Float Types in Linear Schedule Analysis with Singularity Functions

By Gunnar Lucko¹, A.M.ASCE, and Angel A. Peña Orozco², S.M.ASCE

Abstract
This paper describes how float can be calculated exactly for linear schedules by using singularity functions. These functions originate in structural engineering and are newly applied to scheduling. They capture the behavior of an activity or buffer and the range over which it applies and are extensible to an infinite number of change terms. This paper builds upon the critical path analysis for linear schedules, which takes differences between singularity functions and differentiates them. It makes several important case distinctions that extend the earlier concept of rate float. Time and location buffers act along different axis directions. Together with different productivities between and within activities, this can create a complex pattern of critical and non-critical segments. Depending on starts and finishes, areas of float precede or follow these non-critical segments. The schedule of a small project is re-analyzed with case distinctions to demonstrate in detail what float types are generated.

CE Database Subject Headings: Scheduling; linear analysis; time dependence; location; two-dimensional analysis; critical path method; network analysis.

¹ Assistant Professor, Department of Civil Engineering, The Catholic University of America, Washington, DC 20064, email: lucko@cua.edu.

² Graduate Research Assistant, Department of Civil Engineering, The Catholic University of America, Washington, DC 20064, email: 63penaorozco@cua.edu.
1. Introduction
Planning and controlling the schedule of construction projects are among the most important functions of project managers. Whichever scheduling methodology is used, it should clearly and completely represent the project plan as it relates to sequencing across time, ideally augmented by data about other vital aspects, e.g. resource and space use, cost, and the direction of work.
Widely known in the construction industry, the critical path method (CPM) of scheduling yields information about the criticality of activities, i.e. the potential impact of delays, and their float, i.e. how such delays can be absorbed. However, it lacks the ability to model any spatial information (Lucko 2007b), which plays a major role in actual site operations. On the other hand, linear schedules integrate both time and space but important basic theory is still being developed. Previous studies required advanced mathematics that did not preserve the integrity of activities and omitted float (Russell and Caselton 1988) or used only a graphical approach for criticality (Harmelink and Rowings 1998) and float analysis (Harmelink 2001), whose algorithm has since been shown to contain inaccuracies (Kallantzis and Lambropoulos 2004b). Further studies depended on the graphical representation of the linear schedule for describing float (Ammar 2003, Awwad and Ioannou 2007). They described incremental float in terms of time or rate of progress, not continuously across activities. Activities in a linear schedule typically contain operations that repeat over several locations and thus are fewer but larger than in a comparable CPM schedule. Any measure of float should reflect the extent of these continuous activities.
Increasing the acceptance of linear schedules in the U.S. construction industry requires a complete analytical methodology that is at least competitive with CPM. Lucko (2007a) has therefore applied singularity functions to activities and buffers for a complete criticality analysis. Singularity functions are based on geometry but independent of any diagram. A complex activity is modeled with a single continuous and cumulative equation that always yields exact time and location data. This model shows clearly how activities can be partially critical, different from CPM and some studies on linear schedules that focused only on starts and finishes. This paper adds a comprehensive treatment of float to this new method. It enables calculating when and where activities can compensate for delays. Beyond that, float is an input for “resource leveling, time-cost trade-off, schedule updating, claims analysis, and dispute resolution” (Ammar 2003).

1.1 Linear Scheduling Method
As noted by Harris and Ioannou (1998), the literature knows various names for the method of scheduling activities of linear and repetitive projects in a two-dimensional coordinate system with an emphasis on productivity and resource continuity. This paper shall use the term linear scheduling method (LSM) (Johnston 1981). One of the two axes in the coordinate system is always a time axis. The other axis shows a unit of production, often in terms of a geometrical location toward which the workface is progressing. In this paper the time axis is plotted vertically so that equations $y(x) = f(x)$ can be superposed along the $x$-axis. Productivity is defined as the work that was produced divided by the time that it took. The slope of any line in a diagram thus shows the inverse of its productivity. While some readers may need to get accustomed to such a ‘vertical schedule’, it has been used in the literature, e.g. Harmelink and Rowings (1998). Various types of activities had been presented by Vorster et al. (1992), including bars, lines, and blocks, which were refined by Harmelink (1995) to continuous or intermittent fully or partially spanning lines and fully or partially spanning blocks. Such activities can be described with singularity functions, which are introduced in the next main section of this paper. If blocks are supposed to function as non-movable block-outs in the schedule to model a combination of time
and place at which no work is allowed to occur, e.g. a weekend or an inaccessible area, the block can also be modeled as a mathematical constraint using larger-than or smaller-than conditions.

1.2 Construction Projects
How well a project lends itself to analysis with LSM depends on its particular nature. Projects that are horizontally linear (Arditi and Albulak 1986), vertically linear (O’Brien 1975), or feature repetitive operations (Harris and Ioannou 1998) are ideally suited. Axis designations in the LSM diagram typically reflect the geometry of the physical facility under construction; a road building project is best represented with a horizontal location axis and a vertical time axis, but vice versa for a high-rise building project. These two types are thus mutually exclusive. Executing them may contain significant repetition, e.g. in placing multiple truckloads of hot asphalt mix along the road or in installing interior finishes on multiple floors of the high-rise building. Repetitive operations can therefore be considered the most general type, as reflected by numerous papers on the subject (e.g. Harris and Ioannou 1998, Ammar 2003, Awwad and Ioannou 2007). As long as the scheduler can identify a common unit of measurement for the axis that displays the work product, called amount axis by these authors, the construction project or parts thereof can be analyzed with LSM. It is possible to normalize the units of measurement for this axis. Different work products can be included in the same linear schedule by using cost in dollars or percent complete. The scheduler can then compare rates of progress of different activities or the planned versus actual progress of the same activity. Such normalization opens interesting opportunities for future research to combine LSM with concepts from earned value management, which calculates ratios of planned versus actual durations and costs that can be tracked across time.

