

Emmet FLETCHER¹

¹ Analytical Graphics, Inc.
Plaza de la Encina 5, local 5
Tres Cantos
28760 Madrid
Spain
efletcher@stk.com

CONTINGENCY PLANNING AND MISSION PERFORMANCE IMPACTS: A NOVEL APPROACH TO LAUNCH ERROR SIMULATION, CHARACTERISATION AND CORRECTION

Launch uncertainties can cause a significant impact on the scope and lifetime of any satellite mission. The characterisation and impact of these launch inaccuracies is important to ensure that the mission has been designed to be successful through the entire regime of possible errors. Using traditional methods usually entail a simplification of the problem due to constraints imposed to meet schedule deadlines and the difficulty of creating mathematical models that will accurately cover the entire range of possible errors. Through the use of Commercial Off The Shelf (COTS) software, coupled with algorithms tailored to meet the system to be studied can either reduce the time required to completely characterise the boundaries of recoverable errors, or enable a broader range of possible scenarios to be studied.

1. System Tools

In order to reduce the time and complexity required to develop an adequate simulation system, the use of COTS software can have a large impact on the final results. In this study, we used Satellite Tool Kit (STK[®]) from Analytical Graphics, Inc. as the core of the system. In particular, we employed the STK/Astrogator model as the computational engine to drive the initial parameters to a final conclusion which could then be analysed.

1.1. Satellite Tool Kit (STK)

Satellite Tool Kit (STK) is a flexible Commercial Off The Shelf (COTS) simulation and analytical package that is widely used throughout the aerospace industry in order to examine dynamic events through the use of a geometrical engine. The core aerospace component of STK includes various standard orbit propagators, such as the Runge-Kutta-Fehlberg integrator with eight-order error control for advanced orbit analysis (taking into account third-body gravitational effects, drag, solar-radiation pressure etc).

The system has been designed for a rapid analysis and development of “scenarios” which enable many objects to be examined throughout distinct temporal regions with an easy-to-use graphical interface.

1.2. STK/Connect

The development of scenarios, as described in section 1.1 are ideal if one is dealing with a fixed set of initial parameters. If, however we wish to examine the delta between differing initial characteristics, then we would face a laborious task of rerunning the same scenario through different iterations while manually changing the initial values each time.

In order to reduce the necessity of this task, a messaging service has been developed to allow for the control of STK through simple text strings. With the supplied set of integration packages, STK/Connect allows commands to control STK through a TCP/IP interface to be rapidly integrated into common languages and from there allow more advanced scenarios to be developed with changing initial values. STK/Connect commands can be embedded into almost any common programming language such as Perl, C++, Java etc.

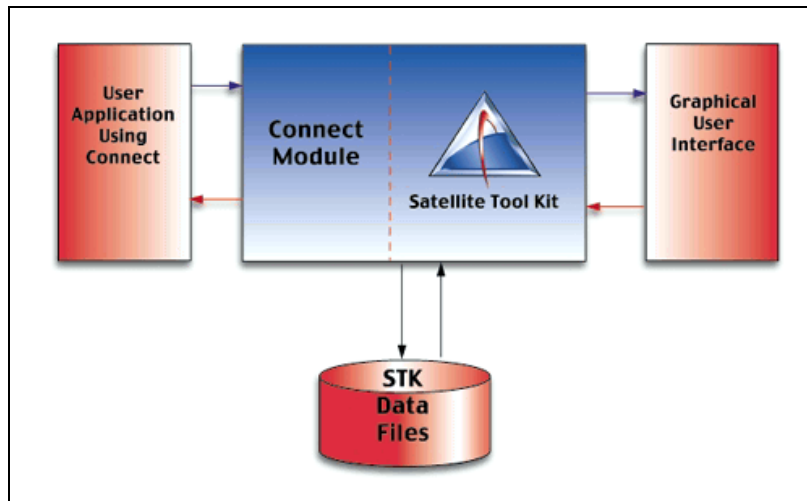


Figure 1: STK/Connect Flow

As can be seen in Figure 1, the STK/Connect module acts as an interpreter to the text TCP/IP messages and allows control over the whole of the STK suite that is installed.

Even though the Graphical User Interface (GUI) is still available, it can be completely discarded by the user of the software if wished.

