A FLEXIBLE APPROACH FOR THE NUMERICAL SOLUTION OF THE INS MECHANIZATION EQUATIONS

José J. Rosales and Ismael Colomina
Institute of Geomatics
Generalitat de Catalunya & Universitat Politècnica de Catalunya
Castelldefels
Spain

Keywords: Inertial Navigation Systems, INS/GNSS integration, ordinary differential equations, numerical analysis.

Abstract

This paper describes an alternative to numerically integrate the differential equations of a strapdown inertial navigation system (INS). The use of predictor-corrector multistep methods with variable step-size is discussed and a fourth-order method of the said type is derived.

The paper starts with a brief review of the most popular numerical methods used in the solution of the first order differential equations of motion of strapdown inertial navigation systems. Then some of their limitations in practical applications are discussed; namely that they are not well suited to the realities of INS and Global Navigation Satellite Systems (GNSS) data streams with data gaps, unsynchronized sensors and deviations from nominal sampling frequencies. Under this circumstances, traditional fixed step-size algorithms lead to either low order methods or cumbersome algorithmic patches.

In the central part of the paper it is shown how variable step-size multistep methods inherit all the advantages of single and multistep methods while avoiding most —if not all— of their pitfalls. Last, a detailed derivation of a fourth-order predictor-corrector multistep methods with variable step-size is given.

1 Introduction

The solution of the differential equation model governing the dynamics of a strapdown inertial navigation system cannot be described by analytical and closed formulae; hence, it has to be solved by means of numerical algorithms. To the best knowledge of the authors, the classical approaches for the numerical integration of such models are the transition matrix, fourth-order Runge-Kutta (RK4) or the combined use of one-step first-order Euler method and RK4. [A numerical method is called n^{th} order if its error term is $\mathcal{O}(h^{n+1})$.] The transition matrix has the main drawback that it significantly increases the computational burden if a reasonable precision for the solution is required. Euler method has the advantage that it does not depend on the integration step (the significance of this fact will be perceived later) but, on the other hand, it is of poor accuracy. RK4, however, is accurate enough but it is based on evaluations of the field that defines the differential equation. This circumstance introduces an extra difficulty due to the fact that an inertial observation is needed every time the field defining the strapdown INS model has to be evaluated. The classical RK4 method is defined with a fixed step-size, therefore the time interval between two consecutive inertial observations must be constant. Unfortunately, although theoretically there is a nominal frequency at which the Inertial Measurement Unit (IMU) outputs inertial observations, in practice the inertial measurements slightly depart from this nominal value. This can be overcame, for example, by interpolating the inertial observations at the nominal frequency. However, manipulation of the inertial observations is not optimal from a numerical point of view and it is an option to be avoided if possible. An alternative to the inertial data interpolation is the use of the Euler method. When the difference between the nominal frequency and the time span between two consecutive inertial measurements is above certain threshold, the integration step can be set as the time difference between such measurements and then, an Euler step performed. This avoids the need of interpolating, but introduces a loss of precision in the navigation solution due to the use of a first-order method.

Another interesting scenario is the navigation with INS aided with GNSS observations, a synergy known as *INS/GNSS navigation*. The classical approach to INS/GNSS navigation is the Kalman filter. The procedure basically is the following: the navigation solution is estimated at each epoch by means of integrating the INS mechanization equations (see section 2) until a GNSS observation is available. Then, by means of a Kalman filter step, an optimal[1] state is estimated. After the filter step, the navigation solution is again estimated by means of the integration of the INS mechanization equations until the next GNSS observation and so forth. This algorithm assumes the GNSS observation and the state prediction eventually concur at some epoch; depending on the quality of the INS/GNSS components and on the acquisition configuration, this coincidence might be quite rare. This leads to a data manipulation or to algorithm patches such as the described in the previous

[1] optimal in the least-square norm sense.

paragraph. It is worth to note that the problems related with the integration step-size do not happen when using the Euler method alone.

