# Fucking Arb Problem

Let's start by writing down the original problem, for reference:

$$\text{maximize: } U(\Psi) \tag{1}$$

$$\text{subject to: } \Psi = \sum\nolimits_{i=1}^{m} A_i(\Lambda_i - \Delta_i) \tag{2}$$

$$\varphi_i(R_i + \gamma_i \Delta_i - \Lambda_i) = \varphi_i(R_i), \ \ (i = 1, ..., m) \tag{3}$$

$$\Delta_i \geq 0, \ \ \Lambda_i \geq 0, \ \ (i = 1, ..., m) \tag{4}$$

Ok, let's dramatically simplify the presentation of this problem. First, the variables $\Delta_i$ and $\Lambda_i$ represent the vectors of tendered/received tokens on the $i^{th}$ market, written with respect to the local (market specific) index (where the matrix $A_i$ transforms them into the global indexing scheme). So, I hate four things about this notation, and I'm going to fix them.

- First, I keep forgetting which is tendered and which is received, so I'm going to use the letters $(t, r)$ instead of $(\Delta, \Lambda)$.
- Second, I keep forgetting that they are vectors, so I'm going to use vector arrows: $(\vec{t}, \vec{r})$
- Third, fuck the local indexing. Let's just assume they are already written in terms of the global indexing scheme. In other words, I'll take *as definition* the vectors $\vec{t}_i$ and $\vec{r}_i$ to be already transformed by $A_i$:

$$\vec{t}_i := A_i \Delta_i \tag{5}$$

$$\vec{r}_i := A_i \Lambda_i \tag{6}$$

- Fourth, I want to use the index $i$ for something else (my *super-vector* that we'll see later). So I'm going to use the index $m$ to keep track of the separate markets, with a total of $M$ markets. Therefore, instead of having $i = 1, .., m$, I'm using $m = 1, ..., M$.

With all of these changes in mind, then the original problem (1)-(4) can be rephrased as the following:

$$\text{maximize: } U(\vec{\Psi}) \tag{7}$$

$$\text{subject to: } \vec{\Psi} = \sum\nolimits_{m=1}^{M} (\vec{r}_m - \vec{t}_m) \tag{8}$$

$$\varphi_m(\vec{R}_m + \gamma_m \vec{t}_m - \vec{r}_m) = \varphi_m(\vec{R}_m), \ \ (m = 1, ..., M) \tag{9}$$

$$\vec{t}_m \geq 0, \ \ \vec{r}_m \geq 0, \ \ (m = 1, ..., M) \tag{10}$$

But there's one more change I want to make. Since we already decided that our utility function will be a simple weighted sum, i.e. $\vec{u}^T \cdot \vec{\Psi}$ for some utility vector $\vec{u}$, then let's just combine (7) and (8) into one line (no need for the $\Psi$ middleman variable):

$$\text{maximize: } \vec{u}^T \cdot \left( \sum\nolimits_{m=1}^{M} (\vec{r}_m - \vec{t}_m) \right) \tag{11}$$

$$\text{subject to: } \varphi_m(\vec{R}_m + \gamma_m \vec{t}_m - \vec{r}_m) = \varphi_m(\vec{R}_m), \ \ (m = 1, ..., M) \tag{12}$$

$$\vec{t}_m \geq 0, \ \ \vec{r}_m \geq 0, \ \ (m = 1, ..., M) \tag{13}$$

And lastly, just to be clear,

- the quantities $\vec{R}_m$, $\gamma_m$, and $\vec{u}$ are *given in advance*
- the quantities $\vec{t}_m$ and $\vec{r}_m$ are the *dynamic variables* with which we want to maximize

Ok but there's actually one more thing I want to do to rephrase the problem, on the next page.

