Beanstalk evaluate its position, *direction* and acceleration at the beginning of each Season in practice?

Beanstalk is either moving away from ideal equilibrium (and into one of the two holes in the space) or towards ideal equilibrium (and out of one of the two holes in the space).

This is due to the strong relationship between P and D^R and L2SR. One could argue Beanstalk is always moving towards one of the two holes and they would be correct.

A future peg maintenance model may use relative position, direction and acceleration relative to the two holes instead of relative to ideal equilibrium. The direction is implied by the price and debt level and therefore does not require extra cases.

This leaves 3 cases with respect to price, 4 cases with respect to debt level and 4 cases with respect to liquidity level for a total of 48 potential positions and directions with respect to price, debt level and liquidity level in the peg maintenance model in place upon the implementation of the Seed Gauge system.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

The natural state of Beanstalk is always accelerating towards one of the two holes. however, it is possible for Beanstalk to measure its health in real time in the market via demand for Soil.

When demand for Soil is increasing Beanstalk is experiencing force towards the (v high P, v low D^R, v high L2SR) hole. When demand for Soil is decreasing Beanstalk is experiencing force towards the (v low P, v high D^R, v low L2SR) hole. When demand for Soil is steady Beanstalk is experiencing only the natural force and its momentum.

Demand for Soil is considered increasing, steady or decreasing.

By combining direction with the force derived from the measurement of demand for soil, Beanstalk can get a more accurate read on its second derivative with respect to ideal equilibrium than just always accelerating.

This leaves 3 cases with respect to acceleration, 3 cases with respect to price, 4 cases with respect to debt level and 4 cases with respect to liquidity level for a total of 144 potential positions, directions and accelerations with respect to price, debt level and liquidity level in the peg maintenance model in place upon the implementation of the Seed Gauge system.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

Whereas demand of Soil is used as a proxy for Beanstalk’s ability to apply force to the overall state, there is no proxy for its ability to apply force with respect to liquidity level.

A future peg maintenance model should consider both demand for Soil and some indicator of demand for Conversions.

(1.5) how should Beanstalk evaluate its position, direction and *acceleration* at the beginning of each Season in practice?

Basic intuition for determining state with respect to ideal equilibrium:

- When P > V, D^R is decreasing and L2SR is likely increasing. When P < V, D^R can only increase and L2SR is likely decreasing.
- Direction is towards ideal equilibrium if D^R is moving towards D^R* and L2SR is moving towards L2SR*.
- Direction is away from ideal equilibrium if D^R is moving away from D^R* and L2SR is moving away L2SR*.
- Unclear how to best classify direction when either D^R or L2SR is moving towards its optimal value and the other is moving away. The Seed Gauge peg maintenance model does not specifically handle this.
- Demand for Soil is used for a proxy for health of the Soil market. This represents Beanstalk's ability to slow down or speed up the natural acceleration to (v low P, v high D^R, v low L2SR) or (v high P, v high D^R, v low L2SR) via issuance of debt.
- Acceleration is determined by whether the change in demand for Soil from Season to Season is:
  - directionally aligned with the natural acceleration wrt D^R —> accelerating
  - constant —> steady
  - directionally aligned against the natural acceleration wrt D^R —> decelerating

we have answered all the questions necessary to properly be able to classify state for the post Seed Gauge implementation of Beanstalk’s peg maintenance model, which operates in all 3 dimensions.

(2) how should Beanstalk respond to its state at the beginning of each Season in an attempt to return to its ideal state?

Responding to State

Some key questions to answer in determining what the proper response to the Beanstalk state is are:

(2.1) What is the natural flow of state given the current position?

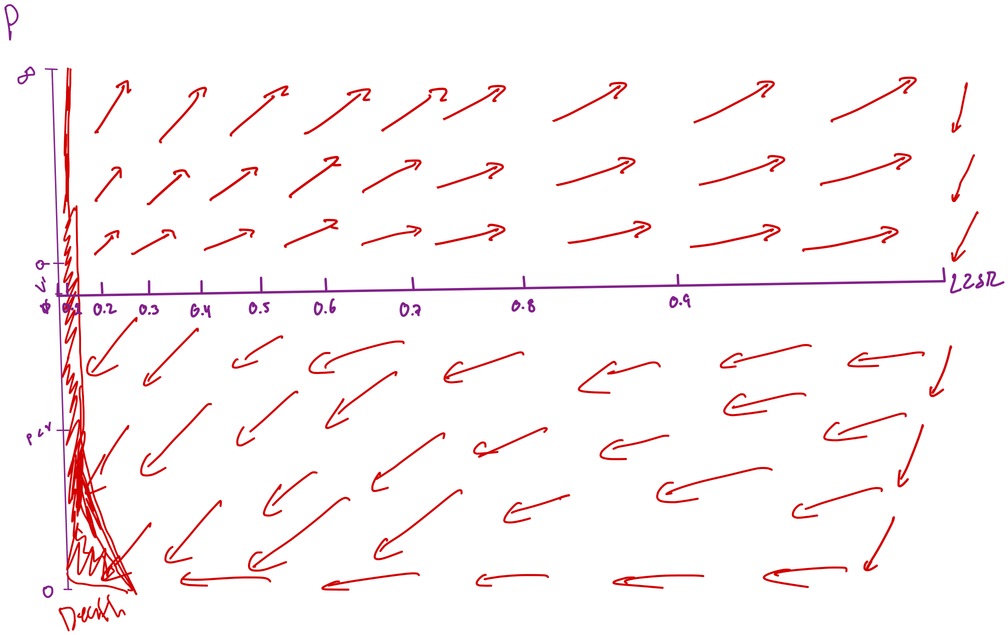
(2.2) What tools does Beanstalk have available to perform peg maintenance?

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

(2.4) How should Beanstalk use these tools in practice?

(2.1) What is the natural flow of state given the current position?

<u>PRICE x L2SR Natural Flow (*i.e.*, part 2 of the Problem)</u>

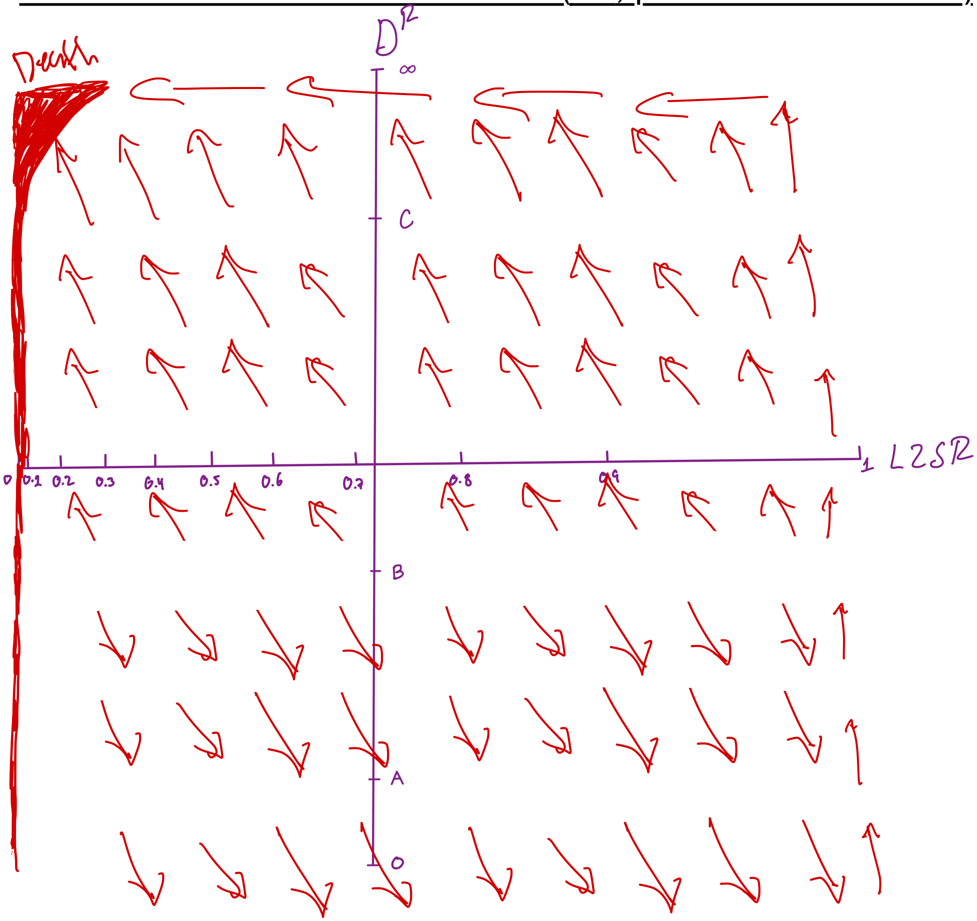


Natural flow: given shape and scale of PRICE x LIQUIDITY LEVEL what is the the tendency of the system?

Any debt based stablecoin system with sufficient incentives to provide liquidity over holding the stablecoin outright is going to tend to pump (*i.e.*, high P, high L2SR). The higher the pump, the greater the dump (*i.e.*, low P, low L2SR).