1.3 Conflict Identification
The plane of the coordinate system helps to identify potential conflicts between two or more activities in a linear schedule. Such conflict may be a physical congestion where activities are in proximity to each other or an actual interference where activities are touching or crossing. Detecting instances where two or more activities would occur at the same time in the same location and removing them from a draft schedule by postponing individual activities, changing their productivity, or re-sequencing will make for a smoother execution and can reduce costs.

1.4 Time and Location Buffers
Buffers specify how close activities are allowed to get while being carried out. The buffers in this paper are similar to Arditi et al. (2002) and provide an envelope to the stepwise flow of work used by Hegazy (2001), who focused on integer crew allocations, and the zigzag flow of work used by Russell and Udaipurwala (2003), who replicated the resolution of a detailed CPM schedule. Each buffer follows the activity to which it is allocated to maintain a minimum distance to any successor. This differs from Kallantzis and Lambropoulos (2004a), who attribute buffers to the successor, not the predecessor. However, the new method presented in this paper is flexible enough to model either attribution. Future research will also extend it to a maximum distance, to model e.g. a newly drilled tunnel, where no more than an allowable distance can be unsupported before shotcrete is applied (Kallantzis and Lambropoulos 2004a). The new method can even model buffers that not just mimic the behavior of their host activity, but take on shapes of their own. Since the edge of the buffer is modeled with a separate equation, one can express e.g. an activity whose buffer decreases from five days initially to only three days later. Consequently, the slope of the buffer may vary from the slope of its host activity.
Not all activities need to feature buffers, but typically a schedule would feature an order of precedence in which activities are connected via buffers between them. Block activities can be considered stationary buffers without any slope that would indicate their progress. Buffers act in the direction of an axis. Two types of buffers can therefore be specified in LSM, which Harmelink and Rowings (1998) called least time and least distance interval. A time buffer allows any successor only to start after minimum duration has passed. A location buffer allows any successor only to start after a minimum distance has been gained. For non-geometric distances in repetitive operations, e.g. the condition “place ten cubic meters of concrete before consolidating the concrete with vibrators”, these authors recommend the more general term amount buffer.

1.5 Existing Criticality Analysis
Harmelink and Rowings (1998) used a graphical approach with an upward and downward pass for linear schedules, reminiscent of the forward and backward pass of CPM. They defined vertices as any start, finish, or change points within activities. Discrete buffers were installed at each vertex in the upward pass. Both time and location buffers were mentioned, but only the latter ones were demonstrated. All vertices were considered potentially critical. Controlling links then connected activities where actual minimum distances occurred. This created controlling and noncontrolling segments within them, the latter ones comprised the controlling activity path, “the longest [continuous] path of activities necessary to complete the project” (Harmelink 2001). The vertex concept is also found in the repetitive scheduling method (Harris and Ioannou 1998), which checks minimum distances only at control points in integer intervals instead of examining buffers along entire activities. The downward pass thus yields an incorrect location critical path for activities A and B in Figure 1 (Ioannou and Yang 2004, Kallantzis and Lambropoulos 2004b), as compared to Figure 2. This discrepancy is caused by strictly moving within activities from their finish to their start point (Harmelink and Rowings 1998): “If, while moving back in time along an activity, the beginning of the activity is reached before a potential controlling link with a preceding activity is reached, a new critical link is established at the beginning of the activity.”

<Figure 1: Controlling Activity Path (Harmelink and Rowings 1998)>
<Figure 2: Linear Schedule with Location Buffers>

At \( \{x, y\} = \{0, 5\} \), a link is correctly established from \( C_1 \) to \( B \), but the subsequent downward movement and link from the start of \( B \) to the middle of \( A \) ignores the closest distance between the finishes of \( B \) and \( A \). Kallantzis and Lambropoulos (2004a) presented a simpler graphical method that is correct for any activity shape and buffer type: “Each activity is formed as a poly line (…). Each activity is then placed the closest possible (in terms of time units) to its predecessor(s).” Lucko (2007a) has independently derived this activity stacking as part of a new linear schedule analysis. Under his method, all activities and their buffers are first modeled with continuous equations, as explained in Section 7. Then they are sequentially added to the schedule along the vertical time axis. Pairwise differences between neighboring equations are taken next and are differentiated to find all minima. Their positions are called critical points. They can occur at any start, finish, or change point (i.e. step or bend) within an activity or can be induced by neighboring activities or their buffers, which creates new segmentation within the current activity. In other words, the buffer of a predecessor touches the successor at such point. Finally, the minima are deducted from the initial equations for activity consolidation. Activities are connected via buffers between them at the critical points to form the critical path. This mathematical algorithm automatically yields the minimum project duration (Lucko 2007b).
Table 1 lists the duration and distance details for a schedule to construct 1,524 m (5,000 feet), or 50 stations, of road. It was examined under research for the Iowa Department of Transportation (Harmelink and Rowings 1995). Its algorithm was reviewed in several other studies (Mattila and Park 2003, Kallantzis et al. 2007). Figures 2 and 3 show the time and location buffer configurations, which were derived from the given schedule, whereas in a real project they would be specified beforehand. Buffers are shown as dark areas and the critical points that they cause are marked with small circles, leading to the two different critical paths, marked with thick lines, that connect the project start and finish points. Table 2 provides a side-by-side comparison of their numerical values and identifies significant differences for each activity except for A.