Let's define the variable $\vec{x}$ consisting of the *stack* of all of our $\vec{t}_m$'s and $\vec{r}_m$'s.

$$\vec{x} := \begin{bmatrix} [\vec{t}_1] \\ [\vec{r}_1] \\ [\vec{t}_2] \\ [\vec{r}_2] \\ \vdots \\ [\vec{t}_M] \\ [\vec{r}_M] \end{bmatrix} \tag{14}$$

Furthermore, we note that the objective function given in (11) can be written as

$$\begin{aligned} \vec{u}^T\cdot \left(\sum_{m=1}^{M} (\vec{r}_m - \vec{t}_m)\right) &= \vec{u}^T\cdot \Big((\vec{r}_1 - \vec{t}_1) \;+\; (\vec{r}_2 - \vec{t}_2) \;+\; ... \;+\; (\vec{r}_M - \vec{t}_M)\Big) \\ &= \vec{u}^T\cdot \vec{r}_1 \;-\; \vec{u}^T\cdot \vec{t}_1 \;+\; \vec{u}^T\cdot \vec{t}_2 \;-\; \vec{u}^T\cdot \vec{r}_2 \;+\; ... \;+\; \vec{u}^T\cdot \vec{t}_M \;-\; \vec{u}^T\cdot \vec{r}_M \\ &= \Bigg[\, [\,-\vec{u}^T\,]\,[\,\vec{u}^T\,]\,[\,-\vec{u}^T\,]\,[\,\vec{u}^T\,]\,...\,[\,-\vec{u}^T\,]\,[\,\vec{u}^T\,] \,\Bigg]\cdot \begin{bmatrix} [\vec{t}_1] \\ [\vec{r}_1] \\ [\vec{t}_2] \\ [\vec{r}_2] \\ \vdots \\ [\vec{t}_M] \\ [\vec{r}_M] \end{bmatrix} &(15) \\ &= \vec{w}^T\cdot \vec{x} &(16) \end{aligned}$$

where I've defined the vector $\vec{w}$ as a bunch of alternating copies of $\vec{u}$:

$$\vec{w}^T := \Bigg[\, [\,-\vec{u}^T\,]\,[\,\vec{u}^T\,]\,[\,-\vec{u}^T\,]\,[\,\vec{u}^T\,]\,...\,[\,-\vec{u}^T\,]\,[\,\vec{u}^T\,] \,\Bigg] \tag{17}$$

Finally, let's define the functions $h_m$ that will encode the market maker formulas expressed in (12):

$$h_m\Big(\vec{x}\Big) := \varphi_m\Big(\vec{R}_m + \gamma_m\vec{t}_m - \vec{r}_m\Big) - \varphi_m\Big(\vec{R}_m\Big) \tag{18}$$

Note that it is legit to say that $h_m$ depends on $\vec{x}$, even though it only depends on the *particular chunck* of $\vec{x}$ given by $\vec{t}_m$ and $\vec{r}_m$. With all of these changes to our notation, we can once again rephrase the problem. Statements (11)-(13) now become

$$\begin{aligned} \text{maximize: } & f(\vec{x}) = \vec{w}^{\,T}\cdot \vec{x} &(19) \\ \text{subject to: } & h_m\Big(\vec{x}\Big) = 0 \;\; (m = 1, ..., M) &(20) \\ & \vec{x} \geq 0 &(21) \end{aligned}$$

Ok, one more thing... not every token is traded on every market, and so some of the components of $\vec{x}$ must be fixed at zero. For example, suppose that token $k$ (in the global indexing) is *not* traded on market $m$. Then we must have $\big(\vec{t}_m\big)_k = \big(\vec{r}_m\big)_k = 0$, always and forever. Since the vector $\vec{x}$ is just a stack of $\vec{t}$'s and $\vec{r}$'s, then this would result in $\big(\vec{x}\big)_{i_0} = \big(\vec{x}\big)_{j_0} = 0$, for some particular $i_0$ and $j_0$.