1.3. STK/Matlab Interface

Given the widespread use of Matlab from The Mathworks Inc., as well as the inherent advantages of using such a package with its powerful mathematical functions and display characteristics, an interface between STK and Matlab, called the STK/Matlab Interface was used to send STK/Connect commands to STK. In this way, an iterative script could be created and tested in the minimum time possible.

The results generated with STK could then be collated and displayed within Matlab and further manipulation of the raw material was then possible.

1.4. STK/Astrogator

STK/Astrogator is the core of the launch error simulator and corrector.

1.4.1. History

Astrogator has a long and proven history that began with the development of the Swingby¹²³ program in 1989 at Computer Sciences Corporation for NASA Goddard Space Flight Center (GSFC). In January 1994, Swingby was used operationally for the Clementine mission⁴⁵. Later that same year, CSC worked with AGI to enhance this program and commercially sell it as a product called Navigator.

Swingby continued to be used operationally for the Wind launch in 1994⁶⁷⁸ and the SOHO launch in 1995⁹. In early 1997 at the request of GSFC, AGI began the design and development of Astrogator, and a prototype was delivered in late 1997. An immediate success story in Astrogator's history occurred in December 1997, when navigator was used to plan the lunar gravity which rescued Hughes' AsiaSat3 from a useless orbit¹⁰. In March 1998, GSFC began beta testing Astrogator and in January 1999 they began using it for MAP mission analysis¹¹.

Since then, STK/Astrogator has been utilised in numerous civil and commercial space mission for preparation, analysis and anomaly resolution.

1.4.2. Functional Analysis¹²

The role of STK/Astrogator is to propagate orbits using custom designed propagators, but what interests us to solve our particular problem is the ability to solve complex targeting problems, given the initial conditions and then the desired final values (with accompanying accuracy requirement).

The method that is used within Astrogator is to take a specific set of orbital goals and then take the initial conditions and the intermediate values that we would like to perturb in order to meet the goals.

The mechanism used in Astrogator is the differential corrector. This has proved to be robust in practice and able to handle a wide range of initial problems and end requirements.

The differential corrector within Astrogator works by expressing the target problem in terms of a pseudo-Taylor series expansion of the goals \mathbf{G} expressed as a function of the variables \mathbf{v} . The problem can be represented by the expression

$$G_1(\mathbf{v}_1) = \bar{G}(\bar{\mathbf{r}}, \bar{\mathbf{v}}, t) = G_0 + \left[\sum_{n=1}^{\infty} \frac{1}{n!} \frac{d^n G}{d\mathbf{v}^n} \Big|_{\mathbf{v}=\mathbf{v}_0} (\mathbf{v}_1 - \mathbf{v}_0)^n \right] \quad (1)$$

where the terms in brackets are the Taylor series expansion of the equation for the goals $G(\mathbf{v})$ expanded in terms of variables \mathbf{v} . Since the problem that is usually stated involved a possibly endless combination of factors (gravitating bodies, drag, solar-radiation pressure etc), there is no closed-form expression for G , and the Taylor series expansion cannot be performed analytically. In addition, even if a formula for G were available, the series expansion is theoretically infinite and, as such, impossible to solve (except in special cases).

An alternative approach is to truncate the Taylor series expansion at a convenient point and evaluate the resulting equation numerically. When the series is truncated at the first derivative, the following equation remains:

$$G_1(\mathbf{v}_1) \approx G_0 + \frac{dG}{d\mathbf{v}}(\mathbf{v}_1 - \mathbf{v}_0) \quad (2)$$

This equation can be rearranged to give:

$$G_1' = \frac{dG}{d\mathbf{v}} \approx \frac{G_1 - G_0}{\mathbf{v}_1 - \mathbf{v}_0} \rightarrow \mathbf{v}_1 \approx \mathbf{v}_0 + \frac{G_T - G_0}{G_1'} = \mathbf{v}_0 + (G_T - G_0)[G_1']^{-1} \quad (3)$$