2. Existing Float Types in Critical Path Method
Two types of float are typically calculated in CPM analyses. Free float (FF) of an activity is the difference between the minimum of all earliest starts (ES) of direct successors and the earliest finish (EF) of the current activity. FF measures how much the current activity can be delayed without impacting any successors. It thus plays an important role for subcontractors, whose own delays could cause a ripple effect on subcontractors further downstream. Total float (TF) is the difference between the latest start (LS) and the ES of said activity, called start float, or alternatively the difference between the latest finish (LF) and the EF, called finish float. TF measures how much the current activity can be delayed without impacting the project finish, shown by a vertical solid line in Figure 5. It thus plays an important role for owners and general contractors, who wish to forecast if the entire project will finish on time. Interfering float (IFF) is the difference between TF and FF and measures how much delay can be absorbed within the entire project by delaying subcontractors. Independent float (IDF) is calculated as the minimum of all ESs of direct successors minus the duration of the current activity minus the maximum of all LFs of direct predecessors. It predicts the worst case scenario where “pressure” is exerted onto an activity from both sides (Halpin and Woodhead 1998). Safety float (SF) is the difference between the LS of the current activity and the maximum of all LFs of predecessors (Elmaghraby 1995). It measures how much the current activity can be delayed if all of its predecessors are delayed without impacting the project finish. The latter three float types are described in theoretical literature, e.g. Elmaghraby (1995) and are included for completeness. These authors use the terms ‘earliest’ and ‘latest’ instead of the more common ‘early’ and ‘late’ to indicate the ends of the spectrum in which an activity can occur. Figure 4 shows the various float types for the middle one of three activities within a larger schedule. The letters E or L indicate its earliest or latest possible constellation. Links to other activities, causing the indicated values, are omitted for clarity. The extent of each float type is shown by a thick dotted line attached to the activity.

3. Existing Float Types in Linear Scheduling
Harmelink (2001, emphasis in original) defined rate float as “the amount that the production rate of a noncontrolling linear activity can be lowered before the activity will become a controlling segment” and thus impact the project finish if delayed. Rate float was measured in units of work produced per time and materialized by rotating noncontrolling activity segments. No case distinctions were made. He identified three possible constellations within an activity: “Portions
of a linear activity that are not controlling segments can occur before the controlling segment, after the controlling segment, or both before and after the controlling segment. This means that, with respect to controlling and noncontrolling segments, a linear activity, at most, can be divided into three segments.” They are shown by activities $A$, $B$, and $C$ in Figure 5:

- Critical segment, non-critical segment;
- Non-critical segment, critical segment, non-critical segment;
- Non-critical segment, critical segment, critical segment.

<Figure 5: Schedule with Different Criticalities and Float Types>

The time float by Ammar (2003) “measures the amount of time a particular activity can be delayed without affecting the scheduled project date” and materialized by shifting activities parallel to the time axis. It was measured in time units and calculated as the difference between either the starts or the finishes of an activity and its direct successor. It was distinguished into total float and free float as under CPM. Awwad and Ioannou (2007) calculated rate float, which is equivalent to an incremental productivity change, and total float with and without enforcing resource continuity. Both studies assumed piecewise constant productivities and assessed buffers and float only at control points in the diagram, not over the entire range of activities. Changes can occur anywhere during their execution, but these graphical approaches will skip over minima between neighboring activities that occur between control points. These approaches cannot model activities that exhibit a curved behavior of higher order (e.g. due to learning or tiring). Currently available commercial software packages for LSM, including Control, TILOS, and LinearPlus, either do not contain any measures of float or at most present only time float as an equivalent to TF under CPM. Overall, these float concepts still have less explanatory power than float in CPM, much less do they fully reflect the two-dimensional nature of linear schedules.

4. Case Distinctions in Linear Scheduling

4.1 Time and Location Buffer

It must be distinguished what created the float in a linear schedule, time or location buffers. As shown by Table 2, the differences between the time and location critical paths can be significant. Schedulers need to be aware of this if they specify a time buffer, e.g. “the paint must dry for two days before any successor can start” or a location buffer, e.g. “the painter must be at least two apartments ahead of any successor.” The schedule of real construction projects may contain a mixture of constraints from time and location buffers. Some activities or segments thereof may be critical only under one of the two scenarios. These authors recommend the terms time or location buffer caused float to indicate the causality of the float (or criticality, respectively).

4.2 Converging and Diverging Activities

Comparing neighboring activities reveals whether the predecessor has a productivity that is larger than, equal to, or smaller than the productivity of its successor. Accordingly, assuming constant productivities, the two activities would be converging, parallel, or diverging during their execution. These activities or segments thereof may be non-critical and have float. For a converging pair of activities the float shrinks during their execution until the activities touch via a buffer between them. Since its maximum occurs between the starts of the activities, it is sensible to call this type early float. For a diverging pair of activities the float grows from where the activities touch. Since its maximum occurs between their finishes, it shall be called late float.

4.3 Convex and Concave Activities
If activities may have varying productivities, two types emerge. An activity with increasing productivity during its execution (whose curve gets flatter in diagrams with a vertical time axis), e.g. due to learning, they shall be called *convex activity*. For decreasing productivity, e.g. due to tiring, it shall be called *concave activity*. Together, the case distinctions create an array of possible constellations. Even the small linear schedule of Figure 5 already contains both early and late time buffer caused float. Activity $C$ in Figures 2 and 3 is convex and $E$ is concave.

