Financial Optimization
ISE 347/447

Lecture 4

Dr. Ted Ralphs

Reading for This Lecture

- C&T Chapter 2

Some Conventions for Linear Optimization Problems

If not otherwise stated, the following conventions will be followed for lecture slides during the course:

- $\mathcal{P}$ will denote a polyhedron contained in $\mathbb{R}^n$.
- A will denote a matrix of dimension m by n.
- b will denote a vector of dimension m.
- x will denote a vector of dimension n.
- c will denote a vector of dimension n.
- $\mathcal{P}$ will either be defined in *standard form* ($\{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$) or *inequality form* ($\{x \in \mathbb{R}^n | Ax \geq b\}$).
- By default, we will be minimizing.

A Quick Review of Linear Optimization

Definition 1. *A* polyhedron *is a set of the form* $\{x \in \mathbb{R}^n | Ax \geq b\}$, *where* $A \in \mathbb{R}^{m \times n}$ *and* $b \in \mathbb{R}^m$.

Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a given polyhedron.

Definition 2. *A vector* $x \in \mathcal{P}$ *is an* extreme point *of* $\mathcal{P}$ *if* $\nexists y, z \in \mathcal{P}, \lambda \in (0, 1)$ *such that* $x = \lambda y + (1 - \lambda) z$.

Definition 3. *A vector* $x \in \mathcal{P}$ *is an* vertex *of* $\mathcal{P}$ *if* $\exists c \in \mathbb{R}^n$ *such that* $c^\top x < c^\top y \; \forall y \in \mathcal{P}, x \neq y$.

Basic Solutions and Extreme Points

Consider a polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n | Ax \geq b\}$ and let $\hat{x} \in \mathbb{R}^n$ be given.

Definition 4. *The vector $\hat{x}$ is a* basic solution *with respect to $\mathcal{P}$ if there exist n linearly independent, binding constraints at $\hat{x}$.*

Definition 5. *If $\hat{x}$ is a basic solution and $\hat{x} \in \mathcal{P}$, then $\hat{x}$ is a* basic feasible solution*.*

Theorem 1. *If $\mathcal{P}$ is nonempty and $\hat{x} \in \mathcal{P}$, then the following are equivalent:*

- *$\hat{x}$ is a vertex.*
- *$\hat{x}$ is an extreme point.*
- *$\hat{x}$ is a basic feasible solution.*

Example

$$
\begin{aligned}
\max \quad & 2x_1 + 5x_2 \\
\text{s.t.} \quad & -x_1 + 3.75x_2 \leq 14.375 \\
& x_1 - 3.4x_2 \leq 4.8 \\
& -1.625x_1 + 1.125x_2 \leq 1 \\
& 3.75x_1 - 1x_2 \leq 23.875 \\
& x_1 + x_2 \leq 12.7 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Example

