Autonomous Orbit Propagation for GPS Equipped Cubesats

Gim J. Der
Der Astrodynamics

Andrew E. Kalman
Pumpkin, Inc.

ABSTRACT

This paper describes the computational performance and implementation of the analytic Vinti algorithm for autonomous orbit propagation on GPS equipped Cubesats. Since the Vinti algorithm propagates directly with the GPS state vector (position and velocity vectors) as input, orbit propagation is straightforward, and unlike SGP4, conversion to TLE mean orbital elements is not required. Also periodic uplinks of the NORAD TLEs, which consume a relatively large amount of Cubesat energy and costly ground operations, are no longer needed. If the time between two GPS locks is 48 hours or less, then Vinti propagation is on average at least as efficient (accurate and fast) if not better than any existing analytic orbit propagator for space objects in any orbit regime. The Goddard trajectory program (GTDS) indicates that the analytic Vinti algorithm has no singularities, unlike other propagators. With the 2-Body and perturbed Kepler equations solved correctly, the Vinti computed state vector errors between GPS locks are less than two kilometers per day on average for Near-Earth objects. The implementation of the analytic Vinti Orbit Propagator (Vinti7 OP) into the Flight Software of the JPL-Stanford LMRST-Sat Cubesat by Pumpkin shows that very little energy (20 to 200 times less than a GPS fix) is consumed for the Vinti7 OP to compute an accurate state vector between GPS locks. Since the Vinti7 OP is accurate, fast, robust, and applicable to all orbit regimes, not to mention easy to use and implement, and consumes very little energy, it is hard to find a more efficient autonomous orbit propagator for GPS equipped Cubesats. In addition, ground processing of orbit determination, tracking and cataloging for cooperative Cubesats is greatly simplified.

1. INTRODUCTION

The USAF has two space catalogs, TLE and SP for respectively analytic SGP4 and numerically integrated (NI) orbit propagation. In general, for orbit propagation, the analytic TLE/SGP4 (catalog/propagator) is faster, but less accurate than the time consuming numerical SP/NI. With these combinations, the USAF has "standardized" orbit propagation for its cataloging procedures. In other words, the powerful analytic Vinti propagator [1] has played no role for decades and therefore has been gradually forgotten. Orbit propagation for small satellites with a GPS state vector as input, however, does not have to use the same propagator that is used for cataloging. For commercial applications, the best propagator is chosen by virtue of better accuracy, speed, robustness, ease of use and implementation with far fewer lines of code, and very little energy consumption (an important requirement for Cubesats).

Traditionally, analytic orbit propagators that include perturbations have been based on the two theories, Brouwer/Kozai (SGP4) and Vinti. Since the USAF emphasized the development and implementation of analytic TLE/SGP4 for over 50 years, many of the excellent textbooks of astrodynamics [2 to 14] except a few [15, 16] have failed to mention the Vinti algorithms or the current Vinti7 OP. The input and output of the Vinti7 OP are the ECI state vectors, which are also used for both numerical integration and GPS output. Unfortunately, important astrodynamics papers [17, 18] have either ignored the existence of the Vinti algorithms or misinformed inexperienced researchers that SGP4 is the only accurate analytic orbit propagator, even for GPS equipped Cubesats [19].

The Brouwer and Vinti theories were developed independently by professors Dirk Brouwer (Yale) and John Vinti (MIT) in 1959 [20, 21]. NASA praised the robustness of the Vinti theory in its trajectory determination bible GTDS [22]. NSA wanted singularity free orbit propagation of Molniya satellites, and initiated code development of the Vinti algorithm (Vinti5) in the early 1970s [23]. War fighters enjoyed the applicability of Vinti trajectory propagation for both missiles and satellites [24, 25] for decades. A version of the Vinti algorithm (vinti2 [1]) was used for missile targeting. Despite the decades long focus on SGP4, continuous interest in the Vinti theory and algorithm has been maintained by premier astrodynamics researchers [see references in 1] in the U.S., Russia, EU, China, Korea, India, among others. Recent research papers on Vinti theory supported by USAF were published by Wiesel [26] in 2014, and Biria and Russell [27] in 2015.

The Vinti theory, as with that of Brouwer/Kozai, begins with the Hamilton Jacobi equations of motion (the ticket to quantum mechanics), and therefore many students and engineers may not have the mathematical skills to acquire a good understanding. This difficulty is resolved by the 1998 Vinti AIAA book [1] that includes Fortran and C source code and numerical examples of the Vinti algorithm (Vinti6). Robustness was improved in Vinti7.

2. THEORETICAL SINGULARITIES

The problems of tracking satellites and orbit propagation at theoretical singularities of the zero and critical inclinations still present difficulties for Brouwer type algorithms (SGP and SGP4). In a letter dated December 23, 1959, Brouwer wrote to Vinti to express his displeasure and criticism that Vinti asserted to have solved the critical inclination singularity problem in the National Bureau of Standards Report, STR-2434. This report and the letters of their dialogue, which were obtained from the family of Vinti, are presented in Brouwer_vs_Vinti [28], http://derastrodynamics.com/docs/brouwer_vs_vinti_v1.pdf.