### 5. New Float Types in Linear Scheduling

*Potential float* is visible as white areas between activities in the LSM diagram. However, only white areas adjacent to non-critical activities or segments thereof may be consumed as float. The float equation thus must only be evaluated across the non-critical segment, even if the white area extends beyond the critical point, e.g. from $x = 30$ to $300/7 \approx 42.86$ for activity $D$ in Figure 6.

<Figure 6: Linear Schedule with Float from Location Buffers>

#### 5.1 Free Float

It is not necessary to derive a new equation for the free float, because it is calculated as the difference between the minimum equation of any successor and the buffer equation of the current activity. Figures 6 and 7 show the time and location buffer caused *free float* as light areas. The pivots around which segments are rotating segments are marked with small circles. Note that an activity must start earlier than scheduled to consume early total float. Late free float thus occurs after an activity, but early free float occurs before it, e.g. the start of activity $D$ in Figure 7.

<Figure 7: Linear Schedule with Float from Time Buffers>

The free float equation yields time float (Ammar 2003) if it is evaluated at any location, especially its start and finish. The cumulative slope of the free float equation obtained is the inverse of the productivity. The ratio therefore needs to be inverted to be equivalent to rate float. Slopes must only be simplified by subtraction after inversion to adhere to basic arithmetic rules.

#### 5.2 Total Float

Different from CPM, the **total float** must be calculated after the free float, which it adds up. The case distinctions into time or location buffer caused float and early or late float apply. Its late float with an upward rotation is calculated as the maximum sum of all free float equations that follow the late non-critical segment until the next critical activity or the project finish is reached. Its early float, rotating downward, is the maximum sum of all free float equations that precede the early non-critical segment until the next critical activity or the project start is reached.

#### 5.3 Interfering Float

The **interfering float** is calculated in analogy to CPM as the difference between the total and free float equations. Accordingly, it is the sum of all free float equations following (for late float) or preceding (for early float) the current non-critical segment except for the directly next one.

#### 5.4 Independent Float

The **independent float** has three components, in analogy to CPM. Its late float is calculated as the minimum equation of any successor minus the difference between the buffer and activity equations of the current activity (i.e. the equivalent of its duration) minus the maximum sum of the buffer equation of any predecessor and its total float equation. Note that the total float
equation converts the buffer equations from their earliest to their latest configuration. Its early float is calculated by switching the words ‘successor’ and ‘predecessor’ in the previous sentence.

5.5 Safety Float
The safety float also uses the case distinctions. Its late float is calculated as the sum of the buffer equation of the current activity and its total float equation (i.e. the equivalent of its latest start) minus the maximum sum of the buffer equation of any predecessor and its total float equation. Its early float again switches ‘successor’ and ‘predecessor’ to reflect the downward rotation.

6. Definition of Singularity Functions
An integrated modeling approach originally conceived by Macaulay (1919) and Föppl (1927) for structural engineering analyses of beams under various loads has been newly applied to construction scheduling (Lucko 2007a). Eq. 1 contains the elementary term of these singularity functions. The first concept of this notation is to make a case distinction for the range where a function is valid or not, akin to an on/off switch that depends on the value of the variable \( x \).

\[
m \cdot (x-a)^n \begin{cases} 0 & \text{for } x < a \\ m \cdot (x-a)^n & \text{for } x \geq a \end{cases}
\]

Eq. 1

where \( x \) is a variable on the horizontal \( x \)-axis, \( a \) is the cutoff value on the \( x \)-axis at which the function becomes valid, the exponent \( n \) is the order of the curve segment that is modeled, and \( m \) is a scaling factor. For example, the order \( n = 0 \) generates a step of the height \( m \) at \( a \), and \( n = 1 \) generates a ramp of the slope \( m \) growing from \( a \), and \( n = 2 \) generates a quadratic curve. Eq. 1 is zero for all values of \( x < a \), and is evaluated normally with round brackets for all values of \( x \geq a \).

The second concept is that complicated shapes of \( y(x) \) can be modeled by superposition of any number of elementary terms. In other words, numerous terms can be compiled additively into a single equation. Within such singularity function, several terms may be simplified into one term by adding their \( m_1 \) and \( m_2 \) as per Eq. 2 if they share the same cutoff value \( a \) and exponent \( n \).

\[
m_1 \cdot (x-a)^n + m_2 \cdot (x-a)^n = (m_1 + m_2) \cdot (x-a)^n
\]

Eq. 2

Note that despite the complexity of shapes that can be described with singularity functions, no advanced mathematical concepts, e.g. a Fourier transform, will become necessary in this notation. A change term is added to the singularity function for each change that occurs in the behavior of \( y(x) \). The number of terms (before simplification using Eq. 2) thus is equal to one (the original term) plus the number of changes, regardless of their order. The singularity function is cumulative – with growing value of \( x \), more and more terms become active. If a term shall be active only for the range from \( a_1 \) to \( a_2 \), it must first be added at \( a_1 \) and then deducted again at \( a_2 \).