The goal of this paper is to design a numerical method for the integration of ordinary differential equations, avoiding all the problems exposed above while preserving the good properties of the three methods presented. In other terms, the goal is to develop a method that does not depend on the integration step-size (as the Euler or the transition matrix methods do) with a reasonable order of accuracy (such as the fourth-order Runge-Kutta), considering also the computational aspects. The proposed alternative is a fourth-order predictor-corrector multistep method with variable step-size. In this paper, the main features of the Euler, RK4 and the transition matrix methods are presented as well as the problems that arise when they are implemented in an Inertial Navigation System. After analyzing the performance of these methods, the proposed alternative to overcome the pitfalls of these approaches is presented.

2 The INS mechanization equations and their integration.

The stochastic differential equation 1 is the mechanization model of an strapdown inertial navigation system in an ECEF coordinate system.

$$\begin{aligned} \dot{\boldsymbol{x}}^e &= \boldsymbol{v}^e + \boldsymbol{w}_v \\ \dot{\boldsymbol{v}}^e &= R_b^e(f^b + \boldsymbol{a}^b + \boldsymbol{w}_f) - 2\Omega_{ie}^e \boldsymbol{v}^e + g^e(\boldsymbol{x}^e) \\ \dot{R}_b^e &= R_b^e\left(\Omega_{ei}^b + \Omega_{ib}^b(\omega_{ib}^b + \boldsymbol{o}^b + \boldsymbol{w}_\omega)\right) \\ \dot{\boldsymbol{o}}^b &= -\beta \boldsymbol{o}^b + \boldsymbol{w}_o \\ \dot{\boldsymbol{a}}^b &= -\alpha \boldsymbol{a}^b + \boldsymbol{w}_a \end{aligned} \tag{1}$$

In this set of equations, f^b and ω_{ib}^b are, respectively, the linear acceleration and the angular rate sensed by the IMU. $\boldsymbol{x}^e$, $\boldsymbol{v}^e$, R_b^e are the position, velocity and the rotation matrix that transforms the coordinates in the body-frame to the ECEF-frame respectively; $\boldsymbol{w}_v$, $\boldsymbol{w}_f$, $\boldsymbol{w}_\omega$, $\boldsymbol{w}_o$ and $\boldsymbol{w}_a$ are white noise generalized processes. Finally, $\boldsymbol{o}^b$ and $\boldsymbol{a}^b$ are the biases for the angular rate sensors and the linear accelerometers respectively and $\alpha, \beta > 0$ the inverse of the correlation times. Other errors such as scale factors or misalignments are not considered for the sake of simplicity. As the objective of this paper is not to disclose about the intricacies of this model, the reader is referred to [2] for a complete description of the equation 1 and its representation in different reference frames.

The way to numerically solve this stochastic model is to integrate the associated differential equation, this is, the ordinary differential equation resulting from taking apart the functional and stochastic components; on the other hand, the covariance is propagated according to the statistical properties of the noise. For further information about the numerical integration of stochastic differential equations, the reader is referred to [4]. This paper deals with the numerical integration of the associated ordinary differential equation. As already mentioned, this differential equation does not allow for an analytical closed formula of the solution, therefore a numerical algorithm has to be used for its integration. The next subsection explains how (and why) the classical —rather, the "popular"— methods used for the numerical integration of equation 1 fail in many INS/GNSS hybrid systems.

2.1 The integration issue

Two considerations have to be taken into account when stating the problem of the numerical integration of equation 1. The first concerns to the numerical method and the second to the nature of the inertial observations. The ideal numerical method would be a method of high accuracy, step-size independent and fast. Unfortunately, usually high accuracy requirements increase the computational burden, making the method slower. Hence, a compromise between accuracy and velocity is required. For navigation purposes, fourth-order methods are a good compromise and usually they fulfill the accuracy requirements. A step-size independent method would not be essential if the field defining the differential equation could be evaluated at whatsoever epoch. For example, if the field had an analytical expression not depending on external observations, this condition would not be necessary at all. Unfortunately, this is not the case of the INS mechanization equations. The model described in equation 1 depends on the inertial measurements —linear accelerations and angular rates— provided by the IMU. Again, step-size independency would not be required if the frequency at which the IMU provides inertial measurements would be constant (typically IMUs output inertial measurements between 50Hz and 400Hz, what is enough for integration purposes). But, as said previously, although there is a nominal frequency, the inertial observations slightly depart from it; even worse, data gaps might occur. Hence, the method becomes step-size dependent.