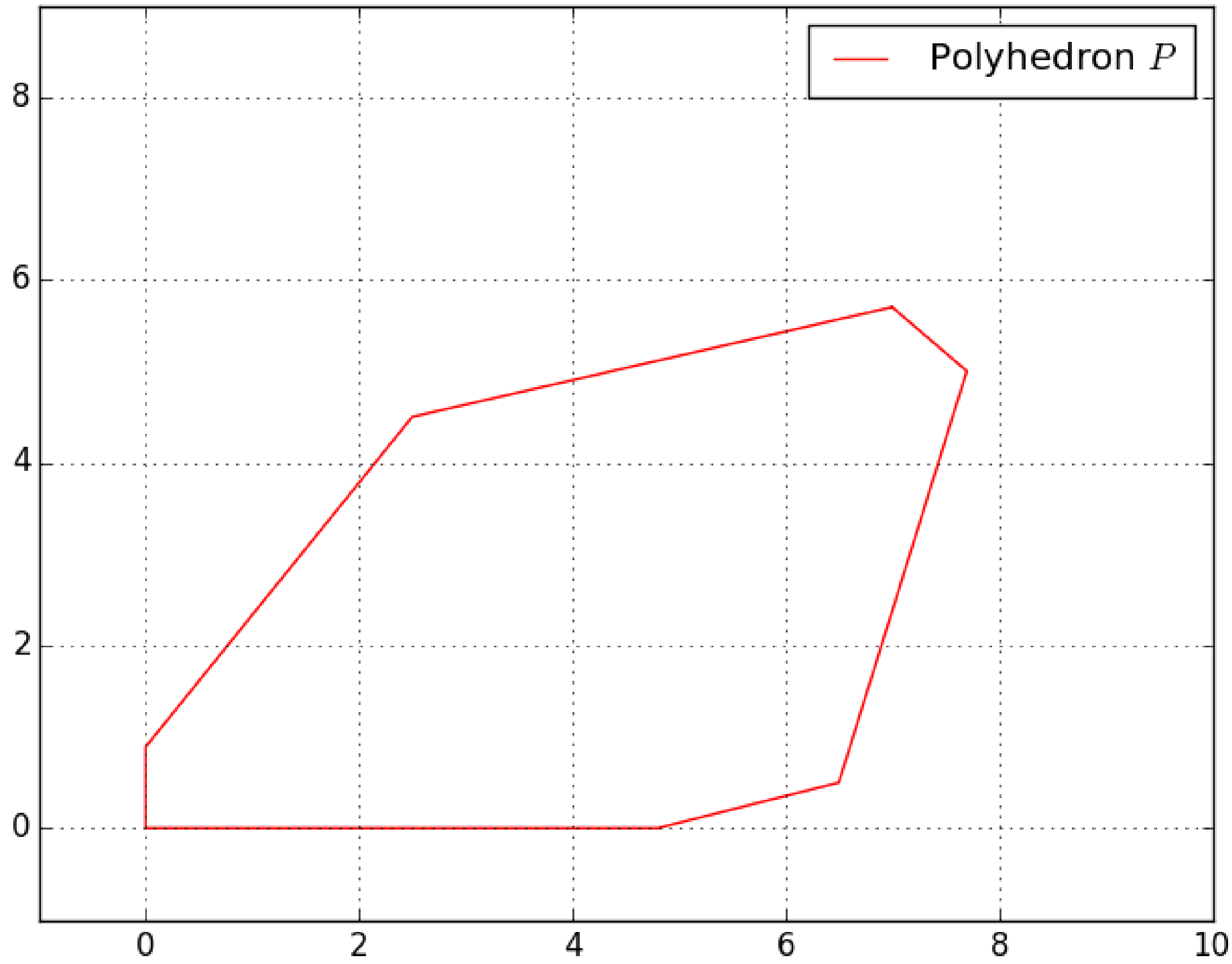

Figure 1: Feasible region for example

Polyhedra in Standard Form

- For the next few slides, we consider the standard form polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n | Ax = b, x \geq 0\}$.
- The feasible region of any linear optimization problem can be expressed equivalently in this form.
- We will assume that the rows of A are linearly independent $\Rightarrow m \leq n$.
- What does a basic feasible solution look like here?

Basic Feasible Solutions in Standard Form

- In standard form, the equations are always binding.
- To obtain a basic solution, we must set $n - m$ of the variables to zero (why?).
- We must also end up with a set of linearly independent constraints.
- Therefore, the variables we pick cannot be arbitrary.

Theorem 2. *Consider a polyhedron* $\mathcal{P}$ *in standard form with* m *linearly independent constraints. A vector* $\hat{x} \in \mathbb{R}^n$ *is a basic solution with respect to* $\mathcal{P}$ *if and only if* $A\hat{x} = b$ *and there exist indices* $B(1), \ldots, B(m)$ *such that:*

- *The columns* $A_{B(1)}, \ldots, A_{B(m)}$ *are linearly independent, and*
- *If* $i \neq B(1), \ldots, B(m)$*, then* $\hat{x}_i = 0$*.*

Some Terminology

- If $\hat{x}$ is a basic solution, then $\hat{x}_{B(1)}, \ldots, \hat{x}_{B(m)}$ are the *basic variables*.
- The columns $A_{B(1)}, \ldots, A_{B(m)}$ are called the *basic columns*.
- Since they are linearly independent, these columns form a *basis* for $\mathbb{R}^m$.
- A set of basic columns form a *basis matrix*, denoted B. So we have,

$$B = \left[A_{B(1)} \; A_{B(2)} \cdots A_{B(m)}\right], \quad x_B = \begin{bmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{bmatrix}$$

Basic Solutions and Bases

- Given a basis matrix B, the values of the basic variables are obtained by solving $Bx_B = b$, whose unique solution is $x_B = B^{-1}b$.
- However, multiple bases can give the same basic solution.
- Two bases are *adjacent* if they differ in only one basic column.
- Two basic solutions are adjacent if and only if they can be obtained from two adjacent bases (proof is homework).