In light of this, let the index $i$ range over all of $\vec{x}$, and then let us introduce the following vocabulary: define an *active index* $i$ to be one that is *not fixed* to be zero. Similarly, we define an *inactive index* $i$ to be one that *is fixed* to be zero. Rather than imposing additional constraints such as $\vec{x}_i = 0$ for all inactive $i$, we should really just remove all of the inactive variables altogether from the whole problem (19)-(21), since these inactive variables are static and immaterial. The process of removing the inactive variables out of $\vec{x}$ will be made explicit when I work out an example later on.

Now, on to the fucking KKT conditions.

Supposing now that we've removed all the inactive variables from our vector $\vec{x}$ (and consequently remove them from $\vec{w}$ as well, so that the dimensions match), then let's define the following functions:

$$g_i(\vec{x}) = -x_i \tag{22}$$

i.e. it's just the negative $i^{th}$ component of $\vec{x}$. With this, then we can rephrase our problem yet again:

$$\text{maximize: } f(\vec{x}) = \vec{w}^{\,T}\boldsymbol{\cdot}\,\vec{x} \tag{23}$$
$$\text{subject to: } h_m\Big(\vec{x}\Big) = 0 \;\; (m = 1, ..., M) \tag{24}$$
$$g_i(\vec{x}) \leq 0 \;\; (i = 1, ..., \text{length of } \vec{x}) \tag{25}$$

**As far as I can tell, this setup is perfectly applicable to premises of the KKT theorem.**

In particular, the internet says that if $\vec{x}^*$ is a solution to the problem (23)-(25), then it can be characterized as follows. There exist values $\mu_i$ $(i = 1, ..., \text{length of } \vec{x})$ and $\lambda_m$ $(m = 1, ..., M)$ such that the following conditions are met:

$$\boxed{\textbf{Primal Feasibility}}$$

$$h_m\big(\vec{x}^*\big) = 0 \;\; (m = 1, ..., M) \tag{26}$$
$$g_i\big(\vec{x}^*\big) \leq 0 \;\; (i = 1, ..., \text{length of } \vec{x}) \tag{27}$$

$$\boxed{\textbf{Dual Feasibility}}$$

$$\nabla L(\vec{x}^*) = \nabla \left( f(\vec{x}) - \sum_i \mu_i g_i(\vec{x}) - \sum_m \lambda_m h_m(\vec{x}) \right)\Bigg|_{\vec{x}^*} = 0 \tag{28}$$
$$\mu_i \geq 0 \tag{29}$$

$$\boxed{\textbf{Complementary Slackness}}$$

$$\mu_i \; g_i\big(\vec{x}\big) = 0 \;\; (\text{for each } i) \tag{30}$$

In other words, we can interpret these conditions as follows:

- **Primal Feasibility:** If you think you've found a solution $\vec{x}^*$, it better at least satisfy the original constraints (24) and (25).
- **Dual Feasibility:** More interestingly, the solution $\vec{x}^*$ will be a critical point for the Lagrangian function defined by $L(\vec{x}) := f(\vec{x}) - \sum_i \mu_i g_i(\vec{x}) - \sum_m \lambda_m h_m(\vec{x})$. Also, the values of $\mu_i$ will be non-negative.
- **Complementary Slackness:** Of the inequality constraints in (25) ranging over $i$, those who are honest inequalities (and not just equal to zero) require the corresponding $\mu_i$ to be zero.

Now, notice that for us, the Lagrangian takes a pretty simple form. We already know that our objective function is $f(\vec{x}) = \vec{w}^{\,T}\boldsymbol{\cdot}\,\vec{x}$. But, if we arrange our collection of $\mu_i$ values in a vector $\vec{\mu}$, then in light of (22), the expression $\sum_i \mu_i g_i(\vec{x})$ can be written as

$$\sum\nolimits_i \mu_i g_i(\vec{x}) = -\vec{\mu}^{\,T}\boldsymbol{\cdot}\,\vec{x} \tag{31}$$