The next subsections present an overview of three popular methods for the numerical integration of ordinary differential equations: Euler, RK4 and the transition matrix. The Euler method has order one and is step-size independent and the RK4 is of order four but step-size dependent. The impact of such features in the

integration of the INS mechanization equations has been already mentioned. The transition matrix is step-size independent and might have arbitrary precision. Nevertheless, it is expensive from a computational point of view, only holds for linear dynamical systems and assumes that the [time dependent] linear relationship between states and observations is constant between two consecutive epochs. Moreover, the transition matrix has to be recalculated each epoch. For the sake of generality, instead of the INS mechanization model, a more general differential equation model will be considered. The statement of the problem is that of the classical initial value problem: given a differential equation of the form

$$\left\{ \begin{array}{lcl} \dot{\boldsymbol{x}} & = & \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \quad \text{for } t \in [t_0, t_f] \\ \boldsymbol{x}(t_0) & = & \boldsymbol{x}_0 \end{array} \right. \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the time dependent state vector, $\boldsymbol{u} \in \mathbb{R}^m$ the observations' vector and t the time, the goal is to numerically compute the solution. From now on, the notation will be the following:

- $\hat{\boldsymbol{x}}$ stands for the numerical approximation of the solution, while $\boldsymbol{x}$ stands for the real solution,
- $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$,
- $\boldsymbol{u}(t_k) = \boldsymbol{u}_k$,
- $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k)$,
- $\hat{\boldsymbol{f}}_k = \boldsymbol{f}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k, t_k)$,
- $h_k = t_{k+1} - t_k$.

2.1.1 One-step Euler method

This method requires a low computational burden, it is of easy implementation and is step-size independent but, alas, is of very poor accuracy ($\mathcal{O}(h^2)$) and highly unstable. However, as it will be seen later, this method might be very useful for some purposes.

Assume that the initial value problem 2 has to be numerically solved. By the formal definition of the derivative of a function, it holds that

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{h_k} \approx \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k)$$

Therefore, isolating $\boldsymbol{x}_{k+1}$ and considering that only approximations of the solution are used, it can be straightforwardly seen that

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + h_k \boldsymbol{f}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k, t_k)$$

Note that in this case, the integration step h_k might not be constant. The integration step can be set as the time span between two consecutive inertial observations and the field can be evaluated. As it will be seen in the next section, this privileged feature is not shared with the RK4.

2.1.2 Fourth-order Runge-Kutta method

The fourth-order Runge-Kutta is perhaps the most known and widely spread numerical method for integrating ordinary differential equations. The general form of an explicit Runge-Kutta of arbitrary order r with fixed integration step h is

$$\left\{ \begin{array}{lcl} \hat{\boldsymbol{x}}_{k+1} & = & \hat{\boldsymbol{x}}_k + h \sum_{j=1}^{r} A_j \boldsymbol{k}_j(\hat{\boldsymbol{x}}_k, h, t_k) \\ \boldsymbol{k}_1(\hat{\boldsymbol{x}}, h, t) & = & \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}, t) \\ \boldsymbol{k}_j(\hat{\boldsymbol{x}}, h, t) & = & \boldsymbol{f}(\hat{\boldsymbol{x}} + h \sum_{s=1}^{j-1} b_{js} \boldsymbol{k}_s(\hat{\boldsymbol{x}}, \boldsymbol{u}, t + a_j h), \boldsymbol{u}(t + a_j h), t + a_j h), \quad j = 2, 3, \ldots, r \end{array} \right.$$

