Stocking up: Executive optimism, option exercise, and share retention
Online appendix

Rik Sen *
Hong Kong University
of Science and Technology
and University of Miami

Robert Tumarkin †
University of New South Wales

April 29, 2015

* riksen@ust.hk, Hong Kong University of Science and Technology, Department of Finance, Clear Water Bay, Kowloon, Hong Kong

† r.tumarkin@unsw.edu.au, School of Banking and Finance, University of New South Wales Business School, Sydney NSW 2052, Australia

1. Numerical Solution of Partial Differential Equations

We numerically solve all partial differential equations using the Du Fort-Frankel (1953) explicit, leapfrog, and unconditionally stable finite difference scheme. While traditional one-step finite difference schemes, such as the Euler explicit approach, are prone to numerical instability, the Du Fort-Frankel scheme has a unique approach to computing the time derivative that ensures the procedure is unconditionally stable numerically. Additionally, this approach is more computationally effective than one-step finite difference schemes. For a given finite difference grid, let h be the maximum number of grid steps in the state space, excluding time. Leapfrog schemes, such as the Du Fort-Frankel, generally require $O(h)$ steps in time for stability, while Euler explicit schemes require $O(h^2)$.

Before implementing the numerical procedure, we transform the state space of our partial differential equations to improve computational efficiency and accuracy. This focuses the numerical procedure on outside wealth and stock price values that are most likely to be encountered by the executive. The transformation is specified to ensure that our starting outside wealth and stock price values are located on the finite difference grid, which eliminates the need to interpolate our final results. We then numerically solve the transformed partial differential equations simultaneously using a Du Fort-Frankel scheme. Details about the solution method follow.

1.1. Transforming the PDEs and state space

Our specific transformation is most easily understood by first analyzing the state space, Ω, of the untransformed problem. Ω consists of all possible values of executive outside wealth, W, stock price, S, and time, t.

$$\Omega = \{(W, S, t) : (W, S, t) \in \mathbb{R}_+ \times \mathbb{R}_+ \times [0, T]\} \tag{1}$$

Numerical analysis requires that we restrict this infinite state space to a finite one, $\Theta \subseteq \Omega$

$$\Theta = \{(W, S, t) : (W, S, t) \in [W_{min}, W_{max}] \times [S_{min}, S_{max}] \times [0, T]\} \tag{2}$$

$W_{min}, W_{max}, S_{min}$, and S_{max} are selected to represent a broad range of possible agent outside wealth and stock price states given the initial conditions and market environment.

We then transform state variables $x \in (S, W)$ using

$$f(x) = (x - x_{min})/(A_x + D_x x) \tag{3}$$

where A_x, D_x, and x_{min} and free constants. We substitute this expression and appropriate derivatives into the original PDEs to derive new PDEs that operate over the transformed space.

The transformation is simply a restatement of the PDEs; it does not affect their economic interpretation. However, by using this transformation, we are able to more efficiently use computation power. The constants of transformation A_x, D_x, and x_{min} are selected so that, for a minimal value of x_{min}, $f(x_{min}) = 0$ and, for a maximum state value x_{max}, $f(x_{max}) = 1$. We also solve the equations so that, for an initial state value x_0, $f(x_0)$ is exactly on the grid. Finally, we pick parameter values to center the transformed grid around an area of interest. Qualitatively, if $f(x_0) < 0.5$, the grid is configured so that we have higher accuracy in states when $x_t > x_0$.

1.2. Implementing the Du Fort-Frankel scheme

As with all leapfrog schemes, the Du Fort-Frankel approach uses Partial Differential Equation (PDE) values at time steps t_{n+1} and t_{n+2} to compute values at t_n. Our model only has information at the terminal value. Therefore, in order to seed the leapfrog method, we need to know the value of the PDE at a second point in time. To compute this second time step value, we use an Euler explicit method, which only requires information at t_{n+1} to compute t_n. However, as the Euler explicit converges at a slower rate than the Du Fort-Frankel, we initially use $O(h^2)$ time steps during the Euler explicit phase before switching over to the Du Fort-Frankel scheme, with order $O(h)$ steps.

In other words, assume that time period T is equally divided into N steps for the Du Fort-Frankel scheme. We set the terminal conditions at step t_N. We then add N additional steps between time step t_{N-1} and t_N and use an Euler Explicit method to solve for PDE values at t_{N-1}. Finally, we apply the Du Fort-Frankel method to solve the PDE using the Euler Explicit solution at t_{N-1} and the terminal condition at t_N.

