Algorithmic Approach for Algebraic Derivation of Time and Distance to Speed during Variable Acceleration

2013-01-2324
Published 09/17/2013

Vincent Goudreault
CAE

doi:10.4271/2013-01-2324

ABSTRACT

The take-off of a departing aircraft is subjected to varying forces, the largest of which are function of the speed of the aircraft itself: the thrust of a jet engine, the aerodynamic drag of the airframe are essentially polynomial function of the airspeed.

First principle field performance determination has relied for the last half century on stepwise integration, a brute force approach that requires properly tuned integration steps, which has the benefit of being fairly reliable but with relatively low computational efficiency.

An alternate, mathematically more formal approach would be to algebraically integrate an accelerative function combining all the forces in presence. While comparatively complex, the derivation of the solution equations permits fast and accurate integration between boundary conditions, which could be orders of magnitude more efficient than stepwise integration, even with relatively high degree of polynomial force functions, while being essentially free of any round-off error which may accumulate at each step of a linearized stepwise integration as the result would instead be derived strictly from the value at the integration limits; the speed advantage remains for most practical models, up to a 6th degree acceleration to speed polynomial function. Moreover, the integrated equations could be used for several airspeed evaluations, leveraging the computational efficiency higher still.

INTRODUCTION

An aircraft on its takeoff acceleration phase is subjected to several forces that ultimately determine how much time and distance will be required before it reaches its rotation speed and can take to the air. Typically, the accelerative force provided by jet engines decreases with increasing airspeed, while the main decelerative force-the aerodynamic drag-goes up with speed. A second retarding force, rolling friction, is also subjected to variations as the increasing aerodynamic lift reduces the weight on wheel, with a further complication brought by operation on contaminated runway as impingement and/or eventual aquaplaning or skidding brings its own nonlinearity to the whole process.

Traditionally, the approach to calculate this variable acceleration is to perform step by step integration [1], where the various forces are considered constant over a small velocity increment. While this method can be considered robust, it nevertheless requires some preliminary work to determine the size of the step that allows proper modeling, and could be a numerical burden even if the approach is optimized by techniques such as pre-calculating the linear fit of the thrust model-as opposed to performing an interpolation between nodes each time-and deterministically deriving the entry into a look-up table-as opposed to searching for the node boundary using an iterative comparison loop.

For many years, the preparation of the performance section of an Aircraft Flight Manual (AFM) required the running of performance expansion programs, which model the aircraft acceleration and operation during for various phase of a takeoff, including engine failure and allowing the determination of the various coefficients or values of interest,

such as V_R and V_2, for a large set of conditions of weight, runway pressure altitude and gradient, wind speed, temperature among others. This very large set, essentially covering the gamut of a First Principle model, is much too large to be provided in detail for inclusion in a computerized AFM, let alone published in series of tables and charts in a paper AFM, and was therefore collapsed using statistical trends. A reference base case, for a restricted set of conditions like a reference weight or absence of wind, would be established, with deviations from this reference being represented by corrections to be applied to the baseline figures. This approach, usually called 'Second Principle', permitted a fairly compact performance model for operators to use, however at the expense of slight reported performance degradation, owing to the corrections of the worst case being universally applied, therefore conservatively so on what would have been the best cases.

Restoring nominal reported performance for the complete operational envelope would call for a computerized AFM, and other operational performance prediction tools such as Manufacturer's Modules conforming to IATA's SCAP standard, to be offered as First Principle, i.e. to be essentially a repackaging of the aircraft manufacturer performance model itself. The advantage to the operators is the elimination of the degraded performance that came from the collapsing of the data. The advantage for the manufacturer, besides a better looking performance envelope from a sales and marketing point of view, may be the elimination of the relatively time consuming and costly statistical collapse process required to prepare a traditional paper AFM, if the production thereof can be avoided. However, this usually shifts the computational burden from the airframer to the operators, as First Principle program computations can be orders of magnitude more complex, and thus slower, than Second Principle performance suite.