An algorithm with theoretical singularities, which may produce unexpected or erroneous state vectors without notice, is undesirable for autonomous orbit propagation. The use of mean orbital elements (mean a, mean e, mean i, . .) in the equations of Brouwer type algorithms provides a picture of the size, shape and orientation of an orbit, but at the expense of theoretical singularities. The innovative Vinti7 OP computes with Cartesian ECI state vectors as input and output, and theoretical singularities do not exist in Cartesian space.

3. KEPLERIAN CONVERGENCE

For hundreds of years, the method of solution for a 2-Body or perturbed Kepler equation is another source of orbit propagation failure. The classical theory ends with solving for the solution of one unknown, the eccentric anomaly E, in the Kepler equation:

$$F(E) = E - e\sin E - M = 0 \tag{1}$$

where the mean anomaly M and the eccentricity e can be computed from the given state vector ($\mathbf{r}(t_1)$, $\mathbf{v}(t_1)$) at the given time t_1. The modern Kepler equation formulated by the Universal variable, x, can be expressed as:

$$F(x) = r_1 U_1 + \sigma_1 U_2 + U_3 - \sqrt{\mu}\,(t_2 - t_1) \tag{2}$$

where the derivation of equations (1) and (2) can be found in many papers and text books [1 to 16]. After E or x is found, the required state vector ($\mathbf{r}(t_2)$ and $\mathbf{v}(t_2)$) at the given time t_2 can be evaluated. Equations (1) and (2) are analytic but not in closed form; iterative methods must be used. Orbit propagation is successful, when the Kepler equation is iterated to converge to the correct state vector. This iterative procedure may be defined as Keplerian Convergence.

In theory, iterating a simple 2-Body or Perturbed Kepler equation of one unknown to convergence appears to be straightforward. However, to guarantee the correct solution can be difficult. Most textbooks and papers derived a 2-Body Kepler equation, and then claim without verification the solution can be found by an iterative algorithm, such as Newton, Halley, Laguerre, among others. In practice, it is easier said than done.

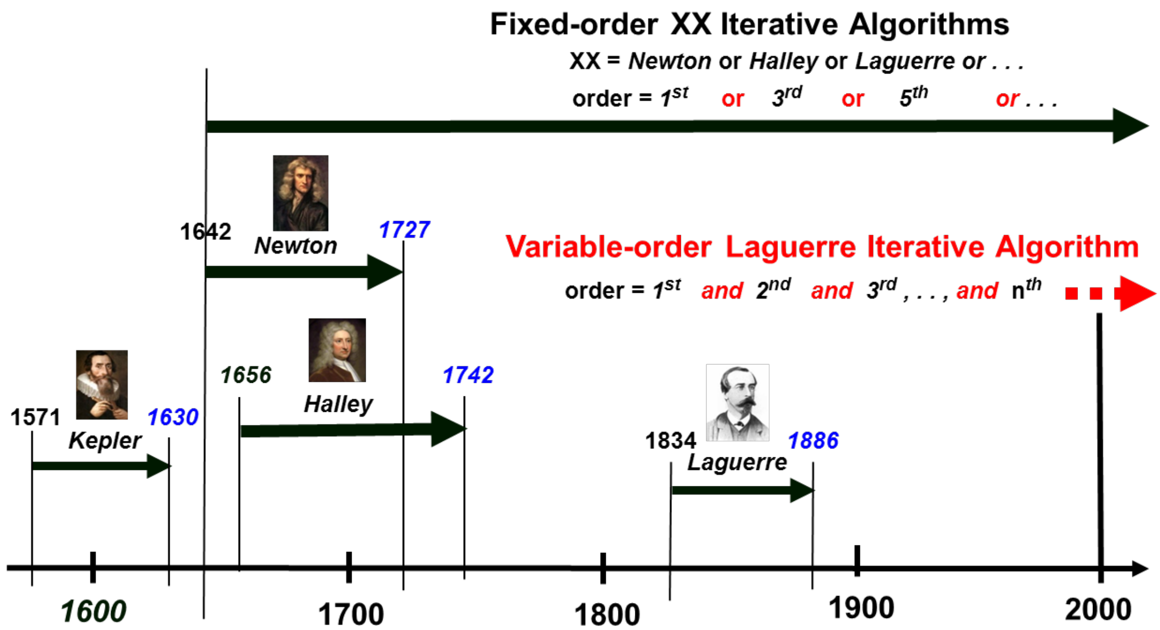

Fig. 1. The difference of fixed- and variable-order, the choice of "or" or "and"

The perturbed orbit propagators, such as the SGP4 and Vinti7 OP, must compute the correct solutions of both the 2-Body and perturbed Kepler equations. From page 7 of the 2006 paper of Vallado, "Revisiting SpaceTrack Report #3" [29], "**Solving Kepler's equation was updated, but not completely fixed. The solution of Kepler's equation continues to present challenges in astrodynamics hundreds of years after its introduction.**" The Keplerian Convergence problem that has remained unresolved for over 400 years is caused by the common use of fixed-order iterative algorithms of great minds instead of variable-order as depicted in Figure 1. It is almost unbelievable, if it were not stated in the outstanding paper of Vallado and his co-authors. This is another unsolved SGP4 problem making its use for autonomous orbit propagation undesirable.