Note that for $r = 1$, this method reduces to the one-step Euler method explained in the previous section. The classical fourth-order method is with $r = 4$. The following schema corresponds to it:

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + h\Big\{\frac{1}{6}\boldsymbol{k}_1 + \frac{1}{3}\boldsymbol{k}_2 + \frac{1}{3}\boldsymbol{k}_3 + \frac{1}{6}\boldsymbol{k}_4\Big\} \quad \text{where}$$

$$\begin{array}{lcl} \boldsymbol{k}_1 & = & \boldsymbol{f}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}(t_k), t_k) \\ \boldsymbol{k}_2 & = & \boldsymbol{f}(\hat{\boldsymbol{x}}_k + h(\boldsymbol{k}_1/2), \boldsymbol{u}(t_k + (h/2)), t_k + (h/2)) \\ \boldsymbol{k}_3 & = & \boldsymbol{f}(\hat{\boldsymbol{x}}_k + h(\boldsymbol{k}_2/2), \boldsymbol{u}(t_k + (h/2)), t_k + (h/2)) \\ \boldsymbol{k}_4 & = & \boldsymbol{f}(\hat{\boldsymbol{x}}_k + h\boldsymbol{k}_3, \boldsymbol{u}(t_k + h), t_k + h) \end{array}$$

The interested reader is referred to [1, 3, 5] for further information about other particular features of the family of Runge-Kutta methods. Note that this method requires four evaluations of the field at different points and times. Note, as well, that a higher r requires more evaluations, increasing this way the computational burden. Moreover, fields with no dependencies on external observation or with constant time span between consecutive observations are required. Non of this conditions are fulfilled in practice by inertial navigation systems in real conditions.

2.1.3 *Transition matrix*

As already said, the transition matrix method suits only for linear dynamical systems. However, the differential equation 2 is not necessary linear. Hence, instead of the differential equation 2, the linear model

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \quad \text{for } t \in [t_0, t_f] \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}_0 \end{cases}$$

will be considered. Note that equation 1 can be expressed in such a fashion.

Assume that the state at epoch t_k is known and the solution at epoch t_{k+1} has to be computed. Assume further the time span between t_k and t_{k+1} is small; hence, $\boldsymbol{A}(t)$ and $\boldsymbol{B}(t)$ can be considered as constant matrices, $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ respectively. It can be shown ([4]) that the solution can be estimated as follows:

$$\hat{\boldsymbol{x}}_{k+1} = \Phi(t_{k+1}, t_k)\hat{\boldsymbol{x}}_k + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\boldsymbol{B}_k\boldsymbol{u}(\tau)d\tau$$

where

$$\Phi(t_{k+1}, \tau) = e^{\boldsymbol{A}_k\Delta_\tau} \tag{3}$$

with $\Delta_\tau = t_{k+1} - \tau$. From a computational point of view, equation 3 can be approximated by means of its expansion in power series up to some degree j.

$$\Phi_j(t_{k+1}, \tau) = I + \boldsymbol{A}_k\Delta_\tau + \frac{1}{2!}(\boldsymbol{A}_k\Delta_\tau)^2 + \frac{1}{3!}(\boldsymbol{A}_k\Delta_\tau)^3 + \cdots + \frac{1}{j!}(\boldsymbol{A}_k\Delta_\tau)^j \tag{4}$$

The accuracy of the navigation solution depends on the approximation of equation 3. As more terms of the serie 4 are calculated, more accurate the solution is. However, this matrix has to be computed at every epoch. Hence, this method highly increases the computational burden if high accuracy is desired, specially when the state vector is of big dimension. The interested reader is referred to [2, 4] for more information about the transition matrix.