Thus, we can find the value of \mathbf{v}_1 that approximates the value of the variables needed to meet our goal G_T if we know the initial value for the variable \mathbf{v}_0 and can find an approximation to the derivative G_1' . To generate this approximation, we need only to perturb each variable by some small amount $\delta\mathbf{v}$ and measure the resulting change in each goal G . Then, by the fundamental theorem of calculus,

$$G_1' = \frac{dG}{d\mathbf{v}} = \lim_{\delta\mathbf{v} \rightarrow 0} \frac{G(\mathbf{v} + \delta\mathbf{v}) - G(\mathbf{v})}{\delta\mathbf{v}} \quad (4)$$

The limit cannot be taken to zero numerically because of the finite precision of standard computing equipment. When one targets a goal in Astrogator, one can enter the value of the perturbation (\mathbf{v} in this case) in the perturbation entry parameters for the appropriate variable in the targeter, or one can let Astrogator use a default value. The targeter then proceeds to run a nominal trajectory to calculate the starting value of the parameter, describing the goal initially set.

Astrogator then adds the perturbation to the variable and performs the targeting sequence. This information is used in equation (4) to calculate an approximate local value for the first derivative of the equation describing the initial goal. This value is then used in equation (3) to determine a new estimate for the variable used in the targeter. This process is repeated until the achieved goal falls within the tolerance for the goal (as set by the user).

Several complications arise when the problem being solved contains more than one variable and goal. One may note that equation (3) contains the inverse of the derivative operator. When the problem being solved contains two variables (x and y for example) and two goals (A and B), then there are two equations that describe the solution:

$$\begin{aligned} f(x + \Delta x, y + \Delta y) &= A \\ g(x + \Delta x, y + \Delta y) &= B \end{aligned} \quad (5)$$

Here $f(x,y)$ is the equation describing the goal that has a targeted value of A, and $g(x,y)$ is the equation describing the goal with a targeted value of B. Each of these equations is evaluated with the perturbation of each variable separately. The resulting matrix operation has the following form:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \approx \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix}^{-1} \begin{bmatrix} A - f(x,y) \\ B - g(x,y) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + [\tilde{M}']^{-1} \begin{bmatrix} A - f(x,y) \\ B - g(x,y) \end{bmatrix} \quad (6)$$

The matrix $[\tilde{M}']^{-1}$ of the partial derivatives is called the 'sensitivity matrix'. This matrix must be invertible for the differential corrector to perform correctly. Amongst other things, this means that there must be no degeneracy in the matrix. Astrogator uses a singular value decomposition algorithm to invert this matrix.

2. Problem Definition

The problem that we want to solve is the Δv costs of differing launch errors. In order to do this, we first create within Astrogator a Mission Command Sequence (MCS) that accurately describes the initial state vector after final stage burnout. A typical MCS is shown in figure 2:

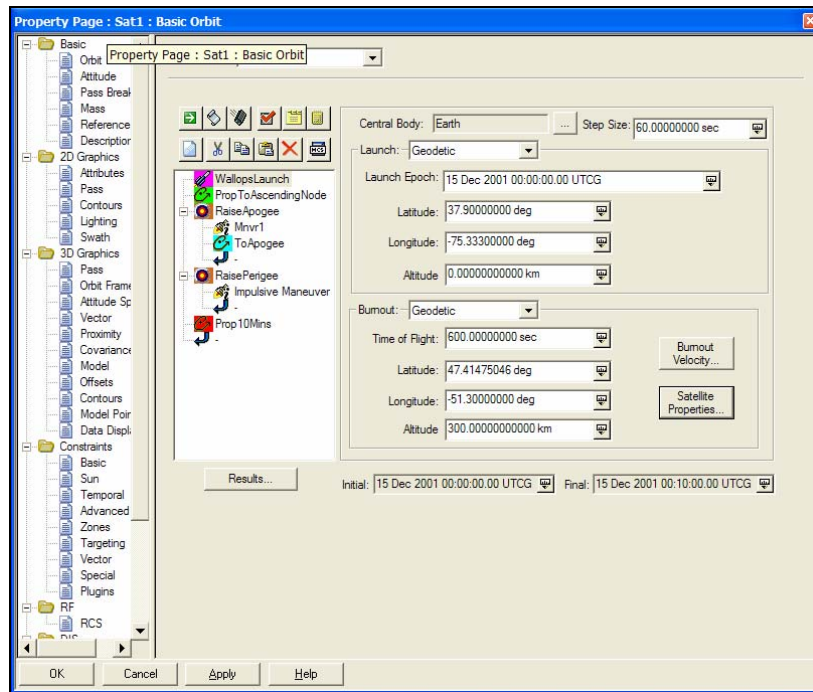


Figure 2: MCS design window

A close-up view of the actual MCS structure is shown in Figure 3:

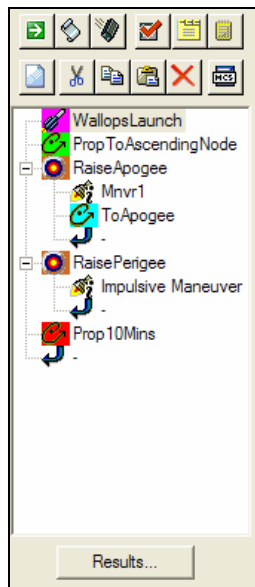


Figure 3: Close-up view of MCS structure

The structure in figure 3 shows the initial state being given by a ballistic launch to initial orbit and then allowing the software to propagate the orbit until the first ascending node passing.

At that point the software computes the Δv to manoeuvre the satellite to perform an apogee raising manoeuvre using the differential corrector described previously. The initial state is provided by the ascending node passing state vector and the final goal G is given by the apogee altitude required. The apogee achieved is measured at the apogee location, since propagation to the apogee point is included within the target sequence.

When the requirements given for a successful are achieved, then a second target sequence is initiated, this time to raise the perigee for the orbit. Yet again, the Δv required to perform this operation is calculated.

2.1. Launch Error Simulation

In order to simulate the errors, we must create a source of errors. As described previously, we can inject external values into STK (and therefore, by definition, to STK/Astrogator) through the use of STK/Connect. We can therefore take the nominal initial state of the launch, as provided in the first entry to the Astrogator MCS and add a small delta to the limits of the launch system manufacturer's specifications.

This has been performed in two ways. Initially a model was created using the *Perl* scripting language, which is an open source language used extensively in the manipulation and processing of text strings. Since the commands within STK/Connect are basically in the form of text messages, the language lends itself to this kind of manipulation. The drawback to this kind of language is that any graphics manipulation would have to rely on extensive post-processing through the use of tools such as Microsoft Excel or similar.

An alternative method, and one that was used in the later stage of the study, involved the use of Matlab from The Mathworks, Inc. Using a COTS interface to STK, STK/Matlab Interface, the same script was ported to a Matlab M-file. This can also manipulate text data, but has the advantage of allowing instantaneous transfers of this data to an easy to read graphical format. The script that we used was created by Space Exploration Engineering, Inc.

A Monte Carlo simulation was created within Matlab to generate a range of delta values to the initial nominal launch trajectory. This was then used to create different initial values to the Astrogator MCS and commands were incorporated to loop the MCS through the entire range.

Using this source of errors, we can now do two things: launch error characterisation and launch error correction.

2.2. Launch Error Characterisation

If we wish to examine the effects of a changing initial state (the change in the launch vehicle's delivery point) then we can run the MCS with the targeter off. In this mode, the entire sequence will run, but taking the nominal mission profile in each case.

This has the result that the Δv supplied by the engines is the same even though the initial conditions have changed. In this way, we can examine the impact on the final orbit of changes of initial conditions.

Even though this is not using the more advanced capabilities of STK/Astrogator, it is a quick and convenient way of characterising the entire mission profile (which could extend to the complete mission lifetime if one wished) to evaluate if the mission objectives can still be met. Given that the STK suite allows for sensor and communication system simulation, it is possible to gauge the impact of varying initial launch insertion values on total system performance.

In our simulation, it we only requested the change in the orbital parameters and the Δv required if a correction is possible. It is also feasible to request the system to display a dispersion in total Earth coverage (for an Earth Resources mission) or the change in the link budget (for a communications infrastructure).

2.3. Launch Error Correction

If our mission has the possibility to make changes in the use of the orbit raising engines, then we can use Astrogator to calculate the change in Δv required to comply with the original values of the final orbit.