1.3. *Solving the executive's investment problem*

We solve the executive's problem, Eq. (9), as follows. At option expiration, we compute the value of the executive's utility function at the terminal payoff. We then step backwards incrementally in time. For each time step t_n, we first set appropriate boundary conditions as described below. With these boundary conditions in place, we assume all points lie within the continuation region, D. The optimal investment policy is determined, which is used to compute the numerical time derivative of the PDE.[1] We use this time derivative, along with information about the PDE values at previously computed time steps, to find the executive's continuation utility at t_n. We then determine the utility assuming option exercise, Eq. (11). Instead of solving a partial differential equation to compute the indirect utility function we use the well-known closed-form solution to the problem for a constant relative risk averse investor.[2] Finally, we determine the optimal exercise policy by comparing the continuation utility to that realized from exercising the option.[3] The maximum of the two utilities is used as the executive's realized utility under optimal investment and exercise.

Boundary conditions are required to implement the above procedure. Prior to expiration, we use the intuition provided by Carpenter, Stanton, and Wallace (2010). When either the stock or outside wealth values hits an upper boundary, Carpenter, Stanton, and Wallace show that the executive follows a nearly optimal policy, yielding a present value of the option approximately equal to the risk-neutral American option market price. Therefore, our boundary condition for the executive at both the maximum stock price and maximum outside wealth treats the option simply as part of the executive's wealth. The utility is equal to that which would be achieved under Eq. (11) assuming the executive has total wealth equal to her outside wealth and American option value, no outside option, and invests optimally. For the lower boundaries of stock price and outside wealth, we also use the American option value. This choice, while incorrect theoretically, represents an upper bound on the value and does not qualitatively impact the results. A lower bound value with zero for the option value provides similar results.

[1] This optimal policy along with the value of the PDE at t_{n+1} and t_{n+2} provide the Du Fort-Frankel estimate of the time derivative. In the Euler explicit phase, only the PDE value at time t_n is used.

[2] Using a numerical method to solve the agent's optimal investment problem without options provides similar results but slows the computation.

[3] With constant relative risk aversion, the indirect utility function after the options are exercised has the closed-form solution to the Merton (1969) portfolio choice problem. $V(W,\tau) = \frac{1}{1-\gamma} W^{1-\gamma} e^{(T-\tau)\epsilon}$, where ϵ is a constant.

The appropriate American option value represents optimal exercise under a risk-neutral measure. This value must also reflect the executive's exercise restriction, i.e. the option can not be exercised for $t < t_v$. We use a hybrid binomial tree-finite difference approach to value the American option. First, we compute the option value using a binomial tree for S_{max} and S_{min} at each point in time.[4] These provide the boundary conditions for the standard option pricing PDE. We then evaluate this PDE using the Du Fort-Frankel method. During the period where the option can be exercised early, we set the PDE value at each (S, t) equal to the maximum of the continuation and the immediate exercise values.[5]

The executive's portfolio choice problem is non-trivial when the agent is optimistic. Without the option, the executive wishes to hold a positive amount of the stock. However, as noted by Carpenter, Stanton, and Wallace, the portfolio choice opportunity set must be bounded to ensure that the numerical scheme converges. For low outside wealth, the convexity of the option makes the risk-averse executive's indirect utility function convex. As the numerical solution uses a discrete approximation to a continuous surface, this convexity can create numerical instabilities that propagate through the numerical grid if the executive is allowed to choose unbounded portfolio allocations. We assume that the executive cannot have a position greater than an exogenously specified multiple, M, of executive outside wealth in both the stock and the market. Per the short-sale constraints, we enforce $\omega_t^S \geq 0$. Together, these constraints require $0 \leq \omega_t^S \leq M$ and $-M \leq \omega_t^M \leq M$.[6] We experimented with a range of values to ensure that the constraint only affects the stability of the solution and does not impact the results presented in the paper. Similarly, different shaped constraint sets do not change the results. For example, a gross leverage constraint on the total portfolio, $|\omega_t^S| + |\omega_t^M| \leq M$, generates results nearly identical to those presented here. Finally, we always select leverage constrains such that the portfolio weights for the Merton solution are in the interior of our solution space.

Given the potential convexity of the indirect utility function, care must be taken when finding the optimal portfolio choice. We use the constraint set described above and consider internal, edge, and corner solutions. Specifically, we compute objective function values at nine potential solutions in (ω^S, ω^M) for each point

[4]Binomial tree approximations of derivative prices converge to true value as the number of steps increases. However, an "odd-even" swing occurs, in which the direction of the approximation error depends on whether the tree has an even or odd number of steps. Using a value that is averaged over an even and odd step binomial tree is known to improve accuracy. To that end, we average the values from a 200-step and 201-step binomial tree for each stock price.

[5]These computations can be done using a binomial tree approach. However, that method is quite slow. For a moderately sized grid, 100-point grid in the stock space and a 10,000-point grid in time, a pure binomial tree approach would require computing 1,000,000 binomial trees. This hybrid provides a high degree of accuracy with only a modest amount of computation time.