3 Multistep and predictor-corrector methods

The numerical methods exposed in the previous sections shared a common feature: the estimation of the state at the epoch t_{k+1} depended only on the solution given at epoch t_k. The family of methods with such characteristic are called *one-step* methods. Contrary on the one-step methods, *multistep* methods use previous states $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{x}}_{k+1}, \ldots, \hat{\boldsymbol{x}}_{k+r-1}, r \geq 2$ for the estimation of the state at epoch t_{k+r}, $\hat{\boldsymbol{x}}_{k+r}$. The order of the method is characterized by the number r; if r previous states are used to compute the solution, it can be proved that the method has order r. On the other hand, if r states are needed to estimate the state at every epoch, the states at the r first epochs are needed in order to start integrating. As this information is not usually available, the use of a one-step method $r-1$ times (for example, Euler method) in order to initialize the multi-step method can be an option.

Assume now that the initial value problem 2 has to be solved for epoch t_{k+r}, and that the solutions $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{x}}_{k+1}, \ldots, \hat{\boldsymbol{x}}_{k+r-1}, r \geq 2$ are available. The two fundamental ideas behind the multistep methods are the fundamental theorem of calculus and the interpolation by polynomials. The fundamental theorem of calculus states that the following equality holds for a continuous function $\boldsymbol{f}$

$$\boldsymbol{x}_{k+r} - \boldsymbol{x}_{k+r-1} = \int_{t_{k+r-1}}^{t_{k+r}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, s)ds \tag{5}$$

Therefore, the goal is to calculate the right side of the equation 5. For this purpose, the function $\boldsymbol{f}$ is interpolated at the epochs $t_k, t_{k+1}, \ldots, t_{k+r}$ by a polynomial of degree r. Let be $\Psi_r(t)$ the Lagrange interpolating polynomial of degree r.

$$\Psi_r(t) = \sum_{j=0}^{r} \boldsymbol{f}(\hat{\boldsymbol{x}}_{k+j}, \boldsymbol{u}_{k+j}, t_{k+j})L_{k+j}(t)$$

where

$$L_{k+j}(t) = \prod_{\substack{i=0 \\ i\neq j}}^{r} \frac{t - t_{k+j}}{t_{k+i} - t_{k+j}}$$

Hence, isolating $\boldsymbol{x}_{k+r}$ and considering again that only approximations of the solution are known, it holds that

$$\hat{\boldsymbol{x}}_{k+r} = \hat{\boldsymbol{x}}_{k+r-1} + \sum_{j=0}^{r} \boldsymbol{f}(\hat{\boldsymbol{x}}_{k+j}, \boldsymbol{u}_{k+j}, t_{k+j}) \int_{t_{k+r-1}}^{t_{k+r}} L_{k+j}(s)ds \tag{6}$$

Assuming that $t_{i+1} - t_i = h, \forall i$ and by means of the change of variable $s = t_{k+r-1} + ht$, the integral of the right side can be calculated and it can be seen that

$$\hat{\boldsymbol{x}}_{k+r} = \hat{\boldsymbol{x}}_{k+r-1} + h \sum_{j=0}^{r} \beta_j \boldsymbol{f}(\hat{\boldsymbol{x}}_{k+j}, \boldsymbol{u}_{k+j}, t_{k+j}) \tag{7}$$

where

$$\beta_j = \int_0^1 \prod_{\substack{i=0 \\ i \neq j}}^{r} \frac{s - (r - j - 1)}{i - j} ds$$

The particular case where $\beta_r \neq 0$ and $\beta_0 = 0$ (not interpolating in the epoch t_k) leads to the well known *Adams-Moulton methods*. On the other hand, in the case where $\beta_r = 0$ and $\beta_0 \neq 0$ (not interpolating in the epoch t_{k+r}) one obtains the family of the *Adams-Bashforth methods*. As an example, the different values for β_i for the case $r = 4$ are summarized in table 1.

	Adams-Moulton	Adams-Bashforth
β_0	–	$\frac{55}{24}$
β_1	$\frac{9}{24}$	$-\frac{59}{24}$
β_2	$\frac{19}{24}$	$\frac{37}{24}$
β_3	$-\frac{5}{24}$	$-\frac{9}{24}$
β_4	$\frac{1}{24}$	–

Tab. 1: Values for the Adams-Moulton and Adams-Bashforth formulae for $r = 4$

Observe that in the case of the Adams-Moulton methods, the unknown appears both in the left and right sides of the equation 7. The methods with this characteristic are called *implicit* methods, in opposition to the *explicit* methods, where the unknown can be isolated in the left side of the equation. Euler, RK4 or Adams-Bashforth methods are examples of explicit methods.