While computers have steadily shown increase in their numerical power over the years, perhaps apparently decreasing the desire to invest in computing efficiency, there remains the fact that airlines operate aircraft with a product cycle life that is many times that of a typical computer, and that the codes that support an aircraft that is a couple of decades old may not be compatible with the latest machines, forcing a less than recent computer hardware to remain the platform of choice for running codes like SCAP modules. Additionally, some operators are known to regularly run the performance estimation programs for aircraft in their fleet, for a vast number of possible cases, in order to assess productivity capability of their aircraft and facilitate assignments of specific model to various routes; while running a single takeoff case that takes a few extra seconds due to inherent inefficiency of the code is usually not a problem, running several thousands cases in one session may escalate the time of computation to uncomfortable levels. At the other end of the spectrum, there are recent, small portable machines-tablets and smart phones-where power consumption is one of the chief concern, and which thus do not offer as much number crunching power as full fledged desktop or laptop computers, but which could function as Electronic Flight Bag if fitted with an acceptably efficient code. For such applications, improved efficiency algorithms could be a better alternative that procuring more powerful computer hardware, assuming that those are readily available in the first place.

For the sake of compactness and portability of code, a desire to improve computational effectiveness, and a concern shown by operators regarding the lack of responsiveness in early first principle performance programs, a free-lanced investigation was started to determine if an improved approach to variable acceleration computation could be done, using formal algebraic formulation and integration, and to assess its suitability in the context of deriving time and distance to speed for an aircraft during takeoff. At this point in time, the development caters only to all engine running acceleration, without rotation, but could be expanded to cover other regimes.

FORMAL DERIVATION

Classical Newtonian physics has that the speed and position of an accelerating object are related in the following manner:

$$\frac{dv}{dt} = a \tag{1}$$

$$\frac{ds}{dt} = v \tag{2}$$

In the case of an aircraft, the acceleration is evidently the result of the summation of all the forces acting on it; thrust, drag, rolling friction, and if the runway is not perfectly level, a component of gravity.

Of course,

$$F = m \cdot a \tag{3}$$

and in a typical takeoff, since the overall mass of an aircraft would change only by the quantity of fuel used, which would roughly be in the order of the accuracy of the known takeoff mass, for all practical purposes the mass can be assumed to be constant. However, the total force will vary. While modeling the exact variation of the resulting force is outside of the scope of this paper, it is nevertheless important to characterize how the forces do change, as the degree of variation has an effect on the complexity of the algebraic formulation of the solution. It is known that the aerodynamic drag varies with the square of the aerodynamic speed, for the essentially constant environmental conditions of pressure and temperature during a takeoff ground run. Similarly, the rolling friction, with a coefficient that normally would be constant on an uncontaminated runway, is actually affected by the changing weight on wheels due to the lift, which is again a function of the square of the aerodynamic speed. And

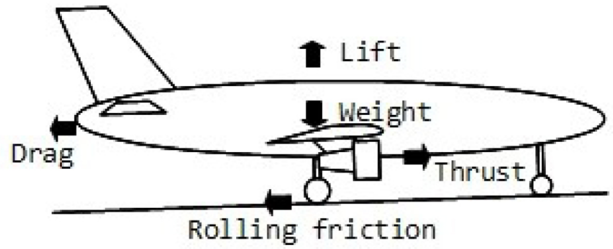

Figure 1. The various forces acting on an accelerating aircraft during takeoff. Most forces are subject to variations as a function of speed, either directly, or indirectly; the rolling friction will vary due to the weight on wheels, which reduces when lift increases as speed goes up. Weight would have an accelerative or decelerative influence along the surface of the strip if the runway is not horizontal

a jet engine thrust, for a given setting, is subjected to variations according to speed. Forces acting on an aircraft are summarized in Figure 1.