7. Modeling Activities and Buffers
Using the new notation, a linear schedule can be described completely by creating a system of singularity functions, one per activity and buffer. Eq. 3 gives the general model of Figure 8. For illustration purposes, it is assumed that the activity consists only of several straight segments (i.e. piecewise constant productivities), but in reality it may contain segments of any order \( n \).

\[
y(x) = \Delta t_0 \cdot (x-0)^0 + \frac{\Delta t_1}{\Delta a_1} \cdot (x-0)^1 + \left( \frac{\Delta t_1}{\Delta a_2} - \frac{\Delta t_1}{\Delta a_1} \right) \cdot (x-(\Delta a_1))^1 + \left( \frac{\Delta t_2}{\Delta a_3} - \frac{\Delta t_2}{\Delta a_2} \right) \cdot (x-(\Delta a_1 + \Delta a_2))^1 + ...
\]

Eq. 3

where \( y \) is the time variable, \( x \) is the location variable, \( \Delta t_i \) are durations on the \( y \)-axis, \( \Delta a_i \) are ranges on the \( x \)-axis, and the activity or buffer curve consists of the segments with the numbering
index \( j \). Eq. 3 contains the intercept \( t_0 \) in its first term, an initial slope \( \Delta t_1 / \Delta a_1 \) in the second term, and change terms in the rectangular brackets thereafter, where a new slope is added and the previous slope is deducted. These cumulative changes are indicated in Figure 8 by rotating from the tentative dashed lines to the actual solid lines. Eq. 3 can also be written with differences of points \( \{x_{j+1} - x_j\} \) and \( \{y_{j+1} - y_j\} \) instead of ranges \( \Delta x_j \) and \( \Delta y_j \), where \( j \) is a numbering index for segments of an activity or buffer that itself has the numbering index \( i \) within the schedule.

<Figure 8: General Model for Singularity Function>

8. Calculations
This section describes how to develop float equations. Only free float is presented for brevity; the other types can be derived as described in Section 5. It is assumed that non-critical segments are rotated and the continuity of activities is maintained. An optimization with interruptability that shifts segments (Vanhoucke 2006) is outside of the scope of this paper. Eqs. 4 through 9 list the activities for the example of Figures 2 and 3 after stacking and consolidation (Lucko 2007a). The differences of new minus old slope for activities \( C \) and \( E \) have been simplified as per Eq. 2.

\[
y(x)_i = 0 \cdot (x-0)^0 + \frac{7}{50} \cdot (x-0)^1 \\
y(x)_n = 4 \cdot (x-0)^0 + \frac{4}{50} \cdot (x-0)^1 \\
y(x)_c = 5 \cdot (x-0)^0 + \frac{6}{35} \cdot (x-0)^1 - \frac{11}{105} \cdot (x-35)^1 \\
y(x)_a = 8 \cdot (x-0)^0 + \frac{7}{50} \cdot (x-0)^1 \\
y(x)_x = 13 \cdot (x-0)^0 \cdot \frac{1}{30} \cdot (x-0)^1 + \frac{1}{6} \cdot (x-30)^1 \\
y(x)_e = 16 \cdot (x-0)^0 \cdot \frac{3}{40} \cdot (x-0)^1 + \frac{9}{40} \cdot (x-40)^1
\]

Eq. 4
Eq. 5
Eq. 6
Eq. 7
Eq. 8
Eq. 9

8.1 Time Buffer Equations
The time buffers (BT) in Eqs. 10 through 12 are easily derived from the equations of their host activities by adding the respective time buffer. Only time buffer equations that are relevant for float are shown for brevity. Activity \( F \) is last in the order of precedence and has no buffers.

\[
y(x)_C^{\text{BT}} = 6.9 \cdot (x-0)^0 + \frac{6}{35} \cdot (x-0)^1 - \frac{11}{105} \cdot (x-35)^1 \\
y(x)_D^{\text{BT}} = 9.8 \cdot (x-0)^0 + \frac{7}{50} \cdot (x-0)^1 \\
y(x)_E^{\text{BT}} = 16 \cdot (x-0)^0 + \frac{1}{30} \cdot (x-0)^1 + \frac{1}{6} \cdot (x-30)^1
\]

Eq. 10
Eq. 11
Eq. 12

8.2 Free Float Equations for Time Buffers
Eqs. 13 through 15 then take differences between the activity equation of the successor and the buffer equation of the current activity. Differences of ratios are not simplified prior to inversion.

\[
y(x)_D - y(x)_C^{\text{BT}} = 1.1 \cdot (x-0)^0 + \left( \frac{7}{50} - \frac{6}{35} \right) \cdot (x-0)^1 + \frac{11}{105} \cdot (x-35)^1
\]

Eq. 13
Finally, these time buffer caused free float equations can be evaluated at any \( x \)-value within the ranges of the non-critical segments from Table 2. Note that early and late float rotate into opposite directions in Figure 7, which causes e.g. Eq. 13 to yield the early float of activity \( D \) and the late float of segment \( C_2 \). At activity starts and finishes the float reaches the maxima in Table 3, which are equivalent to time float. For equivalents of rate float in Table 3, the differences of ratios are inverted and simplified. The inverted slopes from Eqs. 13 and 14 for activity \( D \) receive a small correction. The slope diagonally across the early float area from \{\( x, y \}\} = \{0, 6.9\} to \{30, 12.2\} is 5.3/30 and the light gray area shrinks at 7/50 - 53/300 = -11/300 instead of -11/350. The slope across the late float area from \{35, 12.9\} to \{50, 16.2\} is 3.3/15 = 11/50 but the light gray areas grows at the cumulative slope of 4/20 - 7/50 = 3/50.