The natural tendency to dump when P < V and buy back when P > V is alleviated by the opportunity cost from the grown Stalk from Seeds. by changing the amount of incentive for holding Beans vs LP tokens Beanstalk can incentivize Converts to address the PRICE x LIQUIDITY LEVEL space.

DEBT LEVEL x L2SR Natural Flow (*i.e.*, part 3 of the Problem)



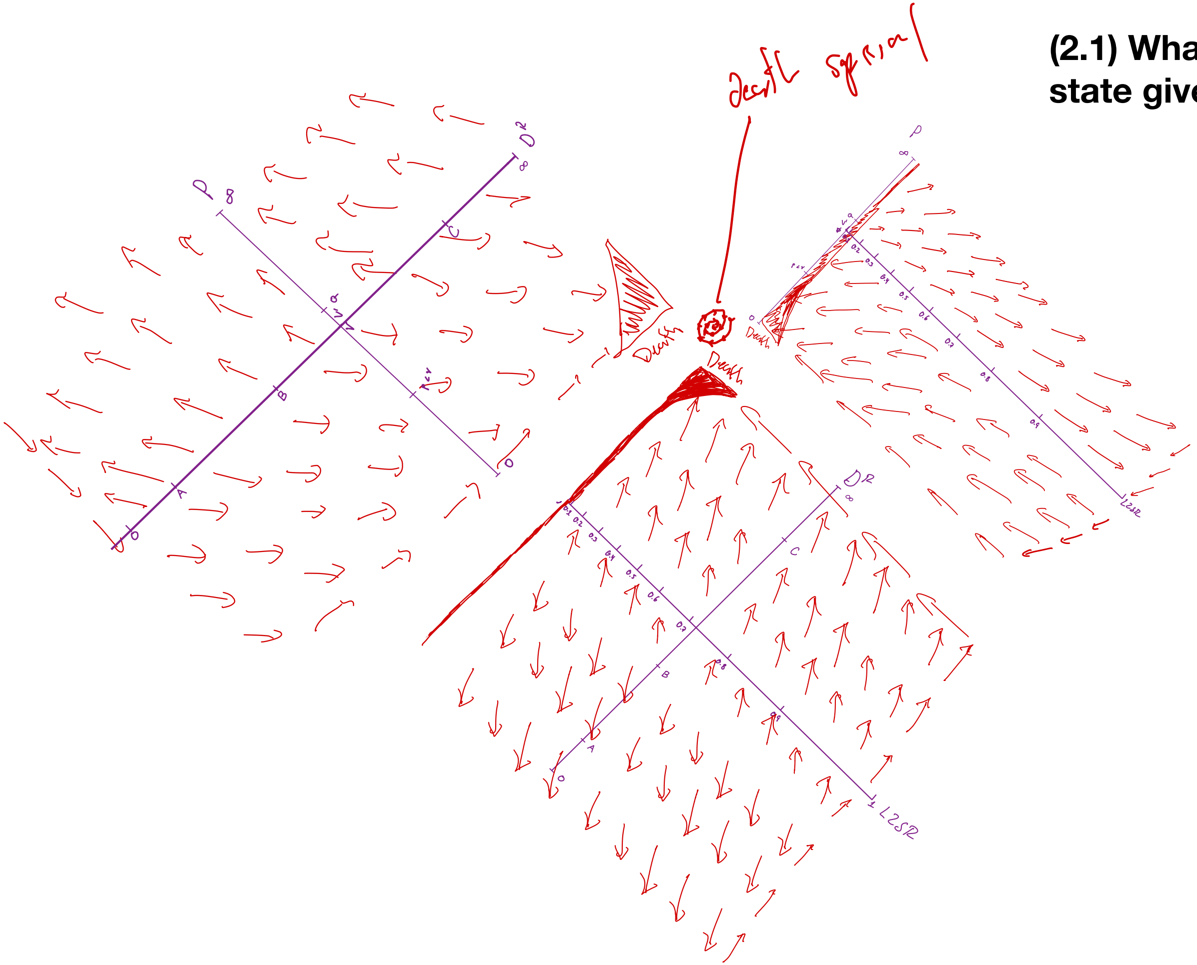
(2.1) What is the natural flow of state given the current position?

Natural flow: given shape and scale of DEBT LEVEL x LIQUIDITY LEVEL what is the the tendency of the system?

Any debt based stablecoin system with sufficient incentives to provide liquidity over holding the stablecoin outright is going to tend to pump (*i.e.*, low D^R, high L2SR). The higher the pump, the greater the dump (*i.e.*, high D^R, low L2SR).

The opportunity to raise and lower the Price of Beans through EITHER a change in the debt level or a change in the liquidity level creates a fascinating tradeoff space in which Beanstalk can operate.

Natural Flow in all 3 pairs of dimensions in 2D



(2.1) What is the natural flow of state given the current position?

(2.2) What tools does Beanstalk have available to perform peg maintenance?

Seed Gauge

(2.2) What tools does Beanstalk have available to perform peg maintenance?

<u>Components of the Guage</u>

- Average Grown Stalk per BDV (agspbdv) — determines the rate at which new Depositors "catch up" to older Depositors <u>over time</u>
- LP vs Bean Seed distribution — determines relative benefits of holding LP vs Bean exposure in the Silo <u>over time</u>
- LP Seed distribution — determines relative benefits of holding a given non-Bean asset in the Silo <u>over time</u>

whereas the Generalized Convert penalty/bonus system imposes a cost/benefit to Converting <u>in a given instant</u>, the gauge system offers marginal incentives that are realized <u>over time</u>. therefore, the two peg maintenance components (i.e., Seed Gauge system and Generalized Convert) ultimately combine to create a dimension to the peg maintenance mechanism that is autonomous and almost entirely independent of and complementary to the Field.

<u>intuition:</u> seeds generate opportunity cost for Withdrawing assets that have been Deposited for longer AND marginal benefit for holding particular assets in the Silo in the form of grown Stalk. marginal opportunity cost and benefit is realized over time. the seed gauge system is design to allow Beanstalk to autonomously toggle all 3 axes with respect to Seeds in arbitrary and modular fashions.

3 new tools: (1) value of time in the Silo
(2) value of Bean vs LP exposure
(3) value of various LP exposures

Tools 1 and 3 are relevant to the Seed Gauge system but not relevant to the overall state space or the general peg maintenance mechanism. Instead, they are refinements. Therefore, they will only be treated lightly.

(2.2) What tools does Beanstalk have available to perform peg maintenance?

New Tool 1: Value of Time

the average seeds per BDV in the Silo is the average rate at which grown Stalk accrues to Silo Deposits

by adjusting the average seeds per bdv (*i.e.*, agspbdv) Beanstalk can affect the overall effect of time on the marginal opportunity cost/benefit of new and existing Deposits.

from a design perspective, Beanstalk will support an arbitrary function that returns the optimal agspbdv for a given Season.

determining agspbdv:
- there is some distribution of gspbdv on existing deposits.
- agspbdv is a function of the distribution:
    - catch up to average (basic, current implementation)
    - catch up to x percentile (complex, future work)

because the agspbdv does not effect the state of Beanstalk in a fashion that is realized at any given point in time (only over time) it is not considered in the overall peg maintenance mechanism and instead only needs to be discussed in its own context.

(2.2) What tools does Beanstalk have available to perform peg maintenance?

New Tool 2: Bean vs LP Seed Distribution

main issue: b/c there are multiple types of LP, how does Beanstalk toggle the overall relationship between Bean and LP. **should Beanstalk use avg LP gspbdv or max LP gspbdv as its reference point?**

because there is a natural limit where the Seeds for Beans should never exceed the Seeds for 1 BDV of all LP, beyond which Beanstalk no longer prioritizes LP Deposits over Beans (likely to cause inorganic demand, overheating and excess volatility), it makes sense to use the max Seeds per 1 BDV of LP as the reference point.

once the max LP to Bean gspbdv ratio is set, the distribution of grown Stalk amongst LP can be handled by the LP gauge system. Beanstalk requires a minimum and maximum ratio to be set between Bean and the max LP grown Stalk per BDV.

Then, at the beginning of each Season, Beanstalk can toggle the distribution of Bean vs LP from minimum to maximum by toggling the gauge points, where 0 is the maximum allocation to LP and 1 is the maximum allocation to Bean.

(2.2) What tools does Beanstalk have available to perform peg maintenance?

New Tool 3: LP Gauge System

- Each LP receives some portion of the grown Stalk allocated relative to max LP based on the amount of LP gauge points it has.
- LP gauge points are set by functions that evaluate the current LP distribution relative to optimal distribution and adjusts the gauge points for a given LP accordingly.