While the variation of drag with speed is a fairly well modeled function, the rolling friction is likely to change in a more radical fashion when contamination is present. Most of the models, such as those from ESDU [2], are essentially empirical and obtained from experimental rigs. Nevertheless, the most complicated functions were found to effectively match a 5th order polynomial function of the ground speed. The final varying force, that of the jet engine thrust, would normally fit to a 2nd order function of the aerodynamic speed, all things being equal. However, that would be for a specific and unchanging engine setting, which may not be typical in this day and age. Advanced engine controllers-Full Authority Digital Engine Control (FADEC)-would take advantage of accurate measurements of parameters inside the engine, such as the temperature at the high-pressure turbine, that change with speed and sometime provide a greater margin to the maximum sustainable value to actually increase the engine setting and fuel burn rate when the conditions allow it. The resulting power boost improves field performance, but makes the modeling of a thrust to speed function a bit more of a challenge. Clearly, the actual dynamic setting adjustments performed when speed increases, allowed by the advanced analysis capability of those engine controls, may differ substantially between engine manufacturers, between engine models, perhaps even between temperature conditions for a given engine, making each thrust to speed function very unique. The thrust to speed tables of one popular turbofan engine variant used in several medium size airliners, featuring such FADEC controlled thrust tuning, was selected with the hope that it would be reasonably representative.

Table 1 shows the thrust to Mach number relationship found in a table look-up used for step-by-step integration, for a specific condition of pressure altitude and temperature. Both the ‘standard’ values-those obtained without advanced FADEC tuning-and those where such additional thrust performance are made available are shown. This data can be used to derive polynomial representations, using the least square regression method, providing continuous and integrable functions.

Table 1. Thrust from a mid-size turbofan, as a function of the Mach number, for a specific condition of temperature and pressure. The ‘Standard’ values are the thrust figures obtained in the absence of advanced FADEC tuning, which are shown in the last column.

Mach	Standard Thrust	Thrust Optimized (FADEC tuned)
0.000	13494	13494
0.025	13161	13161
0.050	12846	12846
0.075	12549	12626
0.100	12269	12421
0.125	11996	12229
0.150	11741	12056
0.175	11501	11812
0.200	11270	11582
0.225	11054	11369
0.250	10849	11162
0.275	10653	10969
0.300	10467	10791
0.325	10290	10607
0.350	10123	10433
0.375	9967	10202
0.400	9820	9978

Table 2 shows the coefficients of a 5th degree polynomial fit to the “FADEC tuned” thrust of Table 1. The fit has a 3.23 RMS error, with a peak deviation of 0.27% from the baseline data. It can however be noted that the data was fitted over the whole range, while the thrust above Mach = 0.2 is of little value for a takeoff ground run; fitting only between zero and Mach 0.25 would have returned a better fit, with the largest error being essentially half of what it is over the complete 0 to 0.4 Mach range. It can also be noted that the thrust figure of the standard and the ‘FADEC tuned’ starts deviating at some unspecified point between Mach 0.05 and 0.075. The actual speed at which the regime starts changing cannot be established with any certainty from the available data; determining that exact speed would require investigation using the engine flight deck itself. Such an effort would most likely allow for a finer description of the engine power that

could potentially be fitted with a still smaller error. Nevertheless, it is felt that the accuracy of the current fit is adequate for the present exercise, with an error that is in the order of magnitude of that obtained from a linear interpolation between the given points, which is the approach used in step-by-step integration.

Table 2. Coefficients of a 5th order polynomial least square fit of 'FADEC tuned' thrust data shown in Table 1, for the complete 0 to 0.4 Mach domain.

a_0	13500.75
a_1	-16996.06
a_2	112858.4
a_3	-654903.9
a_4	1712691
a_5	-1632839

Figure 2 shows the data of Table 1, along with the function based on the coefficients of Table 2, in a graphical form. It can be seen that the 5th order polynomial is a quite adequate representation of the 'FADEC tuned' data, over the complete range of values. The standard thrust could be fitted with a 2nd degree polynomial, at an even lower RMS error of 2.19.