Back to the Adams-Moulton method, this implicit problem might be solved by means of the following iterative scheme:

$$\hat{\boldsymbol{x}}_{k+r}^{(i+1)} = \hat{\boldsymbol{x}}_{k+r-1} + h\beta_r \boldsymbol{f}(\hat{\boldsymbol{x}}_{k+r}^{(i)}, \boldsymbol{u}_{k+r}, t_{k+r}) + h \sum_{j=1}^{r-1} \beta_j \boldsymbol{f}(\hat{\boldsymbol{x}}_{k+j}, \boldsymbol{u}_{k+j}, t_{k+j}) \tag{8}$$

for $i = 1, 2, \ldots$ Of course, a good initial approximation is essential for the convergence of the iterative scheme 8. The explicit Adams-Bashforth methods provide excellent means for estimating a first approximation of the state. The procedure to follow is, first, do a *prediction* step using the Adams-Bashforth explicit method in order to compute a seed for the iterative scheme corresponding to equation 8; afterwards, feed such equation with this estimation and *correct* it by means of an Adams-Moulton implicit method. This sort of procedures are the so-called *predictor-corrector methods*. One might think that the more you iterate equation 8, the best the approximation to the solution; nevertheless, generally one iteration is enough. See [3] for further details about this topic.

Now the philosophy behind the predictor-corrector methods has been explained, but from the way it has been described it suffers from the same drawbacks of the RK4 methods. This drawback is essentially that the integrating step-size is fixed. It has been already commented in previous sections the impact of such constraint. The goal now is go one step further and construct a method, based on the philosophy of the multi-step predictor-corrector methods but with independent step-size, in other words, from now on the assumption that $t_{k+1} - t_k = h, \forall k$ will not be required. The next section will explain how to construct such method.

It is worth to make a last remark about the exposed methods. From formula 7 one might be tempted to think that for an r^{th} order multistep method, r evaluations of the field are needed. However, a closer look at formula 7 shows that for calculating the solution at epoch t_{k+r}, actually just one evaluation is needed since the previous ones at epochs $t_{k+j}, j = 0, 1, \ldots, r-2$, in the case of the Adams-Bashforth methods and $j = 1, 2 \ldots, r-1$ for the Adams-Moulton's, were already used for estimating $\hat{\boldsymbol{x}}_{k+r-1}$. Hence, the there is no need of recalculating them (they might be stored in memory, for example), and only $\boldsymbol{f}(\hat{\boldsymbol{x}}_{k+r-1}, \boldsymbol{u}_{k+r-1}, t_{k+r-1})$ for the Adams-Bashforth case and $\boldsymbol{f}(\hat{\boldsymbol{x}}_{k+r}, \boldsymbol{u}_{k+r}, t_{k+r})$ for the Adams-Moulton need to be computed.

3.1 A fourth-order variable step-size multistep predictor-corrector method

Recall from the previous section that the construction of a multistep method essentially consisted in calculating the different values of $\beta_i, i = 0, 1, \ldots, r$. This values were the result of evaluating the integral written on the

right side of the equation 6. This calculus was done by means of the change of variable $s = t_{k+r-1} - ht$. This change of variable was constrained by a fixed integration step h; now, as said before, this constraint does not necessary hold. One of the strategies for integrating the polynomials $L_{k+j}(s)$ with a non-constant step-size h_k is by means of the change of variables defined by $t = s - t_{k+r-1}$. Note that since the integration step is not necessary constant, the recalculation of the β's at every epoch is unavoidable. Nevertheless, this does not imply a dramatic increasing of the computation time since the formulae are quite simple. The modified formulae for the Adams-Bashforth and Adams-Moulton methods are exposed below. Only the fourth order case ($r = 4$) is considered. Again, the goal is it to numerically solve the initial value problem 2 at epoch t_{k+1} provided that the solutions $\boldsymbol{x}_{k-3}, \boldsymbol{x}_{k-2}, \boldsymbol{x}_{k-1}, \boldsymbol{x}_k$ are known.