This presents 2 key questions around the LP gauge system:

1) how to determine the optimal BDV distribution amongst LP tokens

2) how to determine the adjustment of LP gauge points to various LP token Deposits

the former is not answered in an autonomous fashion by the current Seed Gauge BIP but can be statically set and adjusted via BIP.

the latter is set by the above described functions.

because the distribution of liquidity does not effect the state of Beanstalk at any given point in time (only over time) it is not considered in the overall peg maintenance mechanism and instead only needs to be discussed in the context of the LP gauge system.

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

Changing the Seed distribution between LP and Bean is the only new peg maintenance tool that affects the overall state of Beanstalk. the others are secondary tools that do not affect Beanstalk's overall state.

As a design principle, any time there is a Conversion from Bean to LP, Beanstalk reallocates liquidity along an AMM downward such the attempts to decrease PRICE through Conversions are always coupled with an increase in LIQUIDITY LEVEL.

As a design principle, any time there is a Conversion from LP to Bean, Beanstalk reallocates liquidity along an AMM downward such the attempts to decrease PRICE through Conversions are always coupled with an decrease in LIQUIDITY LEVEL.

(2.3) How do these tools affect Beanstalk’s position along various axes in theory?

Beanstalk can always attempt to increase price for decreased L2SR through the incentivization of Converts from LP to Bean.

Beanstalk can always attempt to decrease price for increased L2SR through the incentivization of Converts from Bean to LP.

Increasing the distribution of Seeds to Beans over LP means Beanstalk is willing to incentivize the Conversion from LP to Bean more.

Increasing the distribution of Seeds to LP over Beans means Beanstalk is willing to incentivize the Conversion from Bean to LP more.

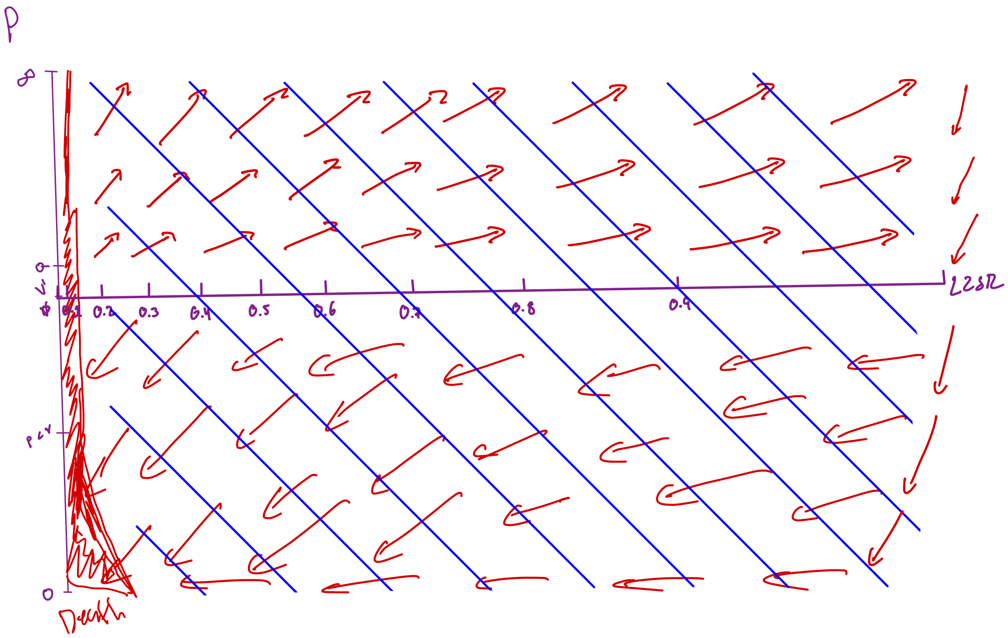
The exact rate of tradeoffs between PRICE AND DEBT LEVEL and LIQUIDITY LEVEL AND DEBT LEVEL are unclear in practice.

The question then becomes under what circumstances should Beanstalk encourage the tradeoff and in which direction?

(2.3) How do these tools affect Beanstalk's position along various axes in theory?

PRICE x L2SR Position Tradeoff

▨ estimated directional tradeoff between PRICE and LIQUIDITY LEVEL.



Changing the distribution of Seeds per BDV to more heavily favor Beans vs LP is applying force down and to the right along the purple lines.

Changing the distribution of Seeds per BDV to more heavily favor LP vs Beans is applying force up and to the left along the purple lines.

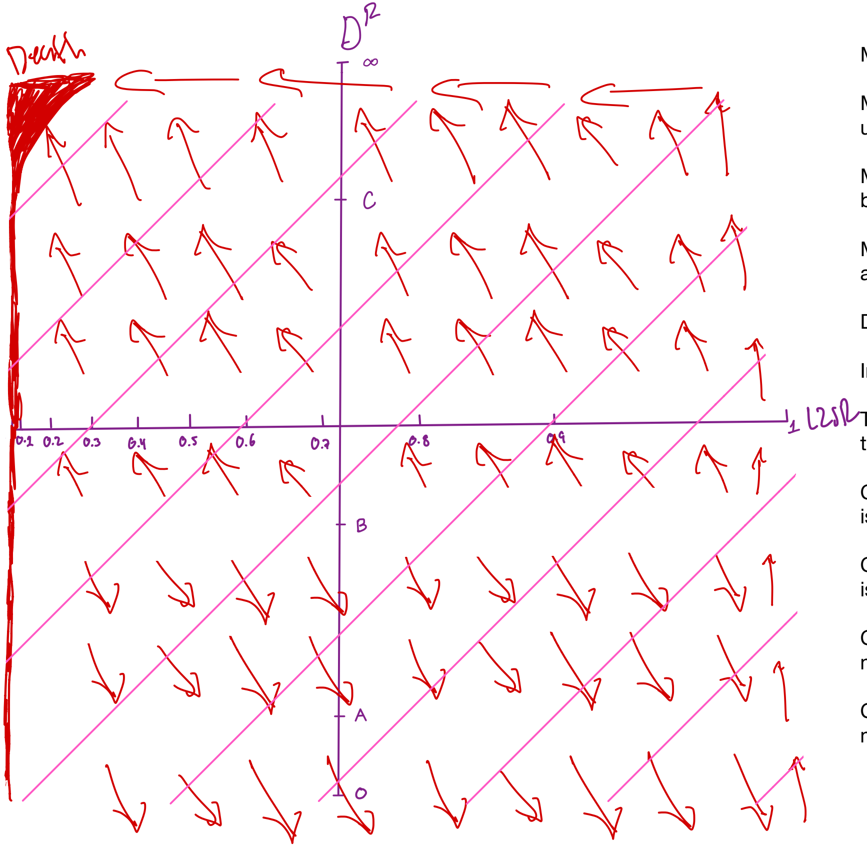
Changing the distribution of Seeds per BDV to more heavily favor Beans vs LP more is applying more force down and to the right along the purple lines.

Changing the distribution of Seeds per BDV to more heavily favor LP vs Beans more is applying more force up and to the left along the purple lines.

(2 C.3) How do these tools affect Beanstalk's position along various axes in theory?

DEBT LEVEL x L2SR Position Tradeoff

▨ estimated directional tradeoff between DEBT LEVEL and LIQUIDITY LEVEL.



Minting Beans is applying force down and to the left along the pink lines.

Minting Soil in an environment where there is demand for it is applying force up and to the right along the pink lines.

Minting more (less) Beans increases (decreases) the downward force along both axes.

Minting more (less) Soil increases (decreases the upward force along both axes.

Decreasing the Maximum Temperature decreases the slope of the pink lines.

Increasing the Maximum Temperature increases the slope of the pink lines.

The larger the amount of increase (decrease) in the Maximum Temperature, the larger the increase (decrease) in slope of the pink lines.

Changing the distribution of Seeds per BDV to more heavily favor Beans vs LP is applying force down and to the left along the pink lines.

Changing the distribution of Seeds per BDV to more heavily favor LP vs Beans is applying force up and to the right along the pink lines.