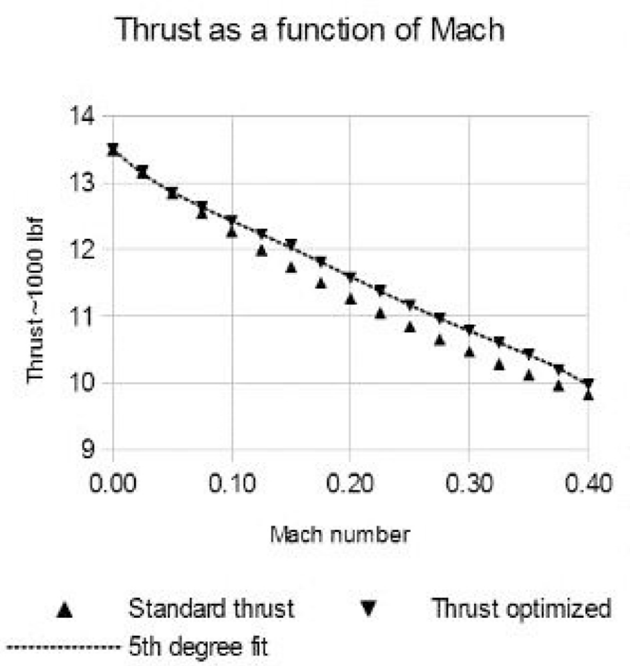

Figure 2. The thrust to Mach data points from Table 1. The curve of the fitted 5th degree polynomial is added.

It should be noted that the actual form of the force model is not critical for the numerical method development; the interest actually resides with the determination of the degree of polynomial needed to properly represent them, as this degree is linked with the level of complexity and ultimately, the resulting computing effort required to derive a solution.

That the forces acting on the accelerating aircraft are ultimately expressible in terms and as a function of ground speed-with the airspeed dependent forces converted to ground speed through a variable change that takes into account wind speed-allows the formulation of a polynomial acceleration function of degree 'n', incorporating the mass and a constant time factor to balance the units:

$$a = f(v) = c_0 + c_1 v + (...) + c_{n-1} v^{n-1} + c_n v^n \tag{4}$$

Inserting Equation (4) in Equation (1) can thus return the time to speed:

$$t = \int \frac{1}{f(v)} dv \tag{5}$$

and the distance traveled can similarly be found with:

$$s = \int \frac{v}{f(v)} dv \tag{6}$$

While the initial formulation is simplistic, the calculation is a bit more complex, owing to the presence of a potentially high degree polynomial in the denominator.

Integration tables [3] do offer antiderivatives for forms having polynomials in the denominator for up to the second degree. For completeness, those are presented here. Note that Equation (9) has three possible formulations, depending on ($4ac-b^2$) being positive, zero, or negative.

$$\int \frac{dx}{ax+b} = \frac{1}{a} \ln|ax+b| + Cte \tag{7}$$

$$\int \frac{x \cdot dx}{ax+b} = \frac{x}{a} - \frac{b}{a^2} \ln|ax+b| + Cte \tag{8}$$

$$\int \frac{dx}{ax^2+bx+c} = \begin{cases} \dfrac{2}{\sqrt{4ac-b^2}} \arctan \dfrac{2ax+b}{\sqrt{4ac-b^2}} + Cte \\ -\dfrac{2}{2ax+b} + Cte \\ \dfrac{1}{\sqrt{b^2-4ac}} \ln\left|\dfrac{2ax+b-\sqrt{b^2-4ac}}{2ax+b+\sqrt{b^2-4ac}}\right| + Cte \end{cases} \tag{9}$$

$$\int \frac{x \cdot dx}{ax^2 + bx + c} = \frac{1}{2a} \ln\left|ax^2 + bx + c\right| - \frac{b}{2a} \int \frac{dx}{ax^2 + bx + c} + Cte \quad (10)$$

The fact that there is no formulation for a polynomial numerator of a degree higher than 2 could be seen as a problem for a numerical method that needs to be, at least in principle, universal. To support modeling of the acceleration of an aircraft using FADEC tuned thrust or operating on a contaminated runway, a worthwhile alternative to step by step integration would need to be able to handle at least to a 5th degree. The core of the method described here is precisely about how higher degree functions are approached.

FACTORIZATION

The first step in handling a higher degree polynomial is to reduce it to several lesser degree ones. There are formal algebraic methods to directly find the roots of 3rd and 4th degree polynomials. For higher degree, an effective method that is however reportedly only partly understood from a theoretical point of view-Laguerre's method-may be a suitable choice. The idea is to reduce the polynomial to a 4th or 3rd degree and to then rely on the algebraic methods for further root finding, as the Laguerre algorithm is an iterative process and not as numerically efficient.