Therefore, the Lagrangian takes the somewhat simpler form of

$$L(\vec{x}) = \Big(\vec{w} + \vec{\mu}\Big)^T \boldsymbol{\cdot}\,\vec{x} - \sum\nolimits_m \lambda_m h_m(\vec{x}) \tag{32}$$

Therefore, when we take the gradient of the Lagrangian, we find

$$\begin{aligned}\nabla L(\vec{x}) &= \nabla\left[\left(\vec{w}+\vec{\mu}\right)^T \cdot \vec{x} - \sum\nolimits_m \lambda_m h_m(\vec{x})\right]\\ &= \left(\vec{w}+\vec{\mu}\right) - \sum\nolimits_m \lambda_m \nabla\Big[h_m(\vec{x})\Big]\end{aligned} \tag{33}$$

where we used the fact that $\nabla\big[\vec{v}^T \cdot \vec{x}\big] = \vec{v}$, for any vector $\vec{v}$.

Now, to find a point $\vec{x}^*$ that solves our optimization problem, the KKT conditions (26)-(30) can be summarized as the following:

$$\left\{\begin{array}{r} \nabla L(\vec{x}^*) = 0 \\ \mu_i g_i(\vec{x}^*) = 0 \\ h_m(\vec{x}^*) = 0 \end{array}\right\} \quad \text{with} \quad \left\{\begin{array}{c} \mu_i \geq 0 \\ \vec{x}^* \geq 0 \end{array}\right\} \quad (\text{for all } m) \tag{34}$$

Thus, we want to solve the (constrained) system of equations

$$\left\{\begin{array}{r} \partial_{x_i} L(\vec{x}) = 0 \\ \mu_i x_i = 0 \\ h_m(\vec{x}) = 0 \end{array}\right\} \quad \text{with} \quad \left\{\begin{array}{c} \mu_i \geq 0 \\ x_i \geq 0 \end{array}\right\} \quad (\text{for all } i \text{ and } m) \tag{35}$$

Does this system make sense? Suppose that the total number of tokens is $N$ (so for example $\vec{t}, \vec{r} \in \mathbb{R}^N$). Recalling that we have $M$ markets, then by definition (14), $\vec{x}$ has $2NM$ components, $\vec{x} \in \mathbb{R}^{2NM}$, as does the vector $\mu \in \mathbb{R}^{2NM}$. Thus, the index $i$ ranges over $i = 1, ..., 2NM$, while the index $m$ is still $m = 1, ..., M$, and so we see that the system (35) consists of $(2 \times 2NM) + M$ equations, or more simply, $(4N+1)M$. Meanwhile, when we list the unknown quantities $\{x_1, ..., x_{2NM}\}$, $\{\mu_1, ..., \mu_{2NM}\}$ and $\{\lambda_1, ...\lambda_M\}$, we see there are $(4N+1)M$ of these as well. Thus, system (35) consists of a $(4N+1)M \times (4N+1)M$ system of equations, and so we can expect well defined solutions could exist. Moreover, because of our linear utility function, it turns out that this system (35) actually *decouples* into $M$ simpler systems.

To see this, let's define subvectors $\vec{x}_m$ for each market by

$$\vec{x}_m := \begin{bmatrix} [\vec{t}_m] \\ [\vec{r}_m] \end{bmatrix} \tag{36}$$

with corresponding subvectors $\vec{w}_m$ given by

$$\vec{w}_m := \begin{bmatrix} [-u] \\ [u] \end{bmatrix} \tag{37}$$

and similarly for $\vec{\mu}_m$. We note that the AMM protocol $h_m(\vec{x})$ only applies to quantities on the $m^{th}$ market, so we may equivalently write $h_m(\vec{x}_m)$. Consequently, let us define $I_m$ to be the set of indices $i$ that pertain to the $m^{th}$ market. When we look at (33) we note that $\partial_{x_i}\big[h_m(\vec{x}_m)\big]$ is only non-zero if $i \in I_m$. Let's define $\nabla_m$ to be the gradient, but only with respect to the variables $x_i$ for $i \in I_m$. Then (33) can be rewritten more explicitly as the following:

$$\nabla L(\vec{x}) = \begin{bmatrix} [\vec{w}_1] \\ [\vec{w}_2] \\ \vdots \\ [\vec{w}_M] \end{bmatrix} + \begin{bmatrix} [\vec{\mu}_1] \\ [\vec{\mu}_2] \\ \vdots \\ [\vec{\mu}_M] \end{bmatrix} - \begin{bmatrix} [\lambda_1 \nabla_1 [h_1(\vec{x}_1)]] \\ [\lambda_1 \nabla_2 [h_2(\vec{x}_2)]] \\ \vdots \\ [\lambda_M \nabla_M [h_M(\vec{x}_M)]] \end{bmatrix} \tag{38}$$

Setting $\nabla L(\vec{x}) = 0$, and reading across the $m^{th}$ block, we see that for each $m$, we have the equation

$$\lambda_m \nabla_m \left[h_m(\vec{x}_m)\right] - (\vec{w}_m + \vec{\mu}_m) = 0 \tag{39}$$

Using (39) then, we see that (35) can be written (rearranged and partitioned) as the following:

$$\left\{\begin{array}{r}\hline \lambda_1 \partial_{x_i} h_1(\vec{x}_1) - (w_i + \mu_i) = 0 \;\; (\text{for } i \in I_1) \\ \mu_i x_i = 0 \;\; (\text{for } i \in I_1) \\ h_1(\vec{x}_1) = 0 \\ \hline \lambda_2 \partial_{x_i} h_2(\vec{x}_2) - (w_i + \mu_i) = 0 \;\; (\text{for } i \in I_2) \\ \mu_i x_i = 0 \;\; (\text{for } i \in I_2) \\ h_2(\vec{x}_2) = 0 \\ \hline \vdots \\ \hline \lambda_M \partial_{x_i} h_M(\vec{x}_M) - (w_i + \mu_i) = 0 \;\; (\text{for } i \in I_M) \\ \mu_i x_i = 0 \;\; (\text{for } i \in I_M) \\ h_M(\vec{x}_M) = 0 \\ \hline \end{array}\right\} \text{ with } \left\{\begin{array}{c} \mu_i \geq 0 \\ x_i \geq 0 \end{array}\right\} \tag{40}$$

In other words, we have $m$ decoupled systems of equations, each one being a $(4N+1) \times (4N+1)$ system of the form

$$\boxed{\left\{\begin{array}{r} \lambda_m \partial_{x_i} h_m(\vec{x}_m) - (w_i + \mu_i) = 0 \;\; (\text{for } i \in I_m) \\ \mu_i x_i = 0 \;\; (\text{for } i \in I_m) \\ h_m(\vec{x}_m) = 0 \end{array}\right\} \text{ with } \left\{\begin{array}{c} \vec{\mu}_m \geq 0 \\ \vec{x}_m \geq 0 \end{array}\right\}} \tag{41}$$

How can we solve such a system? When solving a non-linear system, all roads typically lead to Newton's method. Recall that for a function $\vec{F} : \mathbb{R}^k \to \mathbb{R}^k$, we can find a root $\vec{x}^*$ (so that $\vec{F}(\vec{x}^*) = 0$) by the iterative process:

$$\left\{\begin{array}{l} \text{initialize: } x_0 \\ \text{iterate: } \vec{x}_{n+1} = \vec{x}_n - \boldsymbol{J}^{-1}(\vec{x}_n)\vec{F}(\vec{x}_n) \end{array}\right\} \tag{42}$$