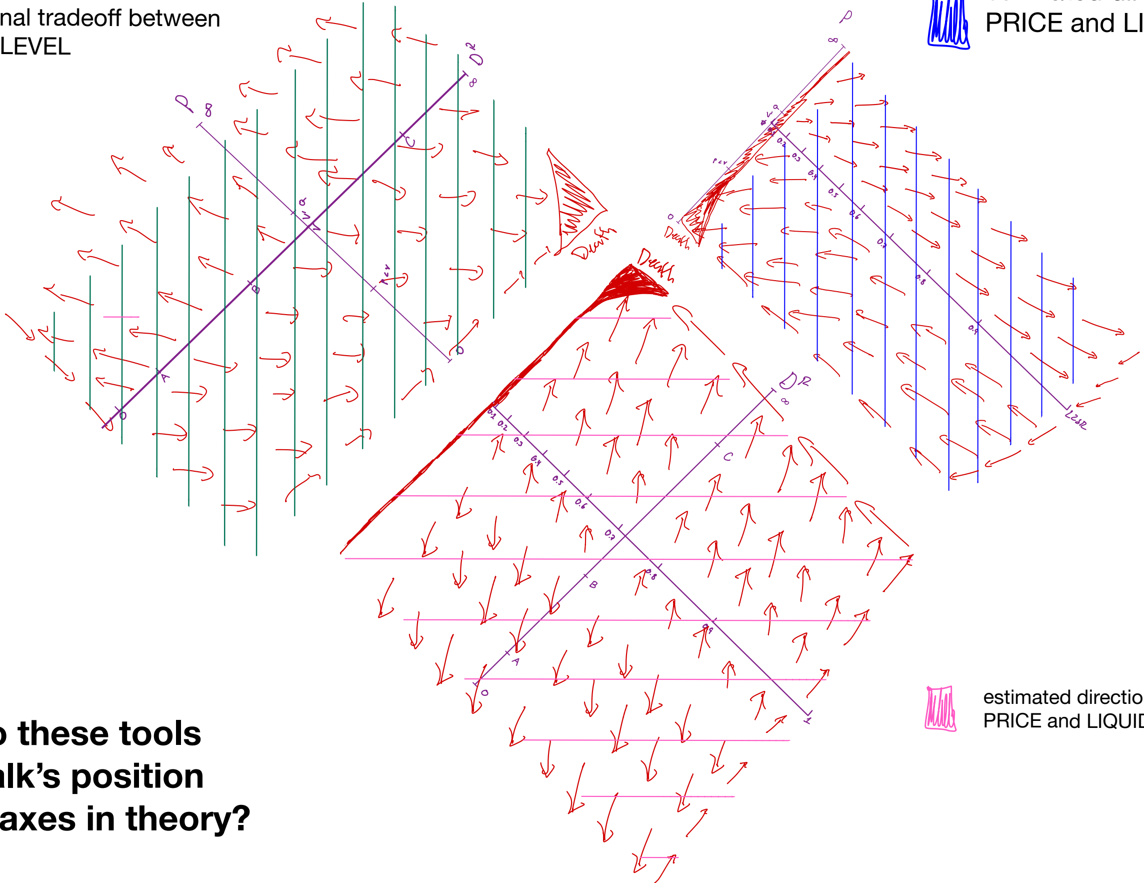
Changing the distribution of Seeds per BDV to more heavily favor Beans vs LP more is applying more force down and to the left along the pink lines.

Changing the distribution of Seeds per BDV to more heavily favor LP vs Beans more is applying more force up and to the right along the pink lines.

Estimated Direction Tradeoff Over Natural Flow for all 3 pairs of dimensions in 2D

🟩 estimated directional tradeoff between PRICE and DEBT LEVEL

🟦 estimated directional tradeoff between PRICE and LIQUIDITY LEVEL.



🟪 estimated directional tradeoff between PRICE and LIQUIDITY LEVEL.

(2 C.3) How do these tools affect Beanstalk's position along various axes in theory?

(2 C.3) How do these tools affect Beanstalk's position along various axes in theory?

Seed Gauge Distribution Points System

Whereas the potential for the Temperature to increase quickly creates an inefficiency in the market for Soil, the potential for the distribution of Seeds between Beans and LP to change quickly does not create an inefficiency in the market for Deposits.

Therefore, when it comes to reflecting preference between issuing debt and changing liquidity levels, changing the distribution points of the Seed Gauge quickly is a much better way to quickly do so.

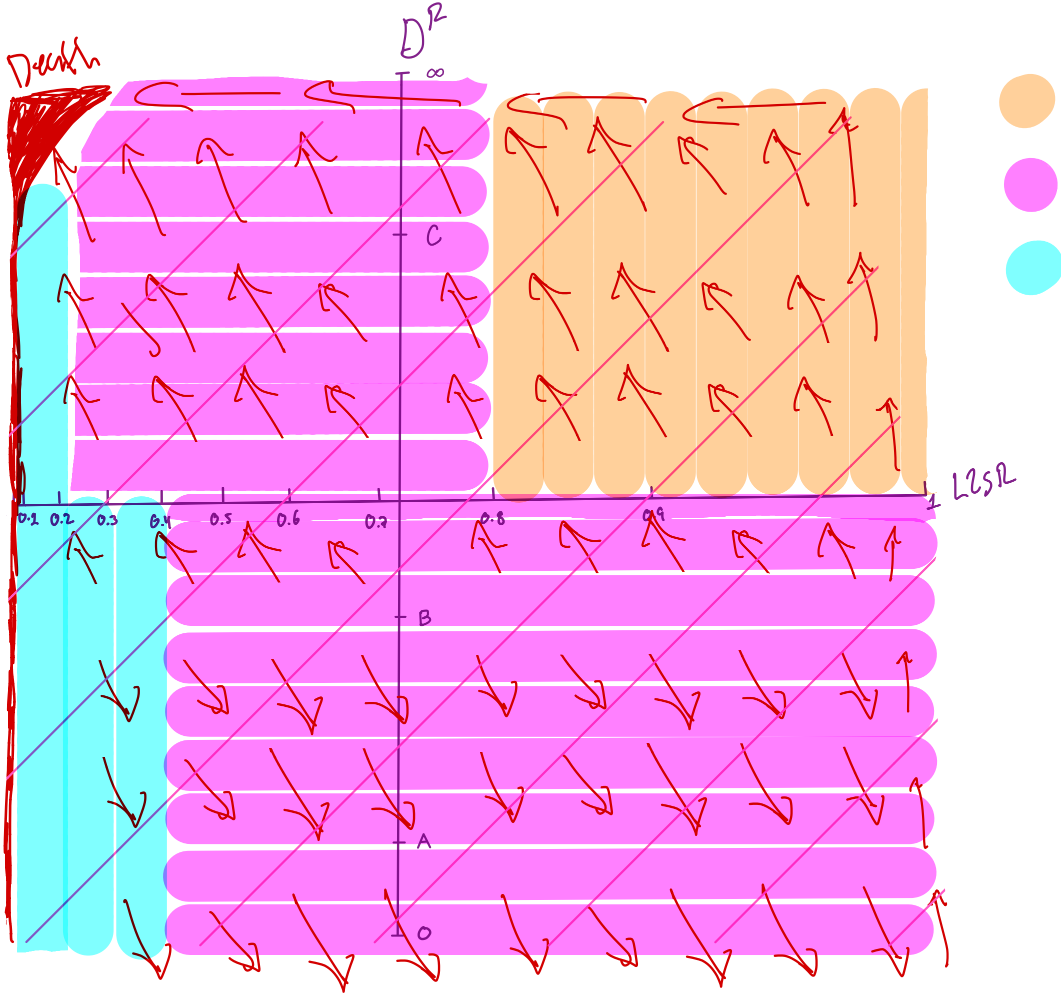
While it was considered to add a multiplicative component to the cases for the Seed Gauge in addition to the additive one present in the Maximum Temperature changing cases, it was ultimately determined that similar aggressive changes to Seed Distribution can be performed using only the additive component. therefore while the implementation now supports multiplicative changes for both Maximum Temperature and Seed Gauge neither is uses it.

(2.4) How should Beanstalk use these tools in practice?

The exact rate of tradeoff between DEBT LEVEL AND L2SR is unclear in practice.

However, intuitively there are certain positions in space where Beanstalk would prefer more debt issuance and more liquidity or less debt issuance and less liquidity.

Beanstalk's preferred tradeoff of DEBT LEVEL x L2SR



(2.4) How should Beanstalk use these tools in practice?

🟠 prefers less debt and liquidity

🟣 lightly prefers more debt and liquidity

🔵 strongly prefers more debt and liquidity

Beanstalk has demonstrated more existential risk from low levels of liquidity than high levels of debt.

therefore, it makes sense for Beanstalk to prefer more debt and more liquidity in all cases except when Beanstalk has extremely high debt and liquidity levels, in which case it would prefer less debt and less liquidity.

the exact ranges where Beanstalk's preference changes directionally is unclear, as is the magnitude in which it changes.

(2.4) How should Beanstalk use these tools in practice?

Natural Question:

Q: Should Beanstalk be more or less aggressive at raising the Temperature and changing the distribution of Seeds between Beans and LP when it is close to a potential death spiral.

A: I would argue at high debt levels Beanstalk shouldn't be more aggressive at raising the Maximum Temperature because it is better to wait out the storm than issue too much debt at too high of an interest rate. Slowly raising the interest rate is a more long term oriented solution. In fact, it may even be beneficial to be more aggressive with Maximum Temperature changes with lower debt levels.

However, Beanstalk should be aggressive at reducing the incentive to Convert from LP to Bean to near 0 quickly when the system is in danger of entering a death spiral due to low liquidity.

(2.4) How should Beanstalk use these tools in practice?

with all of this in mind, let's take a look at the proposed Seed Gauge system case inputs.

because of the ability to quickly reflect changes in Beanstalk's preference with respect to debt and liquidity via the Seed Gauge system there is no change in proposed changes in Maximum Temperature.

the proposed ranges for L2SR are:

- v low L2SR: 0% - 12%
- low L2SR: 12% - 40%
- high L2SR: 40% - 80%
- v high L2SR: 80%+

this would mean the L2SR parameters are:

- L2SR^Lower = 12%
- L2SR* = 40%
- L2SR^Upper = 80%

(2.4) How should Beanstalk use these tools in practice?