Example: Basis Inverse

Basis inverse and corresponding solution when non-basic variables are s_3 and s_4:

```
[ s1    s2    s3    x1    x2    s5    s6    ]

[  1.    0.    0.    1.   -2.75  0.    0.  ]  [  3.32]
[  0.    1.    0.   -0.93  2.47  0.    0.  ]  [ 14.10]
[  0.    0.    1.    0.58 -0.55  0.    0.  ]  [  7.89]
[  0.    0.    0.    0.21  0.21  0.    0.  ]  [  7.70]
[  0.    0.    0.   -0.21  0.79  0.    0.  ]  [  5.00]
[  0.    0.    0.    0.21  0.21  1.    0.  ]  [  7.70]
[  0.    0.    0.   -0.21  0.79  0.    1.  ]  [  5.00]
```

Example

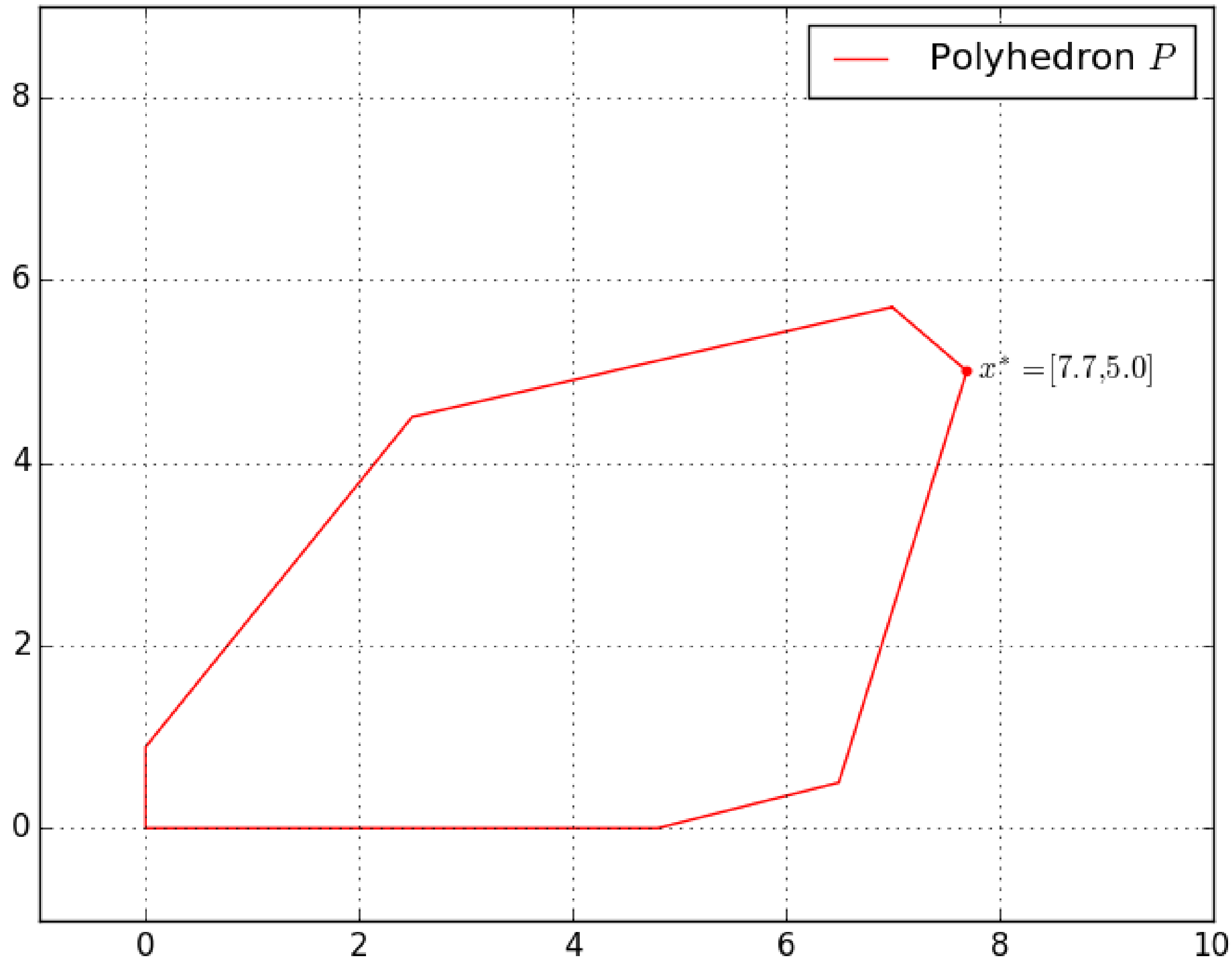

Figure 2: Basic solution when s_3 and s_4 are non-basic

Optimality of Extreme Points

Theorem 3. *Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a polyhedron and consider the problem $\min_{x \in \mathcal{P}} c^\top x$ for a given $c \in \mathbb{R}^n$. If $\mathcal{P}$ has at least one extreme point and there exists an optimal solution, then there exists an optimal solution that is an extreme point.*