[6]These constraints can be interpreted as leverage constraints imposed on the executive by a broker.

(W, S, t) in the continuation set. For the first potential solution, we assume that the objective function is globally concave and compute the unconstrained first-order conditions. Four possible edge solutions are found by examining the boundary of the constrained portfolio choice set. We alternatively assume that the solution has $\omega_t^S = 0, \omega_t^S = M, \omega_t^M = -M$, and $\omega_t^M = M$, where M is the exogenously specified constraint as described above. We substitute each of these assumptions individually into the PDE and derive appropriate first-order conditions for the other variable. We reject any potential solution arising from first-order conditions that lie outside the constrained opportunity set. Finally, we consider the four potential corner solutions $(\omega^S, \omega^M) \in \{(0, -M), (M, -M), (0, M), (M, M)\}$. We select the optimal portfolio choice from among the candidates as the argument that maximizes the objective function.

1.4. Solving partial differential equations under the physical and risk-neutral measures

We solve the remaining PDEs to find the expected remaining time to expiration at exercise, the expected value ratio on exercise, and the expected proportion of stocks retained on exercise under their respective probability measures. In summary, our algorithm sets the terminal function value for each PDE. Then, it works backwards, implementing the Du Fort-Frankel scheme. For each time step, the solver sets boundary conditions and computes function values assuming all points lie within the continuation region. Then, it finds the set of (W, S, t) where the executive exercised the option early and changes the PDE value to reflect the value realized with early exercise as described below.[7]

For the expected proportion of stocks retained on exercise, our boundary conditions are found by dividing the optimal number of shares the executive retains on option exercise by the number of options granted. The expected time-to-expiration PDE boundary value is set to the time when the executive exercises the option. Finally, the value ratio PDE has a boundary condition at exercise equal to the ratio of the intrinsic value of the option divided by the American option value. The boundary condition for these three PDEs is 0 at the low stock price boundary, because it is highly unlikely that the option will ever be exercised.

[7] In practice, each PDE is solved simultaneously with the executive's investment problem using the Du Fort-Frankel method. Solving the equations simultaneously is not strictly necessary. However, this approach maximizes computational efficiency, requiring the optimal exercise policy only at the current time step and eliminating the need to store the full optimal conditional exercise policy decisions for the executive.

2. Identification of CEO Option Exercise Activity

The core sample used in our empirical tests derives from S&P Capital IQ's Compustat North America (Compustat) and Executive Compensation (ExecuComp) databases. CEOs in ExecuComp are marked with unique *execid* identifiers. For each CEO, we find option exercise and stock trading activity as collected by Thomson Reuters Insider Filings (Thomson). Thomson uses *personid* identifiers, which may not be unique for a person. Thomson identifiers are based on legal entity, so each entity through which an executive trades receives a unique *personid.* Option exercises and stock trades are reported by insiders on SEC Forms 3, 4, and 5. These forms require insiders to provide their name, address, and relationship to the company. However, as people may move, use abbreviated names, or change roles within a company, multiple IDs may be created for a single person in Thomson. Thomson does a very good job consolidating trades under different names. However, due to the sheer number of insider trades reported each year to the SEC, some executives may have trades assigned to several *personids.*

We undertook an extensive process of linking the ExecuComp and Thomson datasets to ensure we captured all trades by an insider and avoided developing a fragmented view of trading activity. To accomplish this, we use a computer-assisted manual matching process (described below) that helps us find all Thomson *personid* identifiers for each unique ExecuComp *execid.* Approximately 10% of the CEOs in ExecuComp have more than one *personid* identifier in Thomson.

2.1. Matching Process

We match Execucomp's *execid* identifier to Thomson's *personid* identifier using a computer-assisted manual matching process. For each CEO in ExecuComp, we first find a sample of all Thomson insiders who share a corporate affiliation with the ExecuComp CEO as described below. Any Thomson insider who has the exact name as in ExecuComp, including title and suffix, is automatically approved as a match. As people may have multiple *personids* in Thomson, we then evaluate all the remaining Thomson executives. These Thomson insiders are first sorted using the average text similarity score from five different fuzzy text algorithms comparing the Thomson executive name to that from ExecuComp. We then manually review these sorted insiders to find additional *execid-personid* matches. The fuzzy text algorithms, described below, are designed to maximize accuracy and efficiency of manual matching. The algorithms find matches with commonly encountered differences in name listings across databases, such as name abbreviations, nicknames, title and suffixes, misspellings, and word translocations. Finally, when we are unable to find a match for a Compustat CEO, we hand match on both person and company names using the full sample of Thomson executives without filtering for company affiliations.