•Adams-Bashforth formulas
General formula:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h_k(\beta_{3k}\boldsymbol{f}_k + \beta_{2k}\boldsymbol{f}_{k-1} + \beta_{1k}\boldsymbol{f}_{k-2} + \beta_{0k}\boldsymbol{f}_{k-3})$$

The formulas for $\beta_{0k}, \beta_{1k}, \beta_{2k}$ and β_{3k} are,

$$\begin{aligned}
\beta_{0k} &= \frac{\zeta_k}{12h_{k-1}(h_{k-1}+h_{k-2})(h_{k-1}+h_{k-2}+h_{k-3})} \\
\beta_{1k} &= -\frac{\xi_k}{12h_{k-1}h_{k-2}(h_{k-2}+h_{k-3})}h_k \\
\beta_{2k} &= \frac{3h_k^2 + 4h_k(2h_{k-1}+h_{k-2}+h_{k-3}) + 6h_{k-1}(h_{k-1}+h_{k-2}+h_{k-3})}{12h_{k-2}h_{k-3}(h_{k-1}+h_{k-2})}h_k \\
\beta_{3k} &= -\frac{3h_k^2 + 4h_k(2h_{k-1}+h_{k-2}) + 6h_{k-1}(h_{k-1}+h_{k-2})}{12h_{k-3}(h_{k-2}+h_{k-3})(h_{k-1}+h_{k-2}+h_{k-3})}h_k
\end{aligned}$$

where

$$\begin{aligned}
\zeta_k &= 3h_k^3 + 4h_k^2(3h_{k-1}+2h_{k-2}+h_{k-3}) \\
&\quad +6h_k\Big(h_{k-1}(3h_{k-1}+4h_{k-2}+2h_{k-3}) + h_{k-2}(h_{k-2}+h_{k-3})\Big) \\
&\quad +12h_{k-1}\Big(h_{k-1}(h_{k-1}+2h_{k-2}+h_{k-3}) + h_{k-2}(h_{k-2}+h_{k-3})\Big) \\
\xi_k &= 3h_k^2 + 4h_k(2h_{k-1}+2h_{k-2}+h_{k-3}) \\
&\quad +6h_{k-1}(h_{k-1}+2h_{k-2}+h_{k-3}) \\
&\quad +6h_{k-2}(h_{k-2}+h_{k-3})
\end{aligned}$$

•Adams-Moulton formulas
General formula:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h_k(\beta_{4k}\boldsymbol{f}_{k+1} + \beta_{3k}\boldsymbol{f}_k + \beta_{2k}\boldsymbol{f}_{k-1} + \beta_{1k}\boldsymbol{f}_{k-2}) \tag{9}$$

The formulas for $\beta_{4k}, \beta_{3k}, \beta_{2k}$ and β_{1k} are,

$$\begin{aligned}
\beta_{1k} &= \frac{3h_k^2 + 4h_k(2h_{k-1}+h_{k-2}) + 6h_{k-1}(h_{k-1}+h_{k-2})}{12(h_k+h_{k-1})(h_k+h_{k-1}+h_{k-2})} \\
\beta_{2k} &= \frac{h_k^2 + 2h_k(2h_{k-1}+h_{k-2}) + 6h_{k-1}(h_{k-1}+h_{k-2})}{12h_{k-1}(h_{k-1}+h_{k-2})} \\
\beta_{3k} &= \frac{3h_k - 2(2h_k+h_{k-1}+h_{k-2})}{12h_{k-1}h_{k-2}(h_k+h_{k-1})}h_k^2 \\
\beta_{4k} &= \frac{h_k+2h_{k-1}}{12h_{k-2}(h_{k-1}+h_{k-2})(h_k+h_{k-1}+h_{k-2})}h_k^2
\end{aligned}$$