- For linear optimization, a finite optimal cost is equivalent to the existence of an optimal solution.
- The previous result can be strengthened.
- Since any linear optimization problem can be written in standard form and all standard form polyhedra have an extreme point, we get the following:

Theorem 4. *Consider the linear optimization problem of minimizing $c^\top x$ over a nonempty polyhedron. Then, either the optimal cost is $-\infty$ or there exists an optimal solution.*

Iterative Search Algorithms

- Many optimization algorithms are *iterative* in nature.
- Geometrically, this means that they move from a given starting point to a new point in a specified *search direction*.
- This search direction is calculated to be both *feasible* and *improving*.
- The process stops when we can no longer find a feasible, improving direction.
- For linear optimization problems, it is always possible to find a feasible improving direction if we are not at an optimal point.
- This is essentially what makes linear optimization problems “easy” to solve.

Feasible and Improving Directions

Definition 6. *Let $\hat{x}$ be an element of a polyhedron $\mathcal{P}$. A vector $d \in \mathbb{R}^n$ is said to be a* feasible direction *if there exists $\theta \in \mathbb{R}_+$ such that $\hat{x} + \theta d \in \mathcal{P}$.*

Definition 7. *Consider a polyhedron $\mathcal{P}$ and the associated linear optimization problem $\min_{x \in \mathcal{P}} c^\top x$ for $c \in \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ is said to be an* improving direction *if $c^\top d < 0$.*

Notes:

- Once we find a feasible, improving direction, we want to move along that direction as far as possible.
- Recall that we are interested in extreme points.
- The simplex algorithm moves between adjacent extreme points using improving directions.

Constructing Feasible Search Directions in Linear Optimization

- Consider a BFS $\hat{x}$, so that $\hat{x}_N = 0$.
- Any feasible direction must increase the value of at least one of the nonbasic variables (why?).
- We will consider moving in *basic directions* that increase the value of exactly one of the nonbasic variables, say variable j. This means

$$\begin{aligned} d_j &= 1 \\ d_i &= 0 \text{ for every nonbasic index } i \neq j \end{aligned}$$

- In order to remain feasible, we must also have $Ad = 0$ (why?), which means

$$0 = Ad = \sum_{i=1}^{n} A_i d_i = \sum_{i=1}^{m} A_{B(i)} d_{B(i)} + A_j = B d_B + A_j \Rightarrow d_B = -B^{-1} A_j$$

Constructing Improving Search Directions

- Now we know how to construct feasible search directions—how do we ensure they are improving?
- Recall that we must have $c^\top d < 0$.

Definition 8. *Let $\hat{x}$ be a basic solution, let B be an associated basis matrix, and let c_B be the vector of costs of the basic variables. For each j, we define the* reduced cost *$\bar{c}_j$ of variable j by*

$$\bar{c}_j = c_j - c_B^\top B^{-1} A_j.$$

- The basic direction associated with variable j is improving if and only if $\bar{c}_j < 0$.
- Note that all basic variables have a reduced cost of 0 (why?).

Optimality Conditions

Theorem 5. *Consider a basic feasible solution $\hat{x}$ associated with a basis matrix B and let $\bar{c}$ be the corresponding vector of reduced costs.*

- *If $\bar{c} \geq 0$, then $\hat{x}$ is optimal.*
- *If $\hat{x}$ is optimal and nondegenerate, then $\bar{c} \geq 0$.*

Notes:

- The condition $\bar{c} \geq 0$ implies there are no feasible improving directions.
- However, $\bar{c}_j < 0$ does not ensure the existence of an improving, feasible direction unless the current BFS is nondegenerate

.