Since algebraic integration formulas are available for degree 2 numerator polynomial, any parabolic sub-polynomial need not be further factored. This has the added benefit of reducing the need to deal with complex roots in the case where said parabolic equations were to lack real roots, which would otherwise impact the numerical efficiency of the complete aircraft performance derivation.

PARTIAL FRACTION

An equation of the form

$$\frac{h(x)}{A(x)} = \frac{h(x)}{f_1(x) \cdot f_2(x) \cdot (\ldots) \cdot f_n(x)} \quad (11)$$

where *h(x)* is a polynomial function of *x* of a lower degree than *A(x)*, while *A(x)* is a polynomial function that can be factored into several lower degree functions *f(x)*; can be converted, through partial fractions, to the form

$$\frac{g_1(x)}{f_1(x)} + \frac{g_2(x)}{f_2(x)} + (\ldots) + \frac{g_n(x)}{f_n(x)} \quad (12)$$

by having [3]

$$\sum g_i \cdot \prod_{i \neq j} f_j = h(x) \quad (13)$$

Regrouping terms as a function of individual degree of *'x'* allows the definition of a series of linear equations which, when solved, would provide the value of the coefficients of each *'g_n'* polynomials. In the case of the integrand of the acceleration function of an aircraft, most of those coefficients would end up canceling each other out, since the function *h(x)* of interest is either 1 (time equation) or *v* (distance equation), thus a first degree at most.

IMPLEMENTATION

The actual implementation of an algebraic evaluation of time and distance to speed for an aircraft during ground acceleration would require the following functional numerical modules:

- polynomial root finder, ideally augmented by formal 3rd and 4th degree polynomial solver, with the associated polynomial division logic needed to factor high degree functions
- partial fraction decomposition manager, itself requiring a linear equation solver [4]
- antiderivative evaluator for the type of functions presented in Equation (7), (8), (9) and (10)

While the above, combined with a suitable accelerative and decelerative force summation protocol, would prove adequate to provide a numerical solution, it may not do so in the desired efficient manner if not properly optimized. And since the aim of the method presented herein is to cut down on the computing cycle, it would fail to provide anything of value above and beyond the traditional step by step integration if it was not able to get the same accurate results in a fraction of the time. As a consequence, a demo suite of source code, partly from numerical libraries and partly from custom developed routines, was put together and mercilessly optimized. The result was that the number of operational modules went up, as functions were sometime split among discrete subprograms. In a few cases, two modules ended up doing mostly the same thing, with one doing supplemental calculations that were of no value for a more basic case; rather than having the more complete module carry useless arguments and having to perform tests to control optional computations, the task of deriving the simpler results was associated to a dedicated function.

In keeping with a desire to maintain compatibility with the IATA SCAP standard-which requires modules to be written in unextended Fortran 77-the experimental Computerized Algebraic Newtonian Velocity Algorithm Suite (CANVAS) was therefore written in that computer language for evaluation. A test environment was custom developed to provide the necessary interface and allow comparison with a 'benchmark' method based on a highly optimized step by step integration featuring:

- pre-computed linear look-up thrust function; as opposed to node points requiring interpolation each time a value is required
- random map entry access, where the index into a table is derived directly from the input value; instead of being

subjected to an iterative search using comparison with node reference entry values

It could be noted that the random map entry access technique, when first implemented in an actual first principle takeoff performance program already featuring pre-computed thrust function, brought a seven-fold improvement in computing speed in an earlier, unfortunately undocumented, deployment. Clearly, a lot of potential optimization had been previously overlooked in that area. Since securing source code from an actual takeoff program at the time CANVAS was developed proved next to impossible, the 'benchmark' program was a comparatively simple implementation, solely concentrating in the stepwise integration of a generic aircraft during ground acceleration. While this approach allowed a fairly direct comparison of the computing efficiency of the new approach, it unfortunately prevented establishing any formal claims of the real performance gain over an actual first principle program; which would require the cooperation and involvement of one of the major airframer to properly establish.