where $\boldsymbol{J}$ is the Jacobian of $\vec{F}$. If our initial guess $x_0$ is close enough to the root $\vec{x}^*$, then the iterates $\vec{x}_n$ should converge to the root $\vec{x}^*$. To map our problem (41) onto the algorithm (42), let's define the variable $\vec{\xi}_m$ and the function $\vec{F}_m$ by

$$\vec{\xi}_m := \begin{bmatrix} \vec{x}_m \\ \vec{\mu}_m \\ \lambda_m \end{bmatrix} \qquad \vec{F}_m(\vec{\xi}_m) := \begin{bmatrix} \begin{bmatrix} \vdots \\ \lambda_m \partial_{x_i} h_m(\vec{x}_m) - (w_i + \mu_i) \\ \vdots \end{bmatrix} \\ \begin{bmatrix} \vdots \\ \mu_i x_i \\ \vdots \end{bmatrix} \\ \Big[ h_m(\vec{x}_m) \Big] \end{bmatrix} \tag{43}$$

Clearly then, finding a $\vec{\xi}_m \geq 0$ such that $\vec{F}_m(\vec{\xi}_m) = 0$ is equivalent to solving (41). Unfortunately, we will need to compute the Jacobian. This is done by taking the gradient of each element of $\vec{F}_m$:

$$\boldsymbol{J}_m(\vec{\xi}_m) := \begin{bmatrix} \Big(\partial_{x_1}, \ldots, \partial_{x_{2N}}, \partial_{\mu_1}, \ldots, \partial_{\mu_{2N}}, \partial_{\lambda_m}\Big) \begin{bmatrix} \vdots \\ \lambda_m \partial_{x_i} h_m(\vec{x}_m) - (w_i + \mu_i) \\ \vdots \end{bmatrix} \\ \\ \Big(\partial_{x_1}, \ldots, \partial_{x_{2N}}, \partial_{\mu_1}, \ldots, \partial_{\mu_{2N}}, \partial_{\lambda_m}\Big) \begin{bmatrix} \vdots \\ \mu_i x_i \\ \vdots \end{bmatrix} \\ \\ \Big(\partial_{x_1}, \ldots, \partial_{x_{2N}}, \partial_{\mu_1}, \ldots, \partial_{\mu_{2N}}, \partial_{\lambda_m}\Big) \Big[ h_m(\vec{x}_m) \Big] \end{bmatrix} \tag{44}$$

It's a little tedious, but we can then compute the Jacobian:

$$\boldsymbol{J}_m(\vec{\xi}_m) := \begin{bmatrix} \lambda_m \begin{bmatrix} \partial_{x_1}\partial_{x_1} h_m & \partial_{x_2}\partial_{x_1} h_m & \dots & \partial_{x_{2N}}\partial_{x_1} h_m \\ \partial_{x_1}\partial_{x_2} h_m & \partial_{x_2}\partial_{x_2} h_m & \dots & \partial_{x_{2N}}\partial_{x_2} h_m \\ \vdots & \vdots & & \vdots \\ \partial_{x_1}\partial_{x_{2N}} h_m & \partial_{x_2}\partial_{x_{2N}} h_m & \dots & \partial_{x_{2N}}\partial_{x_{2N}} h_m \end{bmatrix} & \begin{bmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} & \begin{bmatrix} \partial_{x_1} h_m \\ \partial_{x_2} h_m \\ \vdots \\ \partial_{x_{2N}} h_m \end{bmatrix} \\ \begin{bmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \mu_{2N} \end{bmatrix} & \begin{bmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x_{2N} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ \begin{bmatrix} \partial_{x_1} h_1 & \partial_{x_2} h_1 & \dots & \partial_{x_N} h_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix} & \begin{bmatrix} 0 \end{bmatrix} \end{bmatrix} \tag{45}$$