In order to do this, we must enable the differential corrector in the targeting sequences so that the final values for both the apogee and perigee raising manoeuvres can be complied with. In our investigation,

only the final values of the Δv required are calculated. It is a very simple procedure to extend this simulation to account for the impact on the total mission lifetime that is changed due to unforeseen propellant use during the initial circularisation manoeuvres.

3. Results

For evaluation purposes, the system was initialised as above, with a nominal orbit of 2000km altitude, as shown in the ‘nominal’ column in table (1). The initial insertion values were then varied using a simple Monte Carlo script.

Using a large collection of values (at 500 iterations) gave the following values:

Table 1: Results after 500 runs

Value	Nominal	Average	Std Deviation	Units
Semi-major axis	8376.0684	8375.8152	0.16621	km
Eccentricity	0.000443	0.00057792	1.2711e-005	
Inclination	49.9941	49.9979	0.0017975	deg
RAAN	329.149	329.3806	0.007392	deg
ω:	316.7768	307.06	1.6701	deg
True Anomaly	251.7699	246.3732	2.6485	deg
Ascending Node Latitude	208.5467	193.4332	16.4527	deg
Δv magnitude	0.82712	0.8271	0.0001411	km/sec

Using Matlab, this information can then be represented in chart form:

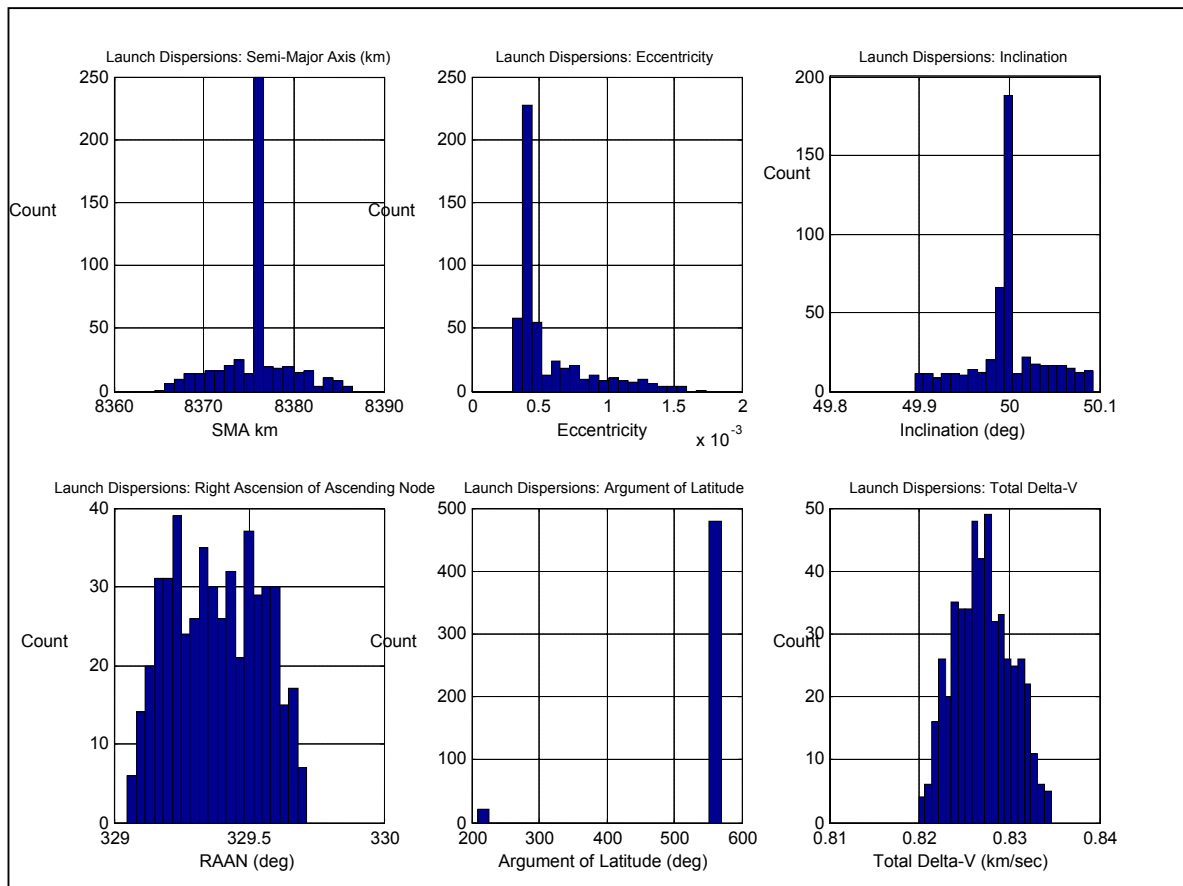


Figure 4: Chart results after 500 iterations

Comparisons can then be made, as well as a risk assessment to assist in the design of the trade-off values for the final mission outline.

