http://en.wikipedia.org/w/index.php?title=T-integration&oldid=402346593

T-Integration is a numerical integration technique developed by Jon Michael Smith to facilitate digital computer controlling and simulating aircraft, space craft and similar computer controlled dynamic systems. Short for "Tunable Numerical Integration", it is a fixed-step iteration formula whose integrand can be adjusted in phase and amplitude. T-Integrators all have *phase* and *gain* adjustable parameters that are similar to phase and gain adjustable parameters in modern aircraft autopilots. The simplest version of the T-Integrator algorithm is as follows:

Let $f(x)$ denote the integrand and P and G the phase and gain parameters. Furthermore, the left-hand side limit of the integral is denoted by x_0 and Δx is the step size. T-integration is defined by the following recursive formula:

$$F_n = F_{n-1} + G\,\Delta x\,(P\,f_n + (1-P)\,f_{n-1}), \text{ and } F_0 = 0.$$

Here f_n stands for $f(x_n)$. The quantity F_n approximates

$$\int_{x_0}^{x_0+n\Delta x} f(x)\,\mathrm{d}x.$$

If $G = 1$, then the method reduces to the following well known numerical integration techniques for the given values of P:

- $P = 0$: the left-hand Rectangle rule known as Eulers method
- $P = 1/2$: the Trapezoid rule,
- $P = 1$: the right-hand Rectangle rule,
- $P = 3/2$: the Adams-Bashfourth corrector rule, (see Linear multistep method)
- P = Etc.

If G and/or P are other real numbers, then a set of new first order integrators is produced.

What is amazing about the T-Integrator is that there are a unique pair of P and G that is ideal for exact simulation of a given linear system. Usually, they are not any of the classical numerical integrators. Since there are an infinity of P and G values (not just the integer values shown above) the T-Integration formula contains an infinity of integrators for use in control and simulation systems. This flexibility enables the selection of a phase-gain pair that allows simulator root locus to exactly match the root locus of a dynamic linear system. Even more surprising is that a small set of these first order integrators can match the Jacobian of a nonlinear system being simulated.

G and P can also be selected empirically by matching the numerically integrated trajectory with a known real world check case. This is particularly useful when simulating aircraft motion for various aircraft configurations. For example, G and P can be selected to match the real motion of the aircraft with the landing gear up, gear down, flaps up, flaps down, high Mach, low Mach, right engine out, left engine out and combination's of these and other aircraft configurations. In these applications, *G* and *P* are changed depending on the landing gear handle position, the flap handle position, the throttle position etc. This is particularly important when getting a simulator certified by the FAA (or other certification organization) for flight training purposes. You simply tune the simulator to satisfy the certifying organizations requirements for pilot flight handling evaluation and approval.

What makes T-Integration so different for classical numerical integration is that the foundation for the derivation of T-Integration is information theory while the foundation for the derivation of classical numerical integrators is approximation theory. The two theories were developed at different times. Information theory, being the more modern of the two, is more suited to modern digital simulation, controls and information sciences applications. Phase and Gain controls are commonplace in information and control systems applications; not so with classical numerical integrators.

T-Integration can be tuned to the problem it is being used to solve. For open-loop problems, setting the gain to 1 and varying the phase produces ALL classical first order numerical integrators and an infinity new integrators heretofore unknown. For closed loop applications the T-Integrator produces an infinity of non-classical integrators that produce exact numerical integration of linear systems and near exact integration of nonlinear systems.

For information systems applications (computer, control and communication and simulation) this first order T-Integrator out performed numerical integrators based on classical approximation theory. Simulating aircraft motion for various aircraft configurations (gear up, gear down, flaps up, flaps down, engine out, stab-aug on, stab-aug off etc.) and dynamic conditions (high Mach, low Mach, take-off, landing etc.) becomes a matter of tuning the T-Integrator (finding the *G* and *P*) to the flight condition being simulated. In this sense the T-Integrator adapts to the problem it is trying to solve.

For new users of T-Integration, rectangular integration ($G = 1$ and $P = 0$) introduces a half sample period of delay by approximating the integrand with a sequence of rectangles (a sort of stair-step approximation). This integrator is used by Fowler in his root locus matching method of 1969. Trapezoidal integration ($G = 1$ and $P = 1/2$) solves this problem by approximating the integrand with a sequence of trapezoids (a sort of a mountain range approximation). Trapezoidal integration is pretty effective for real-time open-loop numerical integration of an integrand. Not so with closed-loop applications. In modern times, Matlab includes many classical numerical integrators. Using Matlab without insight into the phase and gain of the integrator is not advised particularly for flight control system design. The situation persists even now, in 2010. This is of the first places I look for when serving as a control system review consultant because even slight delays in high performance high dimension control system applications can cause flight control instabilities at critical events such as automatic landing, orbital rendezvous, entry flight control and flight at the edge of the vehicle performance margin (as in abort situations).

Closed-loop applications of either of these open-loop integrators introduce an additional sample period of delay. The integrand must use the past values of the integral to solve the differential equations for the sequence of integrand values at each step in the integration/control process. In block diagram form, this is shown as a $1/z$ in the feedback path of the integration/control process.

To compensate for this feedback delay, the T-Integrator should introduce a full sample period of lead into the process ($G = 1$ and $P = 3/2$). This leads to the Adams' Corrector integrator. This integrator was much studied for real time simulators by MIT and IBM in 1972.

When applied to solving real or simulated systems of DEQs, such as solving (using) the quaternion equations of rotational motion, numerical integration should proceed serially not in parallel. Parallel integration introduces yet another sample period of delay. Parallel integration delay occurs because the fourth quaternion must wait three sample periods before the effect of the first quaternion is processed. This is typical of matrix solutions of DEQs as commonly used in MatLab.

A much better way is to compute each quaternion rate and immediately integrate the rate and use it to compute the next quaternion rate, then immediately integrate that rate and use it to compute the next rate etc. This way, the first quaternion immediately shows up in the calculation of the fourth quaternion. Serial use of numerical integration does not introduce delays produced by parallel numerical integration of a system of DEQs.