The analytic Vinti7 OP computes both the 2-Body and Perturbed Kepler equations with the variable-order Laguerre iterative algorithm (Figure 2) such that every iterative solution converges to the correct final state vector at t_2. Also the position error of the final state vector is normally less than two kilometers per day on average for LEO orbit propagation between GPS locks of two days or less. Autonomous orbit propagation by Vinti7 OP has been demonstrated with the use of TLE and NASA debris catalog initial states through billions of simulations to correctly propagate orbits without theoretical singularity and Keplerian convergence problems. Incidentally the first-order Laguerre iterative algorithm (n = 1 in Figure 2) is that of Newton, which usually takes more iterations than the second-order Laguerre iterative algorithm (n = 2). The choice is using Laguerre iterative algorithm for variable-order is oblivious.

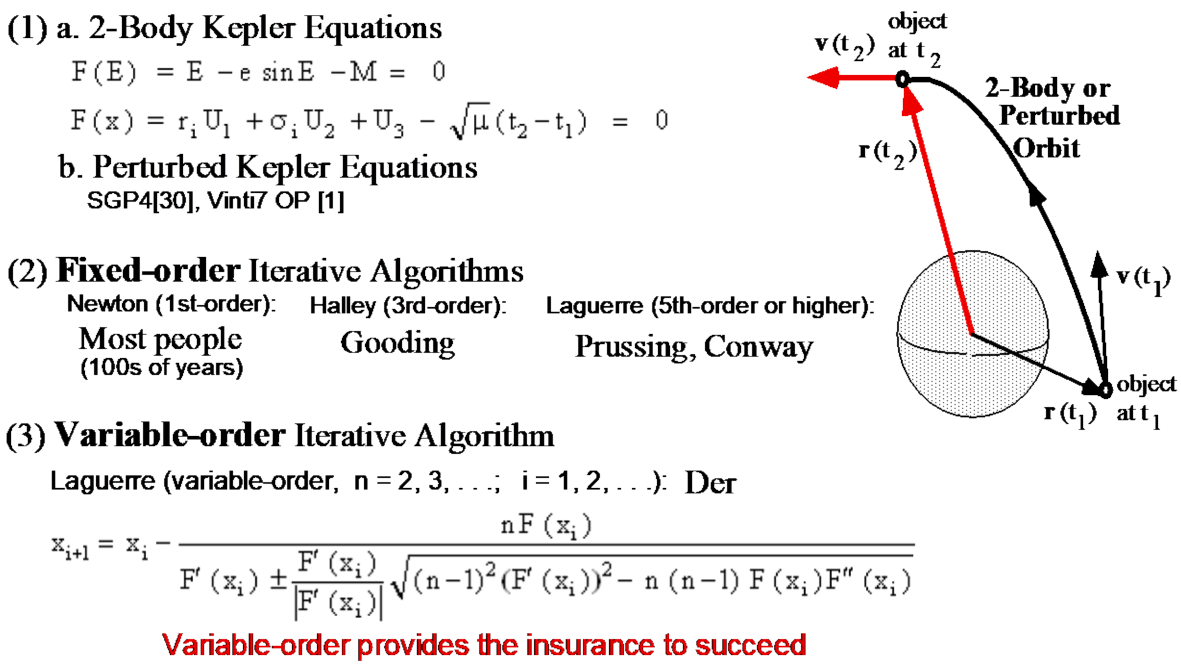

Fig. 2. Computing Kepler equations by the variable-order Laguerre iterative algorithm