<u>New Primary Peg Maintenance Response</u>

1. Mint deltaB Beans if time-weighted average deltaB is positive.
2. Mint negative deltaB Soil if time-weighted average deltaB is negative and enough Soil to issue the most amount of Pods without increasing the outstanding Pods over the course of the Season if the time-weighted average deltaB is positive.
3. Change Maximum Temperature based on position, direction and acceleration wrt ideal equilibrium.
4. Change Seed distribution between Beans and LP Tokens based on position, direction and acceleration wrt ideal equilibrium.

<u>New Secondary Peg Maintenance Response</u>

1. Change agspbdv.
2. Change distribution of Seeds amongst LP Tokens.

in blue zone 2 Seasons to max LP

(2.4) How should Beanstalk use these tools in practice?

debt level

v low L2SR

P & for (demand) changes

		v low	low	high	v high
P > 1	increasing	-.50	-.50	-.50	-.50
	steady	-.50	-.50	-.50	-.50
	decreasing	-.50	-.50	-.50	-.50
P < 1	increasing	-.50	-.50	-.50	-.50
	steady	-.50	-.50	-.50	-.50
	decreasing	-.50	-.50	-.50	-.50
P > Q		-.50	-.50	-.50	-.50

Beanstalk Seed Gauge Distribution Point Changes

in blue zone 2 Seasons to max LP
in green zone system system most
at risk of overheating…2 seasons to max LP
in purple zone small changes based on P

(2.4) How should Beanstalk use these tools in practice?

			debt level			
	low L2SR		v low	low	high	v high
P & soil (demand) changes	P > 1	increasing	-.50	-.50	-.01	-.01
		steady	-.50	-.50	-.01	-.01
		decreasing	-.50	-.50	-.01	-.01
	P < 1	increasing	-.50	-.50	+.01	+.01
		steady	-.50	-.50	+.01	+.01
		decreasing	-.50	-.50	+.01	+.01
	P > Q		-.50	-.50	-.50	-.50

Beanstalk Seed Gauge Distribution Point Changes

in green zone system system most
at risk of overheating…2 seasons to max LP
in purple zone small changes based on P

(2.4) How should Beanstalk use these tools in practice?

Debt level

high	L2SR	v low	low	high	v high
P > 1	increasing	-.01	-.01	-.01	-.01
	steady	-.01	-.01	-.01	-.01
	decreasing	-.01	-.01	-.01	-.01
P < 1	increasing	+.01	+.01	+.01	+.01
	steady	+.01	+.01	+.01	+.01
	decreasing	+.01	+.01	+.01	+.01
P > Q		-.50	-.50	-.50	-.50

P & for 1 [illegible]

Beanstalk Seed Gauge Distribution Point Changes

in green zone system system most
at risk of overheating…2 seasons to max LP
in purple zone small changes based on P
in orange increases faster than it decreases

(2.4) How should Beanstalk use these tools in practice?

P & Δ for T (derived) Δ keys

		debt level			
v high L2SR		v low	low	high	v high
P > 1	increasing	-.01	-.01	-.01	-.01
	steady	-.01	-.01	-.01	-.01
	decreasing	-.01	-.01	-.01	-.01
P < 1	increasing	+.01	+.01	+.02	+.02
	steady	+.01	+.01	+.02	+.02
	decreasing	+.01	+.01	+.02	+.02
P > Q		-.50	-.50	-.50	-.50

Beanstalk Seed Gauge Distribution Point Changes

Future Work within the future PRICE x DEBT LEVEL x LIQUIDITY LEVEL Model

- Research into proper way to estimate A, B and C;
- Research into optimal way to set D^R^Lower, D^R*, D^R^Upper wrt A, B and C;
- Research into proper way to estimate scale of L2SR;
- Research into optimal way to set L2SR^Lower, L2SR*, L2SR^Upper wrt the scale of L2SR;
- Research the interplay between A, B and C and the scale of L2SR;
- Research into how to classify direction in 3D properly.
- Research into measuring the acceleration of price as some function of deltaB over time;
- Research into measuring the acceleration with respect to L2SR;
- Research into a future peg maintenance model that should consider both demand for Soil and some indicator of demand for Conversions; ? maybe Deposits
- Research into measuring demand for Soil in increasingly manipulation resistant fashion via some threshold; demand for Soil in increasingly manipulation resistant fashion;
- Research into changing the amount of time used to measure a given input to position to be over more or less time than is currently used;
- Research into using other sources for measuring position;
- Research into whether a time-weighted SMA or instantaneous EMA is better to use for the L2SR calculation?
- Research into how to properly set an optimal LP distribution where it isn't going to cause losses over time due to improper incentive alignment.
- Research into how to properly set an optimal liability distribution where it isn't going to cause excess growth in liabilities over time.
- Develop an implementation for agspbdv catch up to x percentile (complex, future work)
- A future modification to the implementation of this calculation could implement a liquidity whitelist to modularly include liquidity from given pools at arbitrary weights from 0 to 1 in the sum.
- Research into the effect on peg maintenance of allowing conversions to "lock in" a minimum gspbdv for a period of time after the conversion to make doing so more attractive.

More Discussion

- Setting the average grown stalk per BDV properly
- LP Gauge
- Other tweaks suggested by this study

  - Soil issuance change.
  - Generalized Flood, including L2SR in trigger and adding independent trigger wrt D^R and Price.
- Generalized Convert
- Other related RFCs

  - Decrease BDV
  - Minting Delay Change

Setting the average

Grow stuff fire

TDV (ags pow)

Setting the Average Grown Stalk per BDV (agspbdv)

The Silo uses the Stalk and Seed system for a variety of purposes related to peg maintenance and governance.

Stalk distribution is used to determine the distribution of Bean mints paid to the Silo and voting weight in governance.

Changes in seed distribution between LP and Bean incentivize Converts within the Silo, which affects position.

Changes in seed distribution between assets on the LP Gauge whitelist incentivize Converts within the Silo.

However, most importantly, and the impetus for the existence of the Stalk and Seed system, is to minimize the attractiveness of (1) selling when $P < V$ and buying back again when $P > V$ and (2) inorganic demand when $P > V$.

Stalk and Seed Intuition

Any uncollateralized stablecoin is much more attractive to hold when it is growing (*i.e.*, when P > V) and much less attractive to hold when it is shrinking (*i.e.*, when P < V). This dynamic creates a reflexive feedback loop that leads to pump and dumps. This reflexivity should be well understood after examination of the Beanstalk State Space.

Seeds mint more Stalk over time to Depositors that must be forfeited upon Withdrawal. In doing so, the Silo creates:

- a preference in distribution of Bean mints and governance power to older Depositors.
- opportunity cost in the form of grown Stalk for leaving and coming back.

The preference towards older Depositors and the opportunity cost for leaving and coming back are correlated. The more preference to older Depositors the greater the opportunity cost for leaving and coming back. The less preference to older Depositors the lesser the opportunity cost for leaving and coming back.

The rate at which grown Stalk grows from Seeds compared to the existing grown Stalk is what determines how Beanstalk is handling this tradeoff. The rate at which grown Stalk grows from Seeds is set by the average grown Stalk per BDV (agspbdv).

agspbdv Intuition

one way to understand the relative effect of a given rate at which grown Stalk grows from Seeds is “how many Seasons will it take for a new Depositor Depositing 100BDV to catch up to the:

a) average grown Stalk per BDV at time of Deposit; or
b) X percentile grown Stalk per BDV at time of Deposit.”

Let’s call this value K. K can be thought of as the value of time within the Silo.

As K increases, the value of time within the Silo increases. Preference increases for older Depositors over newer Depositors. The opportunity cost for leaving and coming back increases.

As K decreases, the value of time within the Silo decreases. Preference decreases for older Depositors over newer Depositors. The opportunity cost for leaving and coming back decreases.

Decreasing K is a dangerous game because decreasing the opportunity cost for leaving and coming back can increase the reflexivity of Beanstalk with respect to price and send Beanstalk into a death spiral.

However, having K set too highly can discourage new participation in Beanstalk.

Setting K in Practice

Part of the Seed Gauge BIP includes the ability to set K according to an arbitrary function. The function implemented allows K to be set as a function of time to catch up to the average grown Stalk per BDV at time of Deposit. K is being set with an initial parameter of 6 months.

Setting K so aggressively compared to where it currently is in practice is a response to the massive advantage older Depositors have over ones that has resulted from the extended period without growth since Replant. Given the lack of new participation in Beanstalk it is likely that K is set too high currently. Given that most liquidity is still locked up now is the best time to aggressively set K because it will likely be more dangerous to increase K in the future when the system is more liquid.

The ability to set K as a function of time allows the linear component of Stalk growth from Seeds to be preserved while being able to toggle the slope of growth in an arbitrary fashion.