\(<\text{Table 3: Maximum Time Buffer Caused Free Float}>\)

### 8.3 Location Buffer Equations

The location or amount buffer (BA) equations are derived from the equations of their host activities. They are shifted in the negative direction on the location axis, to the left in Figure 2, by the respective location buffer. The final cumulative slope is then deducted at \( x = x_{\text{max}} - \Delta x \), where \( \Delta x \) is the location buffer, to generate the “plateau”. The value of \( x_{\text{max}} \) is equal to the sum of the incremental ranges \( \Delta a \) over the segments \( a_1 \) in the activity. Finally, the equations are simplified using Eq. 16, whereby \( x \)-values for a left-shifted origin are replaced with \( x \)-values for an unshifted intercept, which accordingly lies higher on the \( y \)-axis due to the initial slope.

\[
y(x)_{\text{shifted}} = \Delta t_0 \cdot (x - (x_{\text{max}} - \Delta x)) + \frac{\Delta t_1}{\Delta a_1} \cdot (x - (x_{\text{max}} - \Delta x)) + \frac{\Delta t_1}{\Delta a_2} \cdot (x - (x_{\text{max}} - \Delta x)) + \ldots
\]

\[
+ \left[ \frac{\Delta t_1}{\Delta a_1} \right] (x - (x_{\text{max}} - \Delta x)) = y(x)_{\text{unshifted}} = \left( \Delta t_0 + \frac{\Delta x \cdot \Delta t_1}{\Delta a_1} \right) \cdot (x - 0) + \frac{\Delta t_1}{\Delta a_1} \cdot (x - 0) + \left[ \frac{\Delta t_1}{\Delta a_2} - \Delta t_1 \right] \cdot (x - (\Delta a_1))
\]

\[
+ \ldots + \left[ \frac{\Delta t_1}{\Delta a_k} \right] (x - (x_{\text{max}} - \Delta x))
\]

\[\text{Eq. 16}\]

Only relevant location buffer equations are shown in Eqs. 17 through 20 for brevity. Note that their intercepts differ from Eq. 5 through 8 because their consolidation used location buffers.

\[
y(x)_{a_1} = (4+1) \cdot (x - 0) + \frac{4}{50} \cdot (x - 0) - \frac{4}{50} \cdot (x - 37.5)
\]

\[\text{Eq. 17}\]

\[
y(x)_{a_2} = \left( 5 + \frac{114}{49} \right) \cdot (x - 0) + \frac{6}{105} \cdot (x - 0) - \frac{11}{105} \cdot \frac{x - 150}{7} - \frac{1}{15} \cdot \frac{x - 255}{7}
\]

\[\text{Eq. 18}\]

\[
y(x)_{a_3} = \left( 8 + \frac{9}{5} \right) \cdot (x - 0) + \frac{7}{50} \cdot (x - 0) - \frac{7}{50} \cdot \frac{x - 260}{7}
\]

\[\text{Eq. 19}\]

\[
y(x)_{a_4} = \left( 13 + \frac{7}{9} \right) \cdot (x - 0) + \frac{1}{30} \cdot (x - 0) + \frac{1}{6} \cdot \frac{x - 20}{3} - \frac{1}{5} \cdot \frac{x - 80}{3}
\]

\[\text{Eq. 20}\]

### 8.4 Free Float Equations for Location Buffers
The difference between the equation of the successor and the buffer equation of the current activity is then taken in Eqs. 21 through 24. Ratios are left unsimplified. The early float in Figure 6 is similar to Figure 7, as the location buffers prior to the plateau are maintained automatically.

\[
y(x)_c - y(x)_{\text{buffer}} = 0 \cdot (x-0)^6 + \left( \frac{6}{35} - \frac{4}{50} \right) \cdot (x-0)^5 \cdot (x-35)^5 - \frac{11}{105} \cdot (x-35)^4 + \frac{4}{50} \cdot (x-37.5)^1
\]

**Eq. 21**

\[
y(x)_a - y(x)_{\text{buffer}} = \frac{33}{49} \cdot (x-0)^6 + \left( \frac{7}{50} - \frac{6}{35} \right) \cdot (x-0)^5 \cdot (x-150)^5 + \frac{11}{105} \cdot (x-150)^4 + \frac{1}{15} \cdot (x-255)^1
\]

**Eq. 22**

\[
y(x)_e - y(x)_{\text{buffer}} = \frac{16}{5} \cdot (x-0)^6 + \left( \frac{1}{30} - \frac{7}{30} \right) \cdot (x-0)^5 \cdot (x-30)^1 + \frac{1}{6} \cdot (x-30)^1 + \frac{7}{50} \cdot (x-260)^1
\]

**Eq. 23**

\[
y(x)_e - y(x)_{\text{buffer}} = \frac{20}{9} \cdot (x-0)^6 + \left( \frac{3}{40} - \frac{1}{30} \right) \cdot (x-0)^5 \cdot (x-20)^1 - \frac{1}{6} \cdot (x-20)^1 + \frac{1}{5} \cdot (x-80)^1 + \frac{9}{40} \cdot (x-40)^1
\]

**Eq. 24**

Finally, these location buffer caused free float equations can be evaluated for any \(x\)-value within the ranges of the non-critical segments from Table 2. The maxima in Table 4 are again equivalent to time float. For equivalents of rate float in Table 4, the differences of ratios are inverted and simplified. The slope from Eq. 24 for activity \(F\) receives a small correction. The slope diagonally across the early float area from \(\{x, y\} = \{124/9 \approx 13.78, 0\}\) to \(\{80/3 \approx 26.67, 18\}\) is \(3/40 - 19/120 = -1/12\) instead of the initial \(1/24\). Note the exception for late float, where the value at the start of the plateau applies to the entire rectangle, in this case \(19/6 \approx 3.17\) for part of activity \(B\) from \(\{37.5, 8\}\), \(1.1\) for segment \(C_2\) from \(\{255/7 \approx 36.43, 12\}\), and \(9/21 \approx 0.43\) for part of activity \(D\) from \(\{260/7 \approx 37.14, 15\}\), but where the valid non-critical segments begin only at \(x = 37.5, 35, \text{and} 300/7 \approx 42.86\) as per Table 2. The slope across the late float rectangles in Figure 6 is their constant duration (equivalent to time float) plus the duration of the location buffer at the critical point, which is one day for activities \(B, C, \text{and} D\). The slope for activity \(B\) thus is \((19/6 + 1) / 12.5 = 1/3\), \((1.1 + 1) / 15 = 7/50\) for \(C\), and \((9/21 + 1) / (50/7) = 1/5\) for \(D\).