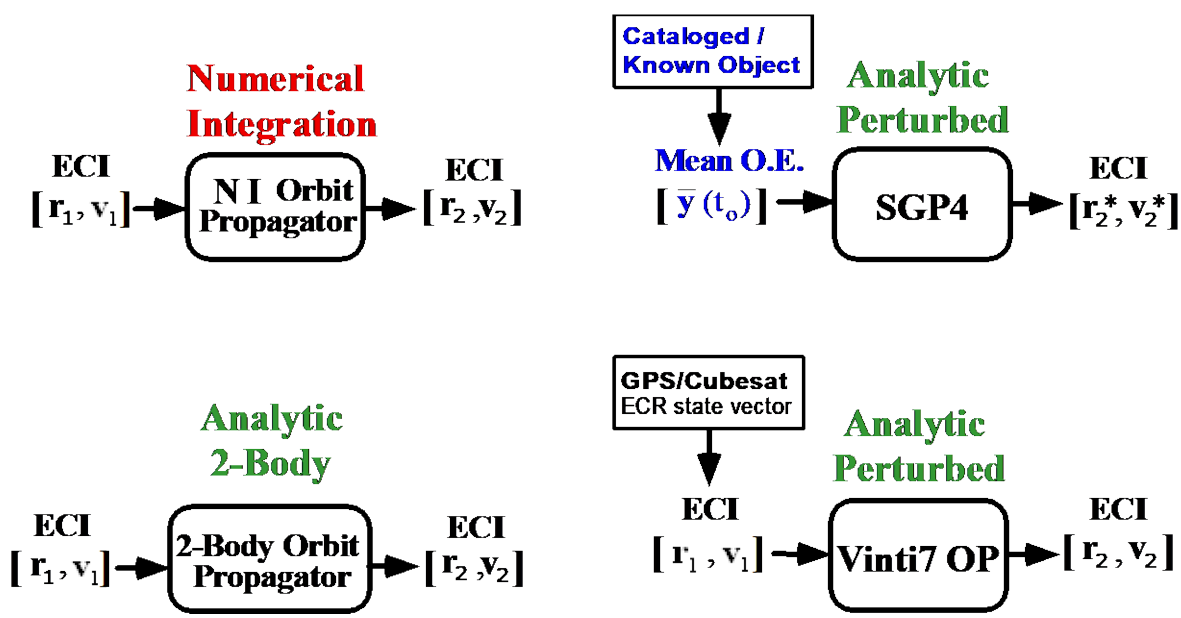

Fig. 3. Four orbit propagators with different I/Os for different purposes

4. INPUT AND OUTPUT COMPATIBILITY

The input and output compatibility of an orbit propagator is important for GPS equipped Cubesats. Figure 3 shows four orbit propagators with different I/Os for different purposes. In general, orbit propagation by numerical integration is accurate but slow, and analytic 2-Body is extremely fast but inaccurate. The analytic perturbed orbit propagators (SGP4 and Vinti7 OP), which propagate a few kilometers of errors per day for LEO objects for the first few days, present the only other choices. Since SGP4 requires Mean Orbital Elements (blue) as input, using the ECI state vector from GPS must require a conversion. Orbit propagation efficiency is reduced with mixed I/O. Given the theoretical singularity and Keplerian convergence problems of SGP4, as described above, the correct ECI state vector from the conversion cannot be guaranteed, let alone more lines-of-code are needed for conversion onboard. The analytic perturbed Vinti7 OP is efficient, since its I/O are ECI state vectors.

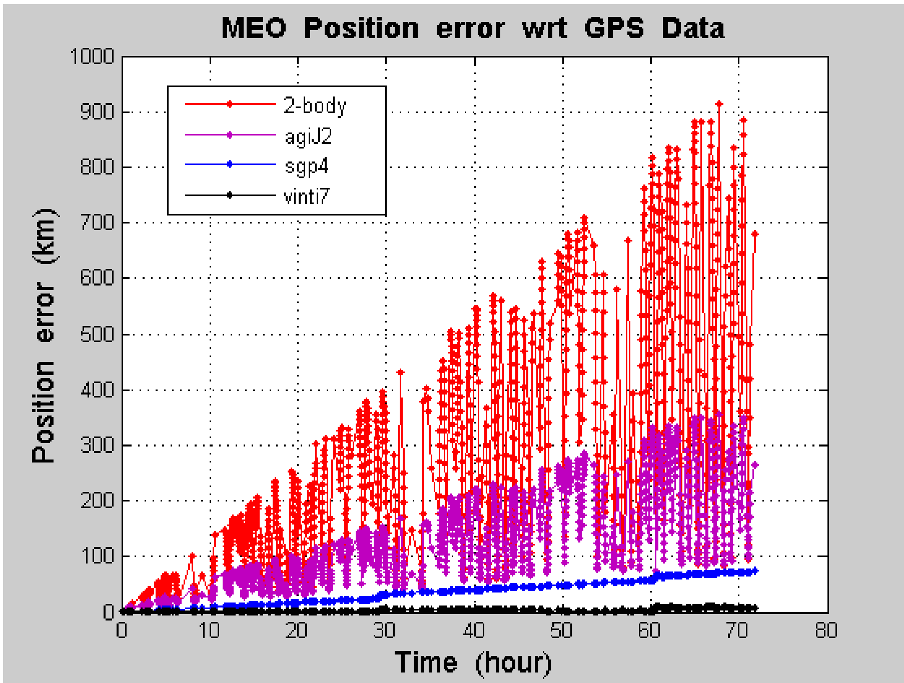

Fig. 4. Position errors for a MEO object computed by propagators 2-body, AGIJ2, SGP4, and Vinti7OP compared against GPS reference data over 72 hours and three GPS locks at 0.0, 29.4 and 60.6 hours.

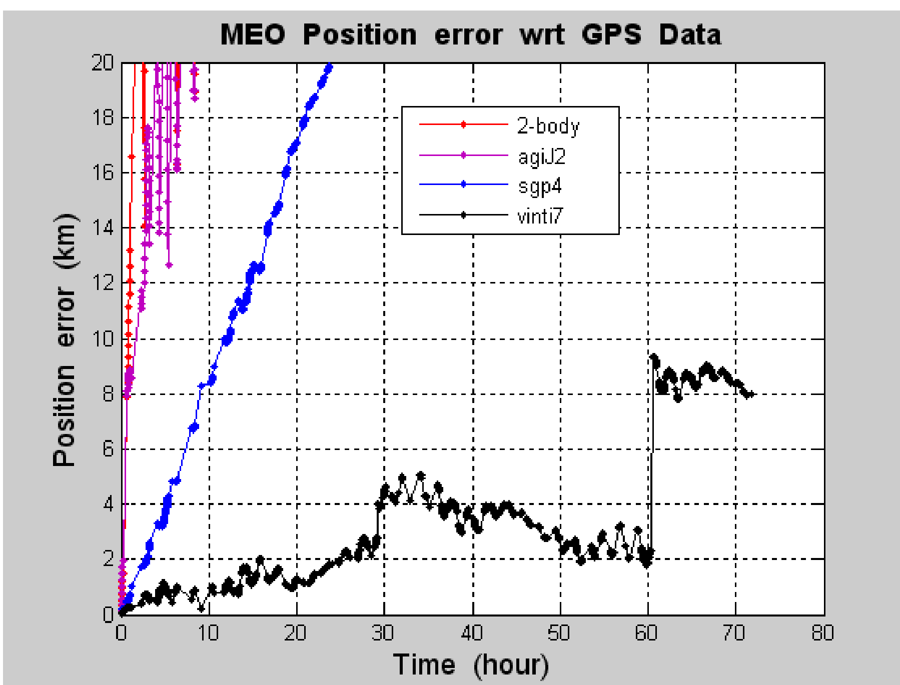

Fig. 5. Zoom in view of position errors for a MEO object in Figure 4.