Future Work

- Implement a function that facilitates K to be set as a function of time to catch up to X percentile grown Stalk per BDV at time of Deposit.
- Research into how to properly understand how Beanstalk should toggle K autonomously.

LP

Gauge

LP Gauge

The LP Gauge system allows Beanstalk to distribute Seeds arbitrarily to Deposit LP in an attempt to move the current LP BDV Distribution to its optimal LP BDV Distribution.

Properly setting the optimal LP BDV Distribution is probably one of the most interesting questions currently facing Beanstalk. Does Beanstalk live and die by the wisdom of the DAO? Otherwise, how can it set optimal exposure to the market? What is optimal exposure to the market for something like Beanstalk?

Perhaps the LP Gauge is useful for macro level signaling (*e.g.*, USDT vs USDC vs 3CRV) but dangerous if used for aggressive position management.

This is an example where just because Beanstalk has a tool at its disposal it doesn't mean it should be used aggressively.

Therefore, the initial implementation of the LP Gauge system is particularly simple.

LP Gauge

An optimal BDV ratio is determined arbitrarily via BIP. When new assets are whitelisted their optimal BDV ratio must be set. All others are reweighted accordingly unless explicitly changed.

At the beginning of each Season, Beanstalk measures the actual BDV ratio relative to the optimal BDV ratio for each asset on the LP Gauge whitelist.

The technical implementation supports arbitrary functions to implement changes to LP Gauge points for each asset on the LP Gauge whitelist. However, the current implementation uses a very basic approach to adjusting LP Gauge points.

If an asset is under allocated it receives one more LP Gauge point.

If an asset is over allocated it receives one less LP Gauge point.

To prevent an asset from gaining an excessive amount of points due to being underweight for an extended period of time, the max LP Gauge points possible is 10^X where X is a parameter of Beanstalk set arbitrarily by a function.

Future Work

- Research into how to create a setting so that a distributed set of participants can properly set the optimal BDV ratio between various LP. This problem seems exceedingly difficult to get right.
- Research into more a sophisticated understanding of position, direction and acceleration amongst LP tokens only.
- Research into a more sophisticated response to position and its derivatives.

Other RFCs

suggested by

this study

Other tweaks suggested by this study

Soil issuance change

Given the importance of minimizing excessive issuance of debt, and thereby entering into a death spiral, Beanstalk should issue Soil as the minimum of the time-weighted average SMA of deltaB and the instantaneous EMA of delta B.

Doing so will minimize instances where Beanstalk issues more debt than is necessary to cross P back above V at no other cost to Beanstalk.

Generalized Flood

In addition to being upgraded to be able to sell down to V across multiple liquidity pools, it likely makes sense to make the following changes to the trigger for a Flood:

- Make the trigger points for Price and D^R independent of the other points along those axes; and
- Include L2SR in the trigger for a flood.
- Reset Maximum Temperature to 1 and Seed Gauge weighting to entirely favor LP.

Generalized

Convert

Generalized Convert

Generalized Convert is the second half of the upgrade to the Beanstalk peg maintenance model that allows the Silo to become much more integrated with the model's response to current state at the beginning of each Season.

There is currently an immense amount of friction Converting between LP tokens within the Silo. Generalized Convert is necessary to allow the LP Gauge system to work properly.

The two upgrades are also complimentary because beyond the ability to apply a penalty and bonus for Conversions, whereas Seed Gauge is able to respond to the current state of Beanstalk by changing the incentive for Converting via changing the distribution of Seeds to LP vs Beans and amongst LP tokens, the benefits of which are realized over time, the Generalized Convert creates the opportunity for Beanstalk to respond to its current State by offering a Conversion incentive (disincentive) whose benefits (penalties) are realized immediately via a Stalk bonus and penalty system.

(1) how should Beanstalk classify a Convert?

Classifying Conversions

Some key questions to answer in determining what the relevant features of a Convert are:

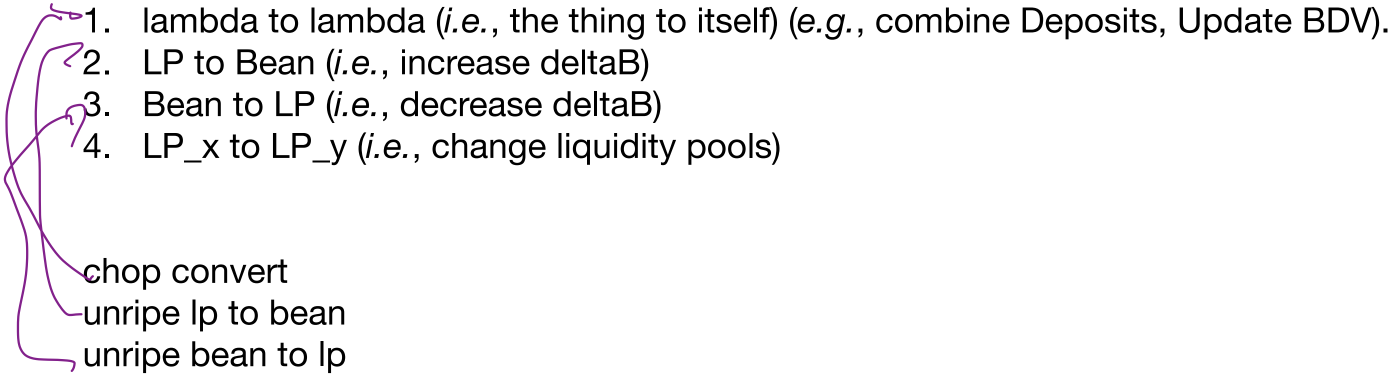
(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

(1.2) which data to use to evaluate a Convert along each axis?

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

Generalized Convert is a single Convert type that can handle all 4 Convert types:



1. lambda to lambda (*i.e.*, the thing to itself) (*e.g.*, combine Deposits, Update BDV).
2. LP to Bean (*i.e.*, increase deltaB)
3. Bean to LP (*i.e.*, decrease deltaB)
4. LP_x to LP_y (*i.e.*, change liquidity pools)

chop convert
unripe lp to bean
unripe bean to lp

(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

lambda to lambda Converts are to maximize the ease in which Farmers can use their Deposits to their full potential.

In certain instances, it may be beneficial to change the Season of Deposit of multiple Deposits (*e.g.*, a contract only accepts Deposits from after a certain Season, combine Deposits to save gas in the future).

When the BDV logged in the Silo is less than the current BDV of the Deposit, Farmers may want to pay gas to update their BDV upwards.

Therefore, the primary axis to evaluate lambda to lambda Converts on is how the distribution of grown Stalk per BDV (*i.e.*, opportunity cost for Withdrawing any given BDV) changes.

In particular, Beanstalk is concerned that there is no decrease in opportunity cost for any BDV as a result of the Convert, when adjusted for decreases in grown Stalk.

(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

LP to Bean Converts increase deltaB immediately and decrease L2SR over time.

While Beanstalk does not want any LP to Bean Converts when P > V, the Generalized Convert does not explicitly enforce a rule that they are not allowed to Convert when P > V. Instead, using a grown stalk penalty Beanstalk can properly discourage LP to Bean Converts when P > V.

Bean to LP Converts decrease deltaB immediately and increase L2SR over time.

While Beanstalk does not want any Bean to LP Converts when P < V, the Generalized Convert does not explicitly enforce a rule that Farmers are not allowed to Convert when P < V. Instead, using a grown stalk penalty Beanstalk can properly discourage Bean to LP Converts when P < V.

Therefore, change to deltaB is the primary axis over which to evaluate LP to Bean and Bean to LP Converts.

(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

LP_x to LP_y Converts do not affect deltaB or the L2SR other than via lost BDV to slippage. Therefore, how Beanstalk treats LP_x to LP_y Converts need not factor into the overall peg maintenance system, and should only be considered in the context of the LP gauge system.

Beanstalk does not want any LP_x to LP_y Converts when X is underweight and Y is overweight compared to the optimal LP BDV Distribution. Generalized Convert does not explicitly enforce a rule that Farmers are not allowed to. Instead, using a grown stalk penalty Beanstalk can properly discourage doing so.

Therefore, some measure of effect on LP BDV distribution relative to potential effect on LP BDV distribution is the primary axis over which to evaluate LP_x to LP_y Converts.

(1.1) what are the relevant axes/dimensions on/in which to evaluate a Convert?

Therefore, there are 3 axes over which a Convert should be evaluated by Beanstalk:

3 axes

- Change in the distribution of grown Stalk across BDV
- Change in deltaB
- Change in LP BDV distribution

(1.2) which data to use to evaluate a Convert along each axis?

Analysis of Converts by Beanstalk
The goal is to create the most generalized Convert possible. From a user's perspective it should be possible to perform an n x n Convert in a single transaction. However, allowing multiple different assets to be input into the Convert allows the swapping of opportunity cost between the assets.

For example, 100 BDV of BEAN:USDC w 0 grown Stalk and 100 BDV of BEAN:ETH w 100 grown Stalk are Converted into 100 BDV of BEAN:USDC w 100 grown Stalk and 100 BDV of BEAN:ETH w 0 grown Stalk.