**<Table 4: Maximum Location Buffer Caused Free Float>**

9. Validation

Validating the free float from Section 8 applies the time and rate float concepts to the original example of Figure 1 (Harmelink 2001). However, the example did not clearly distinguish time and location buffers in its rate float analysis. Time buffers of two or three (between activities \(E\) and \(F\)) days were installed at all vertices (although their length is difficult to ascertain due to a lack of axis scales in four of the six diagrams). Yet the diagram contains a location critical path as shown by the thick dotted lines of the horizontal controlling links. Mixing these two cases may allow float that locally violates constraints or unnecessarily extends the project duration.

Two values for rate float were explicitly derived. The early and late rate floats of activity \(D\) were given as 1.875 stations per day from \(\{x, y\} = \{22.5, 11\}\) and 3.75 stations per day from \(\{42.5, 14\}\) (Harmelink 2001). Comparing these values with Table 2 shows that the vertex coordinates contain a “unit of measurement” inaccuracy that may have been caused by using four marks per ten stations. Activity \(D\) takes 7 days for 50 stations. Interpolating with this slope from its start at 8 days reaches 150/7 = 21.43 stations at 11 days and 300/7 = 42.86 stations at 14 days.

Correcting for these issues in Figure 1 and using the given time floats of one day each yields early and late rate floats of 50/7 - 75/14 = 25/14 \(\approx 1.76\) stations per day and 50/7 - 25/5 = 25/7 \(\approx 3.57\) stations per day. The early and late rate floats of activity \(D\) are the differences of the inverted slopes, which are 1.48 and 2.60 for time buffers in Table 3, and 1.31 and 2.14 for location buffers in Table 4. The productivity of the respective non-critical segments of \(D\) can
decrease by these values without impacting the project finish or violating the constraint of the particular buffer. In comparison, the case distinctions presented in this paper give a more detailed view of rate float. The example somewhat overestimated the rate float of activity $D$ because it used shorter non-critical segments from location buffers with longer time float from time buffers for a larger-than-actual rotation. Other activities showed similar deviations in their rate float. Time buffer caused early and late free floats in Table 3 are 1.1 and 1.2 and location buffer caused free floats in Table 4 are 0.67 and 0.43. The activities can take longer by these values without impacting the project finish. The example overestimated the time float for location buffers. Another explanation is that it rounded to full days to simplify the steps of its graphical analysis. The case distinction into early and late float becomes important for location buffers. While the time buffer caused free float creates only triangular shapes, Figure 6 shows that the late float has a rectangular shape to maintain the constraint of the location buffers across the plateau of the predecessor. Location buffer cause late float differs significantly from the rate float concept. An advantage of this approach over analyzing separate linear functions for each range (Al Sarraj 1990) or using individual vectors to describe line segments for each range (Russell and Caselton 1988) is the ability to manipulate the singularity function as a whole, e.g. differentiate it to determine its rate of change or integrate it to determine its area. The ability to add or subtract terms of any order $n$ at any location $a$ makes singularity functions a uniquely flexible way of modeling complex behaviors of a dependent variable $y(x)$. The geometric intricacy of rate float creates a need to computerize the calculations, especially for larger schedules. Current research also investigates issues such as interruptability, extending the new method to resource and cost analysis, and integrating earned value management, probabilistic durations, and fuzzy logic.

10. Conclusions
Users of CPM scheduling are familiar with different types of float. Some of these had already been translated into equivalent concepts for LSM, while others have been described for the first time in terms of linear schedules by this paper. This paper has used singularity functions for calculating these float types. The mathematical model relies on equations that describe activities and their buffers over a continuous range. Float at any location on a non-critical segment can be determined exactly. Free float is calculated as the difference between the equation of the successor and the buffer equation of its predecessor. Schedulers are thus equipped with various measures to determine how their linear or repetitive construction projects are impacted by delays.
Appendix I: References
Harmelink, D. J. (1995). “Linear scheduling model: the development of a linear scheduling model with micro computer applications for highway construction project control.” A dissertation submitted to the Graduate Faculty in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Mayor: Civil Engineering (Construction Engineering and Management), Iowa State University, Ames, Iowa.


Appendix IIa: Notation
The following symbols and abbreviations are used in this paper:

a  = location (amount), cutoff value
CPM = critical path method
EF  = earliest start
ES  = earliest finish
FF  = free float
IDF = independent float
IFF = interfering float
LF  = latest start
LS  = latest finish
LSM = linear scheduling method
m  = scaling factor, slope, inverse of productivity
n  = exponent
SF  = safety float
TF  = total float
t  = time
x  = variable along horizontal axis
y  = variable along vertical axis
Δ  = delta, difference
⟨ , ⟩ = brackets of singularity function
Appendix IIb: Superscripts and Subscripts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>location (amount) buffer</td>
</tr>
<tr>
<td>BT</td>
<td>time buffer</td>
</tr>
<tr>
<td>i</td>
<td>numbering index for activities of a schedule</td>
</tr>
<tr>
<td>j</td>
<td>numbering index for segments of an activity</td>
</tr>
<tr>
<td>k</td>
<td>number of segments in an activity</td>
</tr>
</tbody>
</table>
List of Figure Captions