5. ACCURACY COMPARISON USING GPS DATA

Figures 4 and 5 show accuracy comparison of four analytic orbit propagators using GPS data of a LEO/MEO object at an altitude of 1,400 km (particularly good for Vinti7 OP, with negligible atmospheric drag and Sun/Moon perturbations). Of course, an example with SGP4 outperforming Vinti7 OP in accuracy is possible. However, if orbit propagation with all objects (20,000 or so) in the TLE or SP space catalog is calculated for the duration of one day, the average Vinti7 OP position errors are usually 2X to 10X less than that of SGP4 using accurate (USAF) and publicly available TLE inputs.

Accurate, fast and robust orbit propagation is indispensable for small satellites with limited energy to operate guidance, navigation and control (GNC) and other applications. When an onboard GPS receiver is turned on, the accurate double-precision ECI state vector is quickly available and can be used directly by the Vinti7 OP. The GPS/Vinti predicted ECI state vector between GPS locks can be used for any purpose, such as transforming into a desired coordinate system, ECEF or (latitude, longitude, altitude), without any loss of accuracy. The conversion of an ECI state vector to mean orbital elements for SGP4 input is not needed.

In case of GPS receiver failure for a GPS equipped Cubesat (or a normal satellite), an ECI state vector may be uplinked for Vinti7 OP every 24 hours, if position accuracy of less than a kilometer is desired. However, from Figure 5, a 2-Body position error of about 20 kilometers (red) occurs in approximately two hours, and therefore an ECI state vector must be uplinked much more frequently.

6. CATALOGING CONSIDERATION

Initially it will be difficult to assign a TLE Satellite ID to new small satellites, especially unknown foreign Cubesats, unreported U.S. Cubesats, and many others [33]. For cooperative Cubesats, ECI state vectors can be computed between GPS locks with Vinti7 OP onboard, and periodically downloaded to ground together with other collected data. The Vinti7 propagated and GPS-fix ECI state vectors from the Cubesats can be easily processed on the ground, converting the data for the input to TLE catalog if desired. Cooperative Cubesat data could greatly benefit JSpOC as well as Cubesat owners/operators by relieving the need for tracking support from JSpOC and the Space Surveillance Network. Tracking and cataloging non-cooperative Cubesats is another challenge outside the scope of this paper.

7. VINTI7 ORBIT PROPAGATION IMPLEMENTATION

Pumpkin implemented the Vinti7 Orbit Propagator (OP) on its GPSRM 1 module, a CubeSat-compatible space-grade GPS receiver, utilizing the Vinti7 OP source code. The GPSRM 1 hosts a NovAtel® OEM615V GPS receiver along with a Supervisor MCU (SupMCU) running Pumpkin firmware. The SupMCU on the GPSRM 1 controls +5V and +3.3V energy to the OEM615V. When powered on by the SupMCU and running, the OEM615V draws approximately 1.1W while functioning as a GPS receiver. On a CubeSat, 1.1W can be a comparatively large energy draw; it is therefore often beneficial to limit GPS receiver ontime. The GPSRM 1's SupMCU is a 16-bit Microchip(R) PIC24EP256MC206 microcontroller running at a nominal 7.37MHz. It responds to SCPI commands over I2C (e.g., "GPS:POW ON") and provides telemetry over I2C.

The Vinti7 OP code spread across three C-language source modules (vinti7.h, vinti7.c & kepler2v.c) consists of 756 lines of code (LoC); C's <math.h> is required. All of the variables are implemented as C doubles (IEEE 754, 64 bits). The OP is included as one of several modules of code that together comprise the GPSRM 1 SupMCU firmware. The Vinti7 code occupies 27,734 bytes of firmware in the SupMCU; this does not include any math-related functions that are used exclusively by the Vinti7 algorithm. The Vinti7 OP is invoked via an I2C SCPI command to the GPSRM 1, and the propagated results are acquired by requesting Vinti 7 OP telemetry from the SupMCU, again over I2C, once the propagator is finished.

8. PERFORMANCE

A single Vinti7 computation, with varying positive and negative values for **t2** (propagated time, forwards or backwards) and using the remaining example input parameters below, requires less than 1.2s of compute time on the SupMCU:

```
t1    = 0.0;               // sec
r1[0] = -18982.9116920829; // km        v1[0] =  2.96990008479315; // km/s
r1[1] = -25047.1371788540; // km        v1[1] = 0.329975213844251; // km/s
r1[2] = -173.044152439773; // km        v1[2] = 0.265794505165530; // km/s
```