This type of Conversion would allow for the circumvention of the bonus/penalty system for changes in LP BDV distribution.

For example, instead of Converting from LP_x to LP_y and receiving a penalty, you can Deposit new LP_y of equivalent BDV to LP_x, switch their grown stalk per BDV and then Withdraw LP_x with no penalty.

While suboptimal, Beanstalk can still create an n x n Conversion experience through the combination of n n x 1 Conversions in a single Pipeline transaction. However, the evaluation by Beanstalk of Conversions will be limited to Conversions that Convert Deposits of n assets to Deposits of 1 asset.

(1.2) which data to use to evaluate a Convert along each axis?

Change in the distribution of grown Stalk per BDV

When evaluating an n x 1 Convert, the fundamental principle is that **the opportunity cost of Withdrawing any amount of BDV from the Silo after the Conversion should be greater than or equal to the opportunity cost of Withdrawing the same amount of BDV from the Silo before the Conversion.** this principle prevents manipulation of opportunity cost via Generalized Converts.

Therefore, the grown Stalk per BDV in and out of the Convert, each sorted in ascending order by grown Stalk, is needed to efficiently ensure compliance with this rule.

(1.2) which data to use to evaluate a Convert along each axis?

Change in deltaB

When evaluating an n x 1 Convert, the change in deltaB is either helpful or hurtful in terms of peg maintenance. Therefore, Beanstalk must measure the change in deltaB (deltaDeltaB) as a result of the Convert and compare it to the BDV input to the Convert.

The ratio U = deltaDeltaB/BDV^in is the best metric for how aligned was the Convert with Beanstalk’s peg maintenance in terms of deltaB.

U is in [-1, 1]. The closer to negative 1, the larger the decrease in deltaB. The closer to 1 the larger the increase in deltaB.

(1.2) which data to use to evaluate a Convert along each axis?

Change in LP BDV Distribution

When evaluating an n x 1 Convert, the change in LP BDV distribution is neither helpful nor hurtful in terms of overall Beanstalk state.

however, it is either helpful or hurtful to the LP Gauge system, which is a secondary peg maintenance component.

therefore, Beanstalk should still measure the change in LP BDV distribution with respect to optimal (deltaLP) as a result of the Convert and compare it to the BDV input to the Convert.

The ratio S = deltaLP/BDV^in is one metric for how aligned was the Convert with Beanstalk's peg maintenance in terms of LP BDV Distribution.

A more sophisticated mechanism could factor in how far each LP is from its optimal distribution and weight certain conversions in deltaLP more than others.

This is a subject for future work.

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in the distribution of grown Stalk per BDV

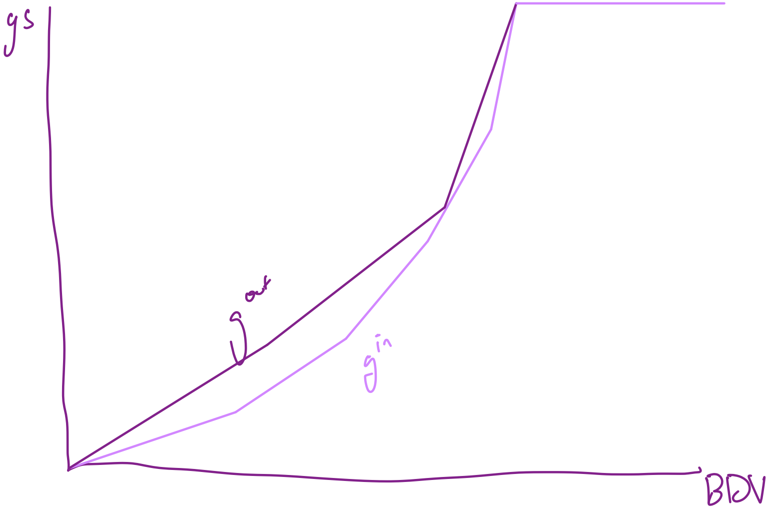
When evaluating an n x 1 Convert, the fundamental principle is that **the opportunity cost of Withdrawing any amount of BDV from the Silo after the Conversion should be greater than or equal to the opportunity cost of Withdrawing the same amount of BDV from the Silo before the Conversion.**

the mathematical representation of this rule is:

$$\sum_{i=0}^{BDV^{in}} g_i^{in} \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \; \forall i : i \leq \max(BDV^{in},\ BDV^{out})$$

where g is grown Stalk.

Visualization of the rule regarding the distribution of grown Stalk per BDV



(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in the distribution of grown Stalk per BDV

There are four special cases that must be accounted for by the formula:

4 special cases

a. BDV increases as a result of the Convert (*i.e.*, BDV^in < BDV^out)
b. BDV decreases as a result of the Convert (*i.e.*, BDV^out < BDV^in)
c. applying the grown Stalk bonus
d. applying the grown Stalk penalty

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in the distribution of grown Stalk per BDV

BDV increases

intuition: BDV can increase for a variety of reasons. in order to prevent the creation of new Deposits with 0 grown Stalk, BDV increases must be applied evenly across all grown Stalk.

condition: BDV^in < BDV^out

algo tweak: multiply each grown Stalk in per BDV by max(BDV^in/BDV^out)

new algo:

$$\sum_{i=0}^{BDV^{in}} g_i^{in} * \min(\frac{BDV^{in}}{BDV^{out}}, 1) \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \; \forall i : i \leq \max(BDV^{in},\ BDV^{out}), BDV^{in} - BDV^{out} < 0$$

Note, lists of Deposits into and Deposits out from the n x 1 Convert must be ordered from least grown Stalk per BDV to greatest to allow Beanstalk to efficiently compare the opportunity cost for Withdrawing any amount of BDV.

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

BDV decreases

intuition: BDV can decrease for a variety of reasons, including Withdrawing value from the system (*i.e.*, trying to circumvent to policy of burning grown Stalk upon Withdrawal). to properly account for BDV decreases Beanstalk must impose a penalty in the form of a loss of the grown Stalk associated with the lost BDV. when BDV decreases, Beanstalk:

1. enforce the distribution of the decrease in BDV evenly across all grown Stalk;
2. allow the distribution of the lost grown Stalk arbitrarily; or
3. enforce the distribution of the lost grown Stalk is by the BDV with the least grown Stalk per BDV.

All 3 options enforce the loss of the proportional amount of grown Stalk associated with the loss of BDV. However, option 3 leaves the Depositor with the highest opportunity cost after the Conversion is completed.

condition: BDV^in - BDV^out = deltaBDV > 0

algo tweak: start counting the grown Stalk per BDV at BDV deltaBDV

new algo:

$$\sum_{i=\max(0,deltaBDV)}^{BDV^{in}} g_i^{in} * \min\left(\frac{BDV^{in}}{BDV^{out}}, 1\right) \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \ \forall i$$

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

<u>Change in the distribution of grown Stalk per BDV</u>

Grown Stalk Bonus

while the component of Convert that this formula is addressing (i.e., ensuring opportunity cost for Withdrawing after Converting is maximized) is entirely independent from the bonus and penalty system, the formula must be able to account for the bonus and penalty to be applied.

intuition: award extra grown Stalk for Converts that help Beanstalk. Beanstalk can:

1. enforce the distribution of the bonus grown Stalk across particular BDV; or
2. allow the distribution of the bonus grown Stalk arbitrarily across BDV.

Both options enforce the same gain in grown Stalk associated with the bonus. However, option 2 leaves more flexibility for Farmers with no downside compared to option 1. Therefore, option 2 is used.

condition: eligible for bonus

algo tweak: due to the help of ordering Deposits by grown Stalk per BDV from least to greatest, the bonus can be allowed to be applied arbitrarily through the distribution of the bonus evenly across the BDV input to the n x 1 Conversion. here b is bonus and t is total grown Stalk input to the Convert.

new algo:

$$\sum_{i=\max(0,deltaBDV)}^{BDV^{in}} g_i^{in} * \min\left(\frac{BDV^{in}}{BDV^{out}}, 1\right) * \left(\frac{1+b}{t}\right) \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \; \forall i$$

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in the distribution of grown Stalk per BDV

Grown Stalk Penalty

while the component of Convert that this formula is addressing (i.e., ensuring opportunity cost for Withdrawing after Converting is maximized) is entirely independent from the bonus and penalty system, the formula must be able to account for the bonus and penalty to be applied.

intuition: total grown Stalk is haircut by (1 - penalty). Beanstalk can:

1. enforce the distribution of the lost grown Stalk evenly across all BDV;
2. allow the distribution of the lost grown Stalk arbitrarily across BDV; or
3. enforce the distribution of the lost grown Stalk is by the BDV with the least grown Stalk per BDV.