While the step by step method approach is essentially immune to the complexity of the function it integrates, as long as the nodes are sufficiently close together to allow proper representation and that the integration step is roughly in the order the spacing between said nodes so as to properly capture any changing trend in the function, the CANVAS approach is affected by the degree of the polynomial it has to integrate, because of the root finding and the partial fraction process, the later being driven by the operation count of a matrix system solution [4], with a computational load that is proportional to 1/3 N^3, where N is the matrix rank, essentially the degree of the acceleration polynomial. It is therefore expected that, while increasingly complex models are tested, the time to solution for the step by step method would essentially remain stable, while the CANVAS implementation will take progressively more time. Determining at which point the new method stops providing a computational speed advantage was one of the main objective of the test.

RESULTS

For the bench test, a non-specific aircraft model, featuring generic lift and drag equation, mass, rolling friction, was combined with a simplified speed to thrust matrix covering a limited set of temperatures and pressure altitudes. The code front end would have to perform the required interpolation to derive a single airspeed to thrust function. That function was then converted to a series of linear speed to thrust equations, one of which would then be accessed, depending on the current speed bracket for the step by step integration, returning the accelerative force due to engine. For the CANVAS side, the input thrust table was in the form of a list of coefficients of polynomial of degree N (trust as a function of airspeed), which was also interpolated in mock temperature and pressure altitude, to return the applicable function for the integration process. The polynomial was generated to deliberately require the highest degree to be tested, which had the side effect of making some of the behavior rather extreme, but guaranteed that every coefficient was meaningful. Understandingly, the results of the time and distance to speed in those cases was rather atypical, but since the aim of the high order function is to test the computational efficiency of the method, the close to real life aspect was irrelevant.

The test environment ran the step by step integration with a default speed increment of 5 ft/s, from zero ground speed to an arbitrary airspeed of 140 knots. Wind speed correction, to model the required conversion between airspeed and ground speed, was also implemented. Pitch-up and initial climb phase was not included in that model, so the model was limited to a mock acceleration to V_R, for the sole purpose of assessing the computational speed gain of the proposed method.

In terms of accuracy, the CANVAS approach matched the step by step method almost perfectly. Small variations that could occasionally be noted in some cases may be attributed to accumulating round-off errors, which step by step integration is susceptible of showing to varying degree, depending on how the linearized small increments are modeled.

In term of speed, as Table 1 below shows, CANVAS is more than one order of magnitude faster than stepwise integration for the simpler cases, where the acceleration is limited to a second degree function of the airspeed, which bypasses the need for the more advanced data manipulation of root finding and partial fractions. For higher degree, tests indicate that the algebraic method should retain of two fold speed improvement over step by step integration when the function is a 5^{th} degree, and thus calling all the numeric functions described herein. More accurate assessment of the value of the method would require to actually implement it in an actual full-fledged takeoff expansion program, something that would imply the active involvement of an airframer.

Table 3. Computational speed gain; ratio of step-by-step integration time over algebraic integration, including factorization and partial fraction steps.

Degree of polynomial accelerative function	Computational time ratio Stepwise/CANVAS
2^{nd} power of speed, parabolic	21.44
3^{rd} power of speed, cubic	5.73
5^{th} power of speed	2.34
7^{th} power of speed	1.41

Despite the limited scope of the test, there remains that the computational performance gain is likely to be conservative, since the bulk of the computation in CANVAS is in the setting up of the solution, not in the derivation of a specific

answer. While a step by step integration is likely going to have to start anew from zero ground speed to a target airspeed if wind conditions were to change, at the exclusion of any other, the CANVAS method only requires to reuse the comparatively fast antiderivative modules with the same coefficients, but with altered integration boundaries, to return a new solution.

One of the key function of first principle field performance programs is to determine the critical decision speed V_1 applicable to the conditions being tested, and to derive the Balanced Field Length. Typically, the determination of the BFL is performed by iteratively assuming an engine failure speed, comparing the accelerate-go and the accelerate-stop distance, adjusting V_1 until they match. The second part of the calculation, when only one engine is running, is an example of a given accelerative-or decelerative, in the case of the accelerate-stop leg-function that uses a single model with alternate integration boundary conditions. Leveraging already derived functions, reusing them with alternate integration limits, is likely to have a further beneficial impact on the overall computational efficiency and cycle time of first principle performance programs.