The Tableau

- The tableau looks like this

$-c_B^\top B^{-1}b$	$c^\top - c_B^\top B^{-1}A$
$B^{-1}b$	$B^{-1}A$

- In more detail, this is

$-c_B^\top x_B$	$\bar{c}_1$	$\cdots$	$\bar{c}_n$
$x_{B(1)}$ $\vdots$ $x_{B(m)}$	$B^{-1}A_1$	$\cdots$	$B^{-1}A_n$

Optimal Tableau in Example

Tableau and reduced costs when non-basic variables are s_3 and s_4:

```
[ 0.    0.    0.    0.    0.    -0.79 -1.21  0.    0.  ]

[-0.   -0.    1.   -0.   -0.     1.   -2.75 -0.   -0.  ] [  3.32]
[-0.   -0.   -0.    1.   -0.    -0.93  2.47 -0.   -0.  ] [ 14.10]
[-0.   -0.   -0.   -0.    1.     0.58 -0.55 -0.   -0.  ] [  7.89]
[ 1.    0.    0.    0.    0.     0.21  0.21  0.    0.  ] [  7.70]
[ 0.    1.    0.    0.    0.    -0.21  0.79  0.    0.  ] [  5.00]
[-0.   -0.   -0.   -0.   -0.     0.21  0.21  1.   -0.  ] [  7.70]
[-0.   -0.   -0.   -0.   -0.    -0.21  0.79 -0.    1.  ] [  5.00]
```