Each option enforces the loss of the same amount of grown Stalk associated with the penalty. However, option 2 leaves the potential to decrease the opportunity cost for Withdrawing some BDV to near 0. Option 3 is less applicable here than compared to covering the BDV decrease because in that case there is the potential for a disguised Withdrawal (and the effect is in some respects a Withdrawal of BDV from the Silo) whereas in this case there is a penalty being applied across all Deposits. Therefore, option 1 is used.

condition: penalty is applied

algo tweak: multiply grown Stalk in by 1 - p, where p is the penalty between 0 and 1.

new algo:

$$\sum_{i=\max(0,deltaBDV)}^{BDV^{in}} g_i^{in} * \min(\frac{BDV^{in}}{BDV^{out}},1) * (\frac{1+b}{t}) * (1-p) \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \; \forall i$$

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in the distribution of grown Stalk per BDV

Manipulation Resistant Values

BDV^in, g_i^in and g_i^out are all Beanstalk native values that can be queried without risk of manipulation because they are downstream of other manipulation resistant components (*i.e.*, the Deposit system).

BDV^out should use the Multi Flow Pump instantaneous EMA for BDV at the time of Conversion.

Similarly, Beanstalk should use the Multi Flow Pump instantaneous EMA to determine the deltaB at the time of Conversion.

using the MFP instantaneous EMA for BDV and deltaB is somewhat inefficient due to the EMA (there is some lag in accuracy) but as of now this is the least amount of lag without loss of multi-block MEV manipulation resistance. Therefore, it is the best source of data that Beanstalk has at the moment.

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in deltaB

U is a value between -1 and 1 which is calculated using the change in deltaB as a result of the Convert (deltaDeltaB) and the total BDV of the Convert (BDV^in).

deltaDeltaB can be measured based on the deltaB before and after the Convert.

BDV^in is already manipulation resistant.

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Change in LP BDV Distribution

The optimal BDV of each whitelisted asset is determined by the product of its percentage of LP BDV points and the total LP BDV in the Silo.

Each whitelisted asset has some deltaBDV between the actual BDV and the optimal BDV. Let’s call this set deltaBDV* = {deltaBDV*_1, … , deltaBDV*_n).

The convert has some effect on the actual BDV of potentially every LP. Let’s call this set deltaBDV = {deltaBDV_1, … , deltaBDV_n)

the bonus and penalty system can evaluate deltaBDV*, deltaBDV and BDV^in.

deltaBDV* can be calculated using the Silo-native values of BDV and total BDV in the Silo. Neither requires special treatment to be manipulation resistant.

deltaBDV can be measured based on the BDV before and after the Convert.

BDV^in is already manipulation resistant.

(1.3) how should Beanstalk evaluate a Convert along each axis in practice?

Observe that the Convert is evaluated with respect to opportunity cost per BDV. the bonus and penalty are applied, but there is no discussion of determining their values.

This is because the opportunity cost check is to prevent manipulation, but is generally unrelated to peg maintenance.

The penalty and bonus are peg maintenance tools that can be used to incentivize or disincentivize a given Convert.

The axes on which to analyze Converts that are relevant to peg maintenance should be considered for the penalty and bonus.

Therefore, deltaB and LP BDV distribution are the axes of analysis for the bonus and penalty.

(2) how should Beanstalk respond to a Convert in terms of acceptance, bonus and penalty?

Responding to Conversions

Some key questions to answer in determining what the proper response to an n x 1 Conversion are:

(2.1) Under what circumstances should Beanstalk accept a Convert?

(2.2) How should Beanstalk determine the appropriate bonus and penalty amounts?

(2.3) How should Beanstalk use these tools in practice?

(2.1) Under what circumstances should Beanstalk accept a Convert?

Beanstalk should accept an n x 1 Convert if and only if the grown Stalk distribution check that handles all 4 special cases passes:

$$\sum_{i=\max(0,deltaBDV)}^{BDV^{in}} g_i^{in} * \min(\frac{BDV^{in}}{BDV^{out}}, 1) * (\frac{1+b}{t}) * (1-p) \leq \sum_{i=0}^{BDV^{out}} g_i^{out} \; \forall i$$

where
deltaBDV = BDV^in - BDV^out
g = grown Stalk
b = bonus
t = total grown stalk in
p = penalty

(2.2) How should Beanstalk determine the appropriate bonus and penalty amounts?

Given Beanstalk's understanding of its current state and the Convert in question's effect on deltaB and LP BDV distribution, how should Beanstalk evaluate the appropriate bonus and penalty to apply?

In it's most simple form, a complete bonus and penalty system could include a bonus function and penalty function that take in current state, U, deltaBDV and deltaBDV* and output values between 0 and 1 ([0,1],[0,1]).

However, properly accounting for current state in addition to U and the change in BDV distribution can be done with near infinite complexity. There is no limit to the amount of analysis that can go into determining the bonus and penalty.

(2.2) How should Beanstalk determine the appropriate bonus and penalty amounts?

In order to prevent people from Withdrawing, Selling, Redepositing and then Converting back to the original asset to gain extra grown Stalk, there is a limit to how much grown Stalk can be given out as a bonus.

The maximum bonus is equal to the sum of the grown Stalk of the BDV of the destination asset with the least grown Stalk in the Silo at the time of Conversion, up to the BDV^in.

Applying the max bonus requires Beanstalk to efficiently be able to query the grown Stalk per BDV of an asset in the Silo in ascending order.

However, in theory, the bonus given should be a product of the return value of the bonus function and the max bonus.

There is no similar limit to the penalty. The penalty can be between 0 and 100%.

(2.3) How should Beanstalk use these tools in practice?

If sufficiently gas efficient, the maximum bonus should be calculated as the sum of grown Stalk of the BDV of the destination asset with the least grown Stalk in the Silo at the time of Conversion, up to the BDV^in. Otherwise, just the minimum grown Stalk per BDV can be stored and used instead.

For now, the penalty system should apply a 100% penalty for assets that move deltaB further from 0 and a 0% penalty for assets that move deltaB closer to 0. whether to apply a linear penalty or something more complex is an area for future study.

It is unclear how much the LP BDV Distribution should factor into the penalty and bonus system compared to deltaB. For now, it is not included in the bonus and penalty system but certainly should be considered in future implementations.

Future Work

- Research into determining the optimal way to understand the effect of a Convert on LP BDV Distribution.
- Research into setting the appropriate penalty based on U.
- Research into a more sophisticated way to determine the appropriate bonus relative to the maximum and the penalty.

Other Changes in RFCs to discuss

BDV decrease

Remove 10 block delay in Earned Bean issuance and implement a 2 Season delay in eligibility for Mints after Deposit

BDV Decrease

Because of the ability to update BDV of Deposits upwards, it is possible for Farmers to be credited for upside volatility in the BDV of their Deposits.

However, when their Deposits experience downside volatility Beanstalk is currently still evaluating the BDV of the Deposit at the higher value.

This is a problem because because stale data is fed to the peg maintenance mechanism.

The solution is to allow anyone to update the BDV of a Deposit Down.

This should create more efficiency in the peg maintenance mechanism.

Future Work

- Calculating BDV at the start of every Season is preferred to requiring Farmers to update BDV upwards and downwards.
- A system to “top up” BDV could be introduced via Tractor. Instructions like “if someone is going to update my BDV downwards, frontrun them to top up the Deposit”. such a system would likely be beneficial to Beanstalk.

Replace Minting Delay Strategy

Beanstalk always must prevent manipulation in the form of buying and Depositing, calling the sunrise and then Withdrawing and Selling. Because the minting of new Beans happens during the sunrise, the only way to prevent this manipulation from occurring is to either:

a) delay the distribution of Earned Beans until some amount of time after the sunrise call and distribute the Earned Beans to stalk that existed at the sunrise call and still exists at the time of distribution; or

b) delay the eligibility of Stalk to earn Beans for a full Season.

Whereas currently the Earned Beans are not distributed until some amount of time after the sunrise call (*i.e.*, manipulation resistance is added at the Stalk level), the Seed Gauge requires knowing the BDV of all the assets in the Silo at the time of the Sunrise in a manipulation resistant fashion.

in addition to being cleaner to implement in practice than a), the implementation of b) can more naturally support Silo BDV manipulation resistance.

Upon implementation of the Seed Gauge and (LP Gauge) Beanstalk will only consider BDV that was still in the Silo for 2 sunrise calls for distributing mints (and changing LP Gauge points).

Replace Minting Delay Strategy

However, the manipulation resistant implemented is far from perfect. Manipulation resistance often comes from the consideration of more time at the cost of accuracy. In the case of the new Minting Delay strategy, the lag in logging the correct BDV for each LP can be up to 2 full Seasons.

This makes using the BDV of each LP token in the Silo for peg maintenance will be using quite stale data.

In practice this only affects LP_x to LP_y Converts. It means that the LP Gauge system will be adjusting LP Gauge points based on data that is one full Season old.

Given the slow, marginal effects of changing the LP Gauge points, using slightly older data is not nearly as problematic as if this was related to something like minting Beans or Soil (*i.e.*, Beanstalk wants to avoid minting Beans when it is below V and Soil when it is above V as much as possible.)