SUMMARY/CONCLUSIONS

An algebraic approach to the modeling of variable acceleration of an aircraft in its takeoff roll is a feasible approach, offering advantages in accuracy and cycle time over the traditional step by step integration. While the derivation of the solution is somewhat complex to implement, a software suite makes it automated. Further, iterative determination of a specific value can re-use the already established equations, as changing the integration boundaries will allow to derive a new answer, leveraging the efficiency of the proposed method. In this manner, when a large set of cases need to be computed, all the instances where the variations is limited to a change in wind speed could use the same set of equations.

While the method is less efficient for very complex models-those with a high degree, or discontinuities-it would be possible to break down a complicated function into several successive splines of a lesser degree, or to combine methods: the algebraic approach can be favored for the smoother parts of the acceleration function, and the step-by-step approach for the those showing harder to model discontinuity. Further refinement of the method is possible, and other phases of flight, as of sections of a takeoff run besides the all engine running acceleration, could similarly benefit from enhanced algebraic methods such as the one presented here.

Specifically, it can be envisioned that a full fledged implementation of an algebraic integration could actually model three different functions, or regimes: an all engine running acceleration, a one engine out acceleration, and a deceleration model. The first set of equations would allow, once integrated between zero ground speed and V_R, the direct determination of most of a normal condition take off distance, only needing to be adjusted by the distance covered between liftoff and reaching the mandated 35 ft altitude above ground. The one engine out and the deceleration functions, after being integrated between a still unknown V_1 and, respectively, V_R and zero ground speed-with the adjustments called, once again, by the initial climb, and for additional delays-could be made equal and the system then solved for V_1, thus permitting a direct derivation of the value, as opposed to relying on an iterative process to determine it. The Balanced Field Length would then be the sum of the distance covered from the now derived V_1 and complete stop or 35 ft above ground with one engine out-essentially equal values in the most favorable cases-and the distance taken to reach that V_1 speed from brake release, with all engines running; adjusted to account for the distance and speed variation from the actual engine failure speed and the recognition speed, the determination of which could very well be done using linear methods and models, including a short step-wise integration if need be-again integrated from the all-engines-operating function.

In all cases where it is used, it is important to note that, as long as all accelerative and retarding forces are properly modeled in a smooth, continuously integrable polynomial form, the algebraic integration method is not an approximation; it is, for the domain where it is valid, the most accurate description possible.

Further work to fully investigate, adapt and improve the method would require the active participation of an aircraft manufacturer willing to benchmark their First Principle code, supplying specific performance based information such as the the modeling of the transition between a full takeoff power rating and APR thrust level, for instance. This would determine which specific phases of an aircraft performance calculation could benefit from the advanced numerical method.

In closing, it has to be remembered that an algebraic integration method such as the one presented here does not need to replace every instances of current step by step integration. For instance, it would prove unsuitable for applications involving a pilot in the loop, like in a flight simulator, as the strength of an integral is to derive the final answer at the boundary condition, without having to compute all the intermediate steps. Properly used, however, it can supplement existing methods, and would allow improved accuracy and speed of calculation, being an addition to the currently available set of tools.

REFERENCES

1. Boeing's Jet Transport Performance Methods, 7th edition, May 1989

2. Balkwill, K.J., ESDU International plc, Development of a comprehensive method for modelling performance of aircraft tyres rolling or braking on dry and precipitation-contaminated runways, Transport Canada TP 14289E, May 2003

3. Bayer, William H.; Standard Mathematical Tables, CRC Press, 1976, ISBN 0-87819-623-4

4. Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P.; Numerical Recipes, Second edition, Cambridge University Press, 1992, ISBN 0-521-43064

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. This process requires a minimum of three (3) reviews by industry experts.

ISSN 0148-7191

SAE Customer Service:
Tel: 877-606-7323 (inside USA and Canada)
Tel: 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE Web Address: http://www.sae.org
Printed in USA