Example

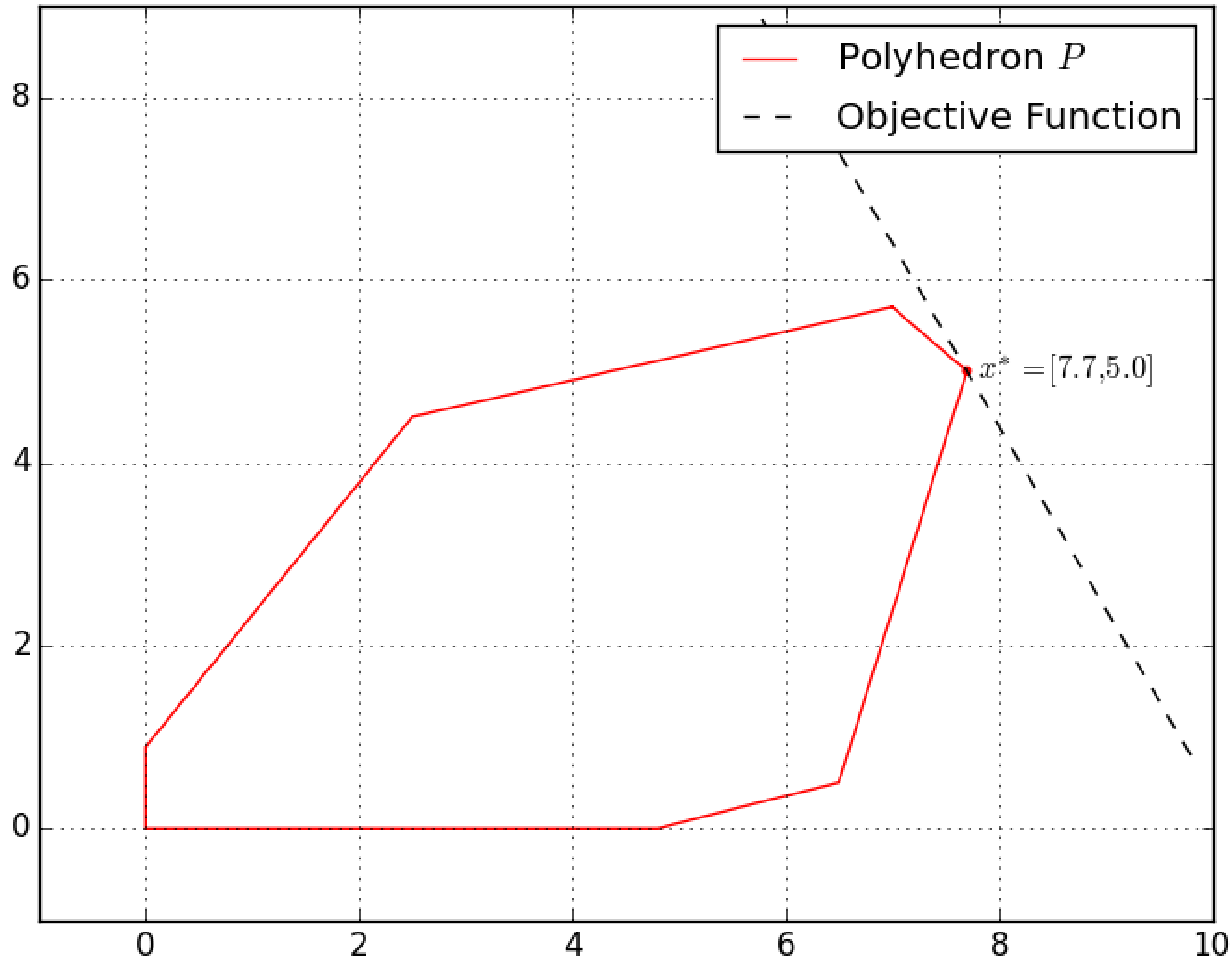

Figure 3: Optimal basic solution for example

Implementing the Simplex Method

"Naive" Implementation

1. Start with a basic feasible solution $\hat{x}$ with indices $B(1), \ldots, B(m)$ corresponding to the current basic variables.
2. Form the basis matrix B and compute $p^\top = c_B^\top B^{-1}$ by solving $p^\top B = c_B^\top$.
3. Compute the reduced costs by the formula $\bar{c}_j = c_j - p^\top A_j$. If $\bar{c} \geq 0$, then $\hat{x}$ is optimal.
4. Select the entering variable j and obtain $u = B^{-1} A_j$ by solving the system $Bu = A_j$. If $u \leq 0$, the LP is unbounded.
5. Determine the step size $\theta^* = \min_{\{i | u_i > 0\}} \frac{\hat{x}_{B(i)}}{u_i}$.
6. Determine the new solution and the leaving variable i.
7. Replace i with j in the list of basic variables.
8. Go to Step 1.

Example: Short Term Financing Revisited

Recall our previous example. A company needs to make provisions for a series of cash flows over a period of T months.

- The following sources of funds are available,
  - Bank credit
  - Issue of zero-coupon bonds
  - Cash reserves in an interest-bearing account.
- How should the company finance these cash flows if no payment obligations are to remain at the end of the period?

Example: AMPL Model for Short Term Financing

```
param T > 0 integer;
param cash_flow {0..T};
param credit_rate;
param bond_yield;
param invest_rate;

var credit {-1..T} >= 0, <= 100;
var bonds {-bond_maturity..T} >= 0;
var invest {-1..T} >= 0;

maximize wealth : invest[T];

subject to balance {t in 0..T} :
credit[t] - (1 + credit_rate)* credit[t-1] +
bonds[t] - (1 + bond_yield) * bonds[t-bond_maturity] -
invest[t] + (1 + invest_rate) * invest[t-1] = cash_flow[t];

subject to initial_credit : credit[-1] = 0;
subject to final_credit   : credit[T]  = 0;
subject to initial_invest : invest[-1] = 0;
subject to initial_bonds {t in 1..bond_maturity}: bonds[-t] = 0;
subject to final_bonds {t in T+1-bond_maturity..T}  : bonds[t] = 0;
```

Example: AMPL Data for Short Term Financing

These are the data for the example in the book.

```
param T := 5;

param : cash_flow :=
0     150
1     100
2    -200
3     200
4     -50
5    -300;

param credit_rate    := .01;
param bond_yield     := .02;
param bond_maturity := 3;
param invest_rate    := .003;
```

Example: AMPL Data for Short Term Financing

```
ampl: model short_term_financing.mod;
ampl: data short_term_financing.dat;
ampl: solve;
ampl: display credit, bonds, invest;
:    credit      bonds       invest      :=
0     0          150          0
1    50.9804      49.0196     0
2     0          203.434    351.944
3     0            0          0
4     0            0          0
5     0            0          92.4969
```

Example: Short Term Financing in PuLP

```
from pulp import LpProblem, LpVariable, lpSum, LpMaximize, value, LpStatus

from short_term_financing_data import cash_flow, credit_rate, bond_yield
from short_term_financing_data import bond_maturity, invest_rate

T = len(cash_flow)
prob = LpProblem("Short Term Financing Model", LpMaximize)

credit = LpVariable.dicts("credit", range(-1, T), 0, None)
bonds = LpVariable.dicts("bonds", range(-bond_maturity, T), 0, None)
invest = LpVariable.dicts("invest", range(-1, T), 0, None)

prob += invest[T-1]
for t in range(0, T):
    prob += (credit[t] - (1 + credit_rate)* credit[t-1] +
             bonds[t] - (1 + bond_yield) * bonds[t-int(bond_maturity)] -
             invest[t] + (1 + invest_rate) * invest[t-1] == cash_flow[t])
prob += credit[-1] == 0
prob += credit[T-1] == 0
prob += invest[-1] == 0
for t in range(-int(bond_maturity), 0):  prob += bonds[t] == 0
for t in range(T-int(bond_maturity), T): prob += bonds[t] == 0
```