4. Conclusions

From this study, we found that the combination of STK/Astrogator as well as input from an external source gave results within a few minutes that would otherwise have taken a disproportionate amount of manpower to achieve.

Due to the inherent simplicity of the system, it could lend itself to such studies as anomaly resolution after a system has been launched, but where unforeseen errors in the launch system have created an off-nominal orbit.

Indeed, the simulation described in this paper could be expanded to an entire range of manoeuvres, such as geostationary orbital insertion, and the requirements to reach a specified spot with the minimum amount of fuel. Studies have been done within AGI to model the entire launch campaign and the required calculations to operational orbit. Given the flexibility of the Astrogator targeter, this could be extended to perform extensive studies with a numerous array of operational constraints (such as sun angle restrictions).

¹ J.P. Carrico, H.L. Hooper, L. Roszman, C. Gramling, "Rapid Design of Gravity Assist Trajectories," Proceedings of the ESA Symposium on Spacecraft Flight Dynamics, Darmstadt, Germany, September/October 1991

² J.P. Carrico, C. Schiff, L. Roszman, H.L. Hooper, D. Folta, K.V. Richon, "An Interactive Tool for Design and Support of Lunar, Gravity Assist, and Libration Point Trajectories," AIAA 93-1126, AIAA/AHS/ASEE Aerospace Design Conference, Irvine, California, USA, February 1993

³ J.P. Carrico, D. Conway, D. Ginn, D. Folta, K.V. Richon, "Operational Use of Swingby-an Interactive Trajectory Design and Maneuver Planning Tool-for Missions to the Moon and Beyond," AAS 95-323, AAS/AIAA Astrodynamics Specialist Conference, Halifax, Nova Scotia, Canada, August 1995

⁴ J. Carrico et al., "Maneuver Planning and Results for Clementine (the Deep Space Program Science Experiment)," AAS 95-129, AAS/AIAA Spaceflight Mechanics Meeting, Albuquerque, New Mexico, USA, February 1995

⁵ D.W. Dunham, R.W. Farquhar, "Trajectory Design for a Lunar Mapping and Near-Earth-Asteroid Flyby Mission," AAS 93-145, *Spaceflight Mechanics 1993*, Vol. 82, Advances in the Astronautical Sciences, ed. R. Melton, pp. 605-624

⁶ P. Sharer, H. Franz, D. Folta, "WIND Trajectory Design and Control," Paper No. MS95/032, CNES International Symposium on Space Dynamics, Toulouse, France, June 1995

⁷ H. Franz, "WIND Nominal Mission Performance and Extended Mission Design," AIAA 98-4467, August 1998

⁸ H. Franz, "WIND Lunar Backflip and Distant Prograde Orbit Implementation," AAS 01-173, AAS/AIAA Spaceflight Mechanics Meeting, Santa Barbara, California, USA, February 2001

⁹ D.W. Dunham, et al., "Transfer Trajectory Design for the SOHO Libration-Point Mission," IAF-92-0066, 43rd Congress of the International Astronautical Federation, Washington, D.C., USA, August/September, 1992

¹⁰ J. Salvatore, C. Ocampo, "Mission Design and Orbit Operations for the First Lunar Flyby Rescue Mission," IAF-99-A.2.01, 50th International Astronautical Congress, Amsterdam, The Netherlands, October 1999

¹¹ M. Mesarch, S. Andrews "The Maneuver Planning Process for the Microwave Anisotropy Probe (MAP)

Mission,” AIAA 2002-4427, AIAA/AAS Astrodynamics Specialist Conference, Monterey, CA, USA

¹² Analytical Graphics, Inc. STK/Astrogator background information:
http://www.stk.com/resources/download/astrogator/astrogator_resources.cfm