Time of propagation	t2 input to propagator (seconds)	SupMCU Time-to-compute (seconds)	SupMCU Energy-to-compute (mWs)
1 minute	60.0	0.69	34
	-60.0	0.69	34
1 hour	3600.0	0.68	34
	-3600.0	0.68	34
1 day	86400.0	1.10	55
	-86400.0	0.72	36
1 week	604800.0	1.06	53
	-604800.0	0.75	38
1 month / 30 days	2592000.0	0.74	37
	-2592000.0	1.07	54
1 year / 365 days	315360000.0	0.93	47
	-315360000.0	0.82	41

Table 1: Example Vinti7 OP compute times and energies for varying t2 inputs; on Pumpkin GPSRM 1 (7.37MHz PIC24EP256MC206 SupMCU, GPSRM 1 firmware v0.3.9a)

With its 7.37MHz clock, the 3.3V SupMCU's instantaneous energy consumption attributed to the Vinti7 OP calculations barely exceed 50mW-s, less than 1/20th of the energy consumption of the OEM615V GPS module. The Vinti7 OP yields a solution in under 1.2s and requires less than 60mW-s of energy; the OEM615V typically requires on the order of seconds to tens of seconds to achieve a GPS lock, thereby requiring 20-200 times as much energy per fix.

Both SGP4 and Vinti7 are analytic prediction algorithms that include perturbations. These advanced prediction algorithms require an initial guess of the final state vector at t2, which is the iterative solution of the 2-body Kepler equation. Clearly the 2-body initial guesses for forward and backward propagation (computed from a given initial state, initial time t1 and final time t2) are not the same, which leads to the first reason of why the SupMCU Time-to-compute are different for forward and backward propagation in Table 2. After the first guess, "perturbations are added" giving a second guess to solve iteratively the "generalized or perturbed Kepler equation". As a consequence, the number of iterations required to converge to the desired solution of the generalized Kepler equation at t2 for forward and backward prediction becomes random due to the accuracies of the first and second guesses. This is the second reason why the SupMCU Time-to-compute values are different or random in Table 2. The stringent convergence criterion of solving the generalized Kepler equation in Vinti7 OP, which also plays a role in the randomness, was originally designed for GHz computers. It can be relaxed for the MHz SupMCU in future Vinti7 OP releases such that one prediction can be computed in less than 1.0s without any loss of accuracy.

9. USAGE

The typical utilization of the GPSRM 1 is to obtain an occasional fix via the OEM615V GPS receiver, and then propagate this fix using the Vinti7 OP until the next GPS fix obtained from the OEM615V. Based on energy available to the CubeSat, the OEM615V GPS receiver might be enabled once every 24 hours for a few minutes

until a GPS lock is acquired; the Vinti7 OP would provide updated position and velocity as demanded by the CubeSat until the next OEM615V fix.
The Space & Systems Design Laboratory (SSDL) at Stanford University incorporated the Pumpkin GPSRM 1 into the joint Stanford/JPL LMRST-Sat CubeSat. The LMRST-Sat flight software (FSW) i) enables the OEM615V receiver for up to 45 minutes every 25 hours, to acquire a GPS fix; ii) provides a means of displaying LMRST-Sat's last positional fix (latitude, longitude & altitude (LLA)) via its beacon -- the fix can come from the OEM615V or from the Vinti7 OP; iii) commands the GPSRM 1 to enable and disable the OEM615V, as well as to perform the Vinti7 calculations and return the results; and iv) performs the necessary coordinate transformations and time conversions. Because the LMRST-Sat FSW sought to compare the SGP4 and Vinti7 OP, a minimal set of coordinate transformations applicable to both OPs was written into the FSW.

The OEM615V GPS receiver outputs position and velocity in the Earth-centered, Earth-fixed (ECEF) coordinate system via its LOG BESTXYZA command. The Vinti7 OP's inputs and outputs are in Earth-Centered Inertial (ECI) coordinate frames. In the LMRST-Sat FSW's default operating mode, the last (ECEF) fix acquired from the OEM615V GPS receiver is used as the basis for the input to the Vinti7 OP. GPS time (week and msec) is converted to Modified Julian Date (MJD); MJD is then converted to Greenwich Mean Sidereal Time (GMST). This ECEF and GMST data are then converted to ECI data. To simplify the command and telemetry interface to the Vinti7 OP, t1 is fixed at 0, and the command to the Vinti7 OP on the SupMCU consists of the ECI position (r1, in km) and velocity (v1, in km/s) in three axes, plus the desired propagation time t2, in seconds. The response (i.e., telemetry) back from the SupMCU consists of ECI-frame data of the newly-propagated position and velocity in three axes. The new time is converted through GPS, UTC and MJD times to GMST; ECI data is converted to ECEF; ECEF data is converted to LLA for display via LMRST-Sat's beacon. The LMRST-Sat FSW can also assist the GPS receiver in acquiring a lock, by utilizing propagated LLA and time and commanding the OEM615V via its SETAPPROXPOS and SETAPPROXTIME commands, respectively.