Figure 1: Controlling Activity Path (Harmelink and Rowings 1998)
Figure 2: Linear Schedule with Location Buffers
Figure 3: Linear Schedule with Time Buffers
Figure 4: Float Types from Critical Path Method
Figure 5: Schedule with Different Criticalities and Float Types
Figure 6: Linear Schedule with Float from Location Buffers
Figure 7: Linear Schedule with Float from Time Buffers
Figure 8: General Model for Singularity Function
Figure 1: Controlling Activity Path (Harmelink and Rowings 1998)
Figure 2: Linear Schedule with Location Buffers
Figure 3: Linear Schedule with Time Buffers
Figure 4: Float Types from Critical Path Method
Figure 5: Schedule with Different Criticalities and Float Types
Figure 6: Linear Schedule with Float from Location Buffers
Figure 7: Linear Schedule with Float from Time Buffers
Figure 8: General Model for Singularity Function
List of Table Captions

Table 1: Linear Schedule Activities
Table 2: Time and Location Critical Path
Table 3: Maximum Time Buffer Caused Free Float
Table 4: Maximum Location Buffer Caused Free Float
### Table 1: Linear Schedule Activities

<table>
<thead>
<tr>
<th>Name</th>
<th>Segment</th>
<th>Activity</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>C</td>
<td>C₁</td>
<td>6</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>C₂</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>E</td>
<td>E₁</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>E₂</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>F</td>
<td>F₁</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>F₂</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Segment</td>
<td>Time</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>A</td>
<td>0.0 to 7.0</td>
<td>0 to 50</td>
<td>0 to 7</td>
</tr>
<tr>
<td>B</td>
<td>4.0 to 8.0</td>
<td>0 to 50</td>
<td>5 to 7</td>
</tr>
<tr>
<td>C₁</td>
<td>5.0 to 11.0</td>
<td>0 to 35</td>
<td>5 to 11</td>
</tr>
<tr>
<td>C₂</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>D</td>
<td>12.2 to 12.9</td>
<td>30 to 35</td>
<td>11 to 14</td>
</tr>
<tr>
<td>E₁</td>
<td>13.0 to 14.0</td>
<td>0 to 30</td>
<td>N/A</td>
</tr>
<tr>
<td>E₂</td>
<td>14.0 to 16.0</td>
<td>30 to 40</td>
<td>14 to 18</td>
</tr>
<tr>
<td>F₁</td>
<td>16.0 to 19.0</td>
<td>0 to 40</td>
<td>18 to 19</td>
</tr>
<tr>
<td>F₂</td>
<td>19.0 to 22.0</td>
<td>40 to 50</td>
<td>19 to 22</td>
</tr>
</tbody>
</table>

Table 2: Time and Location Critical Path
Table 3: Maximum Time Buffer Caused Free Float

<table>
<thead>
<tr>
<th>Segment</th>
<th>Type</th>
<th>Range</th>
<th>Slope</th>
<th>Free Float</th>
<th>Rate Float</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Early float</td>
<td>0 to 30</td>
<td>-11/300</td>
<td>50/7 - 300/53 = 550/371 ≈ 1.48</td>
<td></td>
</tr>
<tr>
<td>C₂</td>
<td>Late float</td>
<td>35 to 50</td>
<td>11/150</td>
<td>15/1 - 50/7 = 55/7 ≈ 7.86</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Late float</td>
<td>35 to 50</td>
<td>3/50</td>
<td>50/7 - 50/11 = 200/77 ≈ 2.60</td>
<td></td>
</tr>
<tr>
<td>E₂</td>
<td>Late float</td>
<td>0 to 50</td>
<td>1/10</td>
<td>20/4 - 10/3 = 5/3 ≈ 1.67</td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Maximum Location Buffer Caused Free Float

<table>
<thead>
<tr>
<th>Segment</th>
<th>Type</th>
<th>Range</th>
<th>Slope</th>
<th>Free Float</th>
<th>Rate Float</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Early float</td>
<td>0 to 150/7</td>
<td>-11/350</td>
<td>33/49 ≈ 0.67</td>
<td>50/7 - 35/6 = 55/42 ≈ 1.31</td>
</tr>
<tr>
<td>E₁</td>
<td>Early float</td>
<td>0 to 30</td>
<td>-8/75</td>
<td>16/5 = 3.20</td>
<td>30/1 - 50/7 = 160/7 ≈ 22.86</td>
</tr>
<tr>
<td>F₁</td>
<td>Early float</td>
<td>0 to 80/3</td>
<td>-1/12</td>
<td>20/9 ≈ 2.22</td>
<td>40/3 - 120/19 = 400/57 ≈ 7.02</td>
</tr>
<tr>
<td>B</td>
<td>Late float</td>
<td>37.5 to 50</td>
<td>N/A</td>
<td>19/6 ≈ 3.17</td>
<td>50/4 - 3/1 = 9.50</td>
</tr>
<tr>
<td>C₂</td>
<td>Late float</td>
<td>255/7 to 50</td>
<td>N/A</td>
<td>11/10 = 1.10</td>
<td>15/1 - 50/7 = 55/7 ≈ 7.86</td>
</tr>
<tr>
<td>D</td>
<td>Late float</td>
<td>260/7 to 50</td>
<td>N/A</td>
<td>9/21 ≈ 0.43</td>
<td>50/7 - 5/1 = 15/7 ≈ 2.14</td>
</tr>
</tbody>
</table>