2.2. Identifying the Sample of Thomson Insiders

Our core name-matching process requires a sample of Thomson insiders matching the corporate affiliations of an ExecuComp CEO. Both datasets contain CUSIP identifiers. However, as a firm's CUSIP may change, the CUSIP listed with an ExecuComp CEO may not be the same as that CEO's corresponding entry in Thomson. Therefore, we begin by building a comprehensive list of CUSIP identifiers for each ExecuComp CEO. The list contains all CUSIP identifiers in ExecuComp associated with a CEO's *execid* identifier. We supplement these observations by collecting all Compustat firm identifiers, *gvkeys*, associated with the *execid* and adding the linked CUSIPs from the CRSP/Compustat Merged Database. The sample of potential Thomson insiders consists of those with a CUSIP contained in the set of CUSIPs for the ExecuComp CEO *execid*.

2.3. Text Distance Measures

We use five different text distance measures to help us match names across the datasets. Each measure is normalized to rate name similarities on a scale from 0 to 1, with lower scores indicating more similar names. A brief description of each distance measure and its role in addressing common problems in matching names across databases are provided below. The algorithms are presented in decreasing order of sensitivity. Any pair of names that are determined to be close under a measure do at least as well, relatively, on any higher-numbered distance measure. In processing the names, we remove all punctuation and ensure uniform use of whitespace.

The first distance measure is a Damerau-Levenshtein (DL) distance algorithm, which helps identify names with spelling differences. The algorithm counts the number of operations required to transform one text string into another. Allowable operations include single letter insertion, deletions, and substitutions, and the transposition of two adjacent letters. We normalize the distance by dividing the number of operations by five and capping the quotient at one. The DL approach requires the names being compared to be structured similarly. That is, the words should appear in the same order, middle names should be present or absent in both versions of the name, and, when appropriate, the names being compared should have the same use of title and suffixes.

The second distance measure is a DL algorithm that allows for words to appear in different order in the two strings. This measure address problems when names may appear either last name first or first name first. To implement this algorithm, we create a set of all possible permutations of word order for the ExecuComp name. Then, we compute the normalized DL distance of the Thomson name with each permutation. The distance is the minimum score, allowing a maximum of five operations and normalizing the score to lie between 0 and 1 as above. As we find all permutations of the ExecuComp name and take the minimum DL distance with the Thomson name, there is no need to find word-order permutation of the Thomson name. This algorithm corrects for word order, but still suffers from various issues, such as when one database uses a middle name, title or suffix and the other database does not. Such issues are addressed by the remaining three algorithms.

The third distance measure is a modified cosine similarity score. This algorithm rates pairs of text strings based on shared words, weighting each word by the information it provides to the matching process. For each CEO, we create a corpus consisting of the ExecuComp name and the list of Thomson insider names from associated firms. Words in the corpus are assigned scores using a Term Frequency-Inverse Document Frequency (TF-IDF) weighting system, which gives extra weight to words that appear infrequently in the corpus. Each name is then considered in a vector space using these TF-IDF weights. To mimic the ranking system using in distance measures 1 and 2, where lower scores indicating better matches, we use a modified cosine similarity score as our measure of similarity. It is defined as one minus the cosine between any two name vectors. This algorithm helps identify executives who may use nicknames or common names sufficiently different from their legal names so that DL algorithms do not identify the match across the databases. The algorithm allows words to appear in any order in a name. Additionally, it will pick up matches that differ structurally, such as when one database uses a middle name and the other does not.

The fourth distance measure uses the modified cosine similarity score of algorithm 3, but allows for misspellings. For each word in an ExecuComp name, we build a vector of the word itself and all possible words found in the text corpus that differ by at most one DL operation. We then take a tensor product of these potential name vectors to create a set of alternate spellings for each ExecuComp name. We then compare each version of the name against the original Thomson name. The score is defined as the minimum modified cosine similarity score across all combinations of possible alternate spellings. By construction, this approach will always include the original spelling of the ExecuComp name and, therefore, will never perform worse than the cosine similarity approach used in the third distance measure. Unlike the second distance measure, this approach is not equivalent to finding all spelling permutations of both the ExecuComp name and the Thomson name due to the way the TF-IDF weighting scheme normalizes word similarity. However, we find no difference in the algorithm performance as implemented against a more general approach, but see an exponential benefit in computational time.

The final distance measure is designed to identify similar-sounding names. This helps identify executives across the Thomson and ExecuComp databases when there are multiple typographical errors. Each name is initially transformed using the SoundEx algorithm, which transforms each word into a code representing its phonetic structure. These transformed names are then fed through the fourth algorithm. As such, this approach allows us to compare names with slight variation in pronunciation, giving extra weight to phonetic structures that appear infrequently. As with the third and fourth distance measures, this approach allows us to compare names with words appearing in different order and with structural differences.