This process is managed in a collection of functions (e.g., ECEF to ECI conversion) and tasks (e.g., a task to manage the propagator) within the LMRST-Sat's FSW, operating on an MSP430F2618 running at 12MHz. These C modules total approximately 1184 LoC, 10.8KB program memory and 3.3KB RAM. The "meat" of the LMRST-Sat FSW's propagator task is listed below for reference:

```
if (gps_receiver_status()->valid_fix) {
        // TIME OF PROPAGATION
        // GPS reading of the last valid fix: time
        propagated_data.call_time = OSGetTicks() / TICKS_PER_SEC;

        // Prop_time is the time in seconds to propagate forward from the last valid fix; feed to propagator
        prop_time = propagated_data.call_time - gps_last_fix()->call_time + gps_receiver_status()->prop_delay;

        // LAST FIX AND DATA CONVERSIONS
        // GPS reading of the last valid fix: state
        ECEF.rx = gps_last_fix()->x;
        ECEF.ry = gps_last_fix()->y;
        ECEF.rz = gps_last_fix()->z;
        ECEF.vx = gps_last_fix()->vx;
        ECEF.vy = gps_last_fix()->vy;
        ECEF.vz = gps_last_fix()->vz;
        // GPS time of last fix
        gps.week = gps_last_fix()->week;
        gps.msec = gps_last_fix()->msec;
        // GPS time to UTC time
        gps2utc(&gps, &utc);
        // GPS time to seconds
        gps2sec(&gps, &sec);
        // UTC to MJD; (MJD, UTC) to GMST
        GMST = mjd2gmst(utc2mjd(&utc), &utc);
        // ECEF to ECI
        ecef2eci(ECEF, GMST, &ECI);

        // RUN PROPAGATOR ON REMOTE SUPMCU / GPSRM 1
        // Now that the data is set, send the i2c commands to the SUP MCU to run the right propagator
        run_propagator();
        // Wait for command to be sent; average command takes ~2 sec to execute
```

```
OS_Delay(250);
// Send/write the command that will asks for the propagator output
I2C_write((unsigned char *)STR_SUPMCU_PROPREAD, strlen(STR_SUPMCU_PROPREAD),
I2C_ADDR_GPSRM1);
// Wait for command to be sent
OS_Delay(150);
// Read the raw propagation response and store the string into local string
I2C_read(str_tmp, VINTI_RESPONSE_SIZE, I2C_ADDR_GPSRM1);

// PROPAGATED DATA AND CONVERSIONS
// Parse the response into 8 byte sections and store them so they can be accessed with a function
memcpy(&ECI.rx, str_tmp + 22, sizeof(double));
memcpy(&ECI.ry, str_tmp + 30, sizeof(double));
memcpy(&ECI.rz, str_tmp + 38, sizeof(double));
memcpy(&ECI.vx, str_tmp + 46, sizeof(double));
memcpy(&ECI.vy, str_tmp + 54, sizeof(double));
memcpy(&ECI.vz, str_tmp + 62, sizeof(double));
// Advance the time of the propagation
sec = sec + prop_time;
// seconds to GPS time
sec2gps(sec, &gps);
// GPS time to UTC time
gps2utc(&gps, &utc);
// UTC to MJD; (MJD, UTC) to GMST
GMST = mjd2gmst(utc2mjd(&utc), &utc);
// ECI to ECEF
eci2ecef(ECI, GMST, &ECEF);
// ECEF to LatLonAlt
ecef2lla(ECEF, &LatLon_PROP_OUT);
// Put the output data in type_gps_reading structure
propagated_data.latitude  = LatLon_PROP_OUT.lat;  // Latitude [ddmm.mmmm] (Negative = South)
propagated_data.longitude = LatLon_PROP_OUT.lon;     // Longitude [dddmm.mmmm] (Negative =
West)
propagated_data.altitude  = LatLon_PROP_OUT.alt;        // Altitude from Mean Sea Level [meters]
propagated_data.x      = ECEF.rx;
propagated_data.y      = ECEF.ry;
propagated_data.z      = ECEF.rz;
propagated_data.vx                 = ECEF.vx;
propagated_data.vy                 = ECEF.vy;
propagated_data.vz                 = ECEF.vz;
propagated_data.week     = gps.week;
propagated_data.msec     = gps.msec;
}
```

10. CONCLUSIONS

This paper describes the computational performance and implementation of the Vinti algorithm for GPS equipped Cubesats to perform autonomous orbit propagation between GPS locks. The Vinti7 OP is accurate, fast, robust, easy to use and implement, and consumes a few percent of energy as compared to a GPS lock. Since Vinti7 OP has no singularity whatsoever and the Kepler equations are computed correctly, accurate orbit propagation has been extensively tested for objects in all orbit regimes. If the time between two GPS locks is 24 hours or less, the position errors are minimal as compared to real or reference data, and therefore the Vinti7 OP has the flexibility to accommodate longer times, a few days, if desired between two GPS locks.

Pumpkin integrated the Vinti7 OP into the Stanford/JPL LMRST-Sat Cubesat, a Cubesat-compatible space-grade GPS receiver. Compared to the energy required to operate a GPS receiver, the Vinti7 OP provides a very low-energy method for propagating a Cubesat's position and velocity. A 7.37MHz 16-bit MCU with IEEE 754 floating-point support can complete all calculations associated with an arbitrary Vinti7 orbit propagation in under 1.2s, requiring energy under 60mW-s per propagation. A 16-bit MCU can also perform the time and coordinate conversions between ECEF, ECI and LLA coordinate systems that are required when exchanging information between a GPS receiver and the Vinti7 OP. The OP can be used in place of the GPS receiver, until the next GPS lock. The Vinti7 OP as implemented on the Pumpkin GPSRM 1 typically consumes less than 1% of the energy required by the OEM615V GPS receiver.