A simpler way of writing this would be as

$$\boldsymbol{J}(\vec{\xi}_m) := \begin{bmatrix} \begin{bmatrix} \underset{(2N \times 2N)}{\lambda_m \nabla^2 h_m(\vec{x}_m)} \end{bmatrix} & \begin{bmatrix} \underset{(2N \times 2N)}{-1} \end{bmatrix} & \begin{bmatrix} \underset{(2N \times 1)}{\nabla h_m(\vec{x}_m)} \end{bmatrix} \\ \begin{bmatrix} \underset{(2N \times 2N)}{\mathrm{diag}(\vec{\mu}_m)} \end{bmatrix} & \begin{bmatrix} \underset{(2N \times 2N)}{\mathrm{diag}(\vec{x}_m)} \end{bmatrix} & \begin{bmatrix} \underset{(2N \times 1)}{0} \end{bmatrix} \\ \begin{bmatrix} \nabla h_m(\vec{x})_{\,(1 \times 2N)} \end{bmatrix} & \begin{bmatrix} 0_{\,(1 \times 2N)} \end{bmatrix} & \begin{bmatrix} 0_{\,(1 \times 1)} \end{bmatrix} \end{bmatrix} \tag{46}$$

where $\nabla^2 h$ represents the *Hessian* matrix of second derivatives.

## Let's Summarize

Let's recap:

- Given our current state of the system, $\{\vec{R}_m\}$, $\{\gamma_m\}$ and $\{h_m(\cdot)\}$ (for each market $m$), and our utility vector $\vec{u}$, then we define the variable $\vec{x}_m$ by (36), as the possible tendered and received amounts for each market, and we similarly define $\vec{w}_m$ by (37)
- We then define a new variable $\vec{\xi}_m$ by (43), using Lagrange multiplier variables $\vec{\mu}_m$ and $\lambda_m$, and we define the function $\vec{F}_m(\vec{\xi}_m)$ by (43).
- We look for roots of this function by employing Newton's method in (42), where the necessary Jacobian $\boldsymbol{J}_m$ is computed using (45) or (46).
- We find the roots to this function, which should solve the system (41), but we only accept a solution if it satisfies the inequality constraints in (41).
- We do this for each $m$ to obtain optimal vectors $\vec{x}^*_m$ that tell us how to choose the tendered and received amount for each market such that the overall utility function in (11) is maximized.

In principle, this should do the job. The problem is that the system may also have roots that violate the inequality conditions $\vec{x}_m \geq 0$ and $\vec{\mu}_m \geq 0$. If our initial guess isn't close enough to our desired solution, then Newton's method may converge to the negative solutions.

So I *did* do some testing. I did a really simple example of two markets trading two tokens:

| **Market:** | 1 | 2 |
|---|---|---|
| Tokens Traded: | $\{A, B\}$ | $\{A, B\}$ |
| Tendered Vector: | $\vec{t}_1 = \begin{bmatrix} x_1 \\ x_5 \end{bmatrix}$ | $\vec{t}_2 = \begin{bmatrix} x_8 \\ x_6 \end{bmatrix}$ |
| Received Vector: | $\vec{r}_1 = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$ | $\vec{r}_2 = \begin{bmatrix} x_7 \\ x_8 \end{bmatrix}$ |

I started out with with the reserves

$$\vec{R}_1 = \begin{bmatrix} 2 \\ 10 \end{bmatrix} \quad \vec{R}_1 = \begin{bmatrix} 5 \\ 4 \end{bmatrix} \tag{47}$$

(I'm assuming for simplicity that they have the same constant product, not that it matters.). We can solve this problem *by hand*. Because of my choice of $\vec{R}$, we can see that some of asset $A$ must move from market 2 to market 1, while some of asset $B$ must move from market 1 to market 2. This means that the solution must have $\{x_2, x_3, x_5, x_8\} = 0$. Think about it: if we are moving some of asset $A$ from market 2 to market 1, then we definitely don't *add* any $A$ to market 2, so the amount tendered to it ($x_8$) must be zero. Similar reasoning shows that $x_2$, $x_3$ and $x_5$ are zero as well.