11. ACKNOWLEDGMENTS

At Stanford, graduate students Stephen Wolf, Connor Beierle, Grant McLaughlin and Sonja Brajovic wrote all of LMRST-Sat's FSW with respect to interfaces to the OEM615V GPS receiver and the Vinti7 orbit propagator. The orbit propagation material in this paper is partially supported by the research funding of Null Geodesic Alterations, LLC.

12. REFERENCES

1. Vinti, J.P., *Orbital and Celestial Mechanics*, AIAA, Volume 177, 1998.
2. Escobal, P. R., *Methods of Orbit Determination*, Robert E. Krieger Publishing Company, Malabar, Florida, USA, 1965.
3. Bate, R., Mueller, D., and White, W., *Fundamentals of Astrodynamics*, Dover Publications, Inc., Toronto, Canada, 1971.
4. Roy, A. E., *Orbital Motion*, Institute of Physics Publishing, (1st, 2nd, 3rd Editions), 1978, 1982, 1988.
5. Battin, R.H., *An Introduction to The Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, Education Series, 1987.
6. Chobotov, V. A. ed, *Orbital Mechanics*, AIAA, Education Serious, 1991.
7. Prussing, J.E., Conway, B.A., *Orbital Mechanics*, Oxford University, 1993.
8. Vallado, D. and McClain, W. D., *Fundamentals of Astrodynamics and Applications*, (1st, 2nd, 3rd Editions), 1997, 2001
9. Montebruck, O. and Gill, O., *Satellite Orbits*, Springer Verlag, 2000.
10. Tapley, B.D., Schutz, B.E.., and Born, G.H., *Statistical Orbit Determination*, Elsevier Inc, 2004.
11. Tewari, A., *Atmospheric and Space Flight Dynamics*, Birkhauser, 2007.
12. Wiesel, W.E., *Modern Astrodynamics*, (1st, 2nd Editions), 2003, 2010.
13. Wiesel, W.E., *Modern Orbit Determination*, (1st, 2nd Editions), 2003, 2010.
14. Milani, A., Gronchi, G. F., *Theory of Orbit Determination*, Cambridge University Press, 2010.
15. Herrick, S., *Astrodynamics*, Vol 1, Van Nostrand Reinhold, London, 1971.
16. Goldstein, H., *Classical Mechanics*, Second EditionAddison-Wesley Publishing Company, 1980.
17. Woodburn, J., "Methods of Orbit Propagation", *AGI 2005 User' Conference*, Oct. 2005. www.agi.com/downloads/events/2005-users-conference-resources/crash-courses-1-lecture Series/CCLS.MethodsOfOrbitPropagation.Woodburn.UC05.v2.ppt
18. Vetter, J.R., "Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods," *Johns Hopkins APL Technical Digest*, Vol. 27, #3, 2007.
19. Greene, M.R., Zee, R.E., "Increasing the Accuracy of Orbital Position Information from NORAD SGP4 Using Intermittent GPS Readings," *Small Satellite conference*, 2009.
20. Brouwer, D., "Solution of the Problem of Artificial Satellite Theory without Drag," *The Astronomical Journal*, Vol. 64, No. 1274, November 1959.
21. Vinti, J.P., "Theory of the Effect of Drag on the Orbital Inclination of an Earth Satellite," *Journal of Research of the National Bureau of Standards*, Volume 62, No. 2, pp. 79-88, February 1959.
22. Long, A.C., Cappellari, J.O., Velez, C.E., Fuchs, A.J., *Goddard Trajectory Determination System (GTDS) Mathematical Theory*, Chapter 5.12, Revision 1, 1989.
23. Getchell, B.C., "Orbit Computation with the Vinti Potential and Universal Variables," *Journal of Spacecraft*, Volume 7, No. 4, April, 1970.
24. Wadsworth, D.V., "Vinti Solution for Free-Flight Rocket Trajectories," *AIAA Journal 1*, pp.1351-1354, June, 1963.
25. Borchers, R.V., "A Satellite Orbit Computation Program for IZSAK's Second-Order Solution of Vinti's Dynamical Problem," *Goddard Space Flight Center Technical Note, NASA TN D-1539*, February, 1965.
26. Wiesel, W.E., "Numerical Solution to Vinti's Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 9, 2014.
27. Biria, A.D., Russell, R.P., "Periodic Orbits in the Elliptical Relative Motion Problem with Space Surveillance Applications," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015.
28. Der, G.J., " Brouwer_vs_Vinti," http://derastrodynamics.com/docs/brouwer_vs_vinti_v1.pdf
29. Vallado, D.A., Crawford, P., Hujsak, R., "Revisiting Spacetrack Report #3," AIAA, 2006-6753, 2006.

30. Hoots, F.R., Roehrich, R.L., "Spacetrack Report No. 3, Models for Propagation of NORAD Element Sets," *Office of Astrodynamic Applications*, Fourteenth Aerospace Force, Ent AFB, December, 1980.
31. Der, G.J., Danchick, R " Conversion of Osculating Orbital Elements to Mean Orbital Elements," *Flight Dynamics Conference*, Goddard SFC, 1995.
32. Lee, B.S., "NORAD TLE Conversion from Osculating Orbital Elements," *J. Astro. Space Sci.* 19(4), 2002.
33. JSpOC, JFCC SPACE, "JSpOC Recommendations for Optimal Cubesat Operations," August 4, 2015.