Thus, our solution will only involve $\{x_1, x_4, x_6, x_7\}$. In particular, the constant product protocol means that we must have

$$\Big(R_{1A} + \gamma_1 x_1\Big)\Big(R_{1B} - x_4\Big) = R_{1A}R_{1B} \tag{48}$$

$$\Big(R_{2A} - x_7\Big)\Big(R_{2B} + \gamma_2 x_6\Big) = R_{2A}R_{2B} \tag{49}$$

and we want to maximize the utility function

$$f = u_A\Big(x_7 - x_1\Big) + u_B\Big(x_4 - x_6\Big) \tag{50}$$

subject to

$$\{x_1, x_4, x_6, x_7\} \geq 0 \tag{51}$$

You can check that this is all just a particular instance of the general problem outlined in (23)-(25). To solve this by hand, we can eliminate half of the variables by solving (48) and (49) for, let's say, $x_4$ and $x_7$, respectively:

$$x_4 = R_{1B} - \frac{R_{1A}R_{1B}}{R_{1A} + \gamma_1 x_1} \tag{52}$$

$$x_7 = R_{2A} - \frac{R_{2A}R_{2B}}{R_{2A} + \gamma_2 x_6} \tag{53}$$

$$\tag{54}$$

Then our utility function (50) becomes

$$f = u_A\left(R_{2A} - \frac{R_{2A}R_{2B}}{R_{2A} + \gamma_2 x_6} - x_1\right) + u_B\left(R_{1B} - \frac{R_{1A}R_{1B}}{R_{1A} + \gamma_1 x_1} - x_6\right) \tag{55}$$

We can find the maximum (critical point) of $f$ by taking the gradient with respect to $x_1$ and $x_6$ and setting it equal to zero:

$$\frac{\partial f}{\partial x_1} = \frac{u_B R_{1A} R_{1B}}{(R_{1A} + \gamma_1 x_1)^2} - u_A = 0 \tag{56}$$

$$\frac{\partial f}{\partial x_6} = \frac{u_a R_{2A} R_{2B}}{(R_{2B} + \gamma_2 x_6)^2} - u_B = 0 \tag{57}$$

These are easily solved for $x_1$ and $x_6$:

$$x_1 = \frac{1}{\gamma_1}\left(\sqrt{\frac{u_B R_{1A} R_{1B}}{u_A}} - R_{1A}\right) \tag{58}$$

$$x_6 = \frac{1}{\gamma_2}\left(\sqrt{\frac{u_A R_{2A} R_{2B}}{u_B}} - R_{2B}\right) \tag{59}$$

Now that we know the exact solution, we try the whole Newton's method thing outlined previously. For a utility function of $u = [2, 1]$, we solve this problem numerically and Matlab prints out the following result:

```
--------------------------------
(by algebra:)
         market 1:    market 2:
tndr[A]:  1.1658         0
tndr[B]:     0         2.3315
rcvd[A]:     0         1.8377
rcvd[B]:  3.6754         0

(by Newton's method:)
         market 1:    market 2:
tndr[A]:  1.161          0
tndr[B]:   -0          2.322
rcvd[A]:    0          1.833
rcvd[B]:  3.6659         0
--------------------------------
```

Ok, they're close! But weirdly not equal....? Not sure why... And then, for different initial values of our variable $\vec{\xi}$ (meaning different initial guesses for $\vec{x}$, $\vec{\mu}$, $\lambda$, etc.), we get something like this:

```
--------------------------------
(by algebra:)
         market 1:    market 2:
tndr[A]:  1.1658        0
tndr[B]:    0         2.3315
rcvd[A]:    0         1.8377
rcvd[B]:  3.6754        0

(by Newton's method:)
         market 1:    market 2:
tndr[A]:   -0         -8.1868
tndr[B]:    0         -10.3556
rcvd[A]:   -0          0
rcvd[B]:    0          0
--------------------------------
```

Like, what the hell?