4 Applications to INS/GNSS integration

With the exception of strategic-grade IMUs, it is well known that an INS by itself does not perform well when estimating precise trajectories during long periods of time. Free inertial navigation is affected by large, time-dependent, time-growing errors; for this reason, the INS is usually aided with external measurements. Typically, this aid comes from the observations of one or two GNSS receivers in order to bound the errors and drifts of the IMU sensors and to calibrate them. The combined use of INS and GNSS technologies leads to a technology synergy widely known as INS/GNSS navigation. As the goal of this paper is not to discuss the performance details of the INS/GNSS systems, the reader is referred to [2] for further information. This section is devoted to the applications of the above presented numerical algorithm for the INS/GNSS integration. First, the reality

of the INS/GNSS will be presented and afterwards it will be seen that the proposed approach adapts to that reality and it overcomes the limitations of the present algorithms.

As mentioned in section 1, the classical approach to the INS/GNSS integration is the Kalman filter. For the sake of clarity, a brief description is outlined in the following lines. In the Kalman filter approach, the navigation solution is essentially calculated integrating equation 1 until a GNSS observation comes. When one inertial measurement and one GNSS observation occur at the same epoch, the navigator first estimates the state by means of an integration step of equation 1 and then, an optimal state is calculated by blending the predicted state and the GNSS observation in a Kalman filter. The interested reader in the Kalman filter approach to INS/GNSS integration and trajectory determination is again referred to [2] for further details.

The Kalman filter approach assumes that the clocks of the IMU and the GNSS receiver are synchronized and that one inertial measurement and one GNSS observation concur at the same epoch periodically. This theoretical hypothesis is depicted in figure 1. As seen in picture 1, the two data streams have constant time spans between two consecutive observations; moreover, there is an epoch at which IMU measurement and a GNSS observation coincide (at such epoch the Kalman filter step is performed). Let f_{IMU} and f_{GNSS} be the IMU frequency and GNSS receiver frequency respectively and t_k^{IMU} and t_k^{GNSS} the epochs of the IMU and GNSS receiver clocks respectively. In this ideal case it holds that $f_{IMU} = n \cdot f_{GNSS}$ for some $n \in \mathbb{N}$ and $t_k^{IMU} = t_k^{GNSS}$ for some k. The reality, however, is not that simple and, depending on the quality of the IMU's clock and the particular features of the IMU/GNSS assembly, several scenarios might occur. Figure 2 depicts the situation where the output data rate of the IMU is constant but the data stream is slightly shifted from the stream of the GNSS observations. In other words, $f_{IMU} = n \cdot f_{GNSS}$ for some $n \in \mathbb{N}$ and $t_k^{IMU} = t_k^{GNSS} + t_0$ for some k and fixed t_0. Going one step further, an even worse scenario can be considered, and this is the case where the frequency of the IMU, although being constant, is such that $f_{IMU} \neq n \cdot f_{GNSS}, \forall n \in \mathbb{N}$. This situation is depicted in figure 3. Finally, the worst scenario is the case where the $f_{IMU} \approx f$ and $t_k^{IMU} = t_k^{GNSS} + t_0$, where f is the nominal frequency of the IMU. Figure 4 shows this situation. Note that there are data gaps both in the IMU and GNSS data streams. This situation is quite common in the INS/GNSS integration and will be the one considered in this paper. (If the proposed approach succeeds in overcoming all the inconveniences of this scenario, it will also be able to deal with the other ones.) As mentioned before, depending on the quality of the IMU, the GNSS receiver and the data acquisition system, the consistency level of the inertial and GNSS data streams can be very different. The worst consistency level is usually found when integrating low-cost (typically automotive grade) IMU with low-cost GNSS receivers. Therefore, the proposed approach can be used in the currently emerging fields of automotive, pedestrian, lightweight UAV navigation, etc...

In other words, it is better to give up any assumption on data synchronization, specially for IMU observations. The classical solution to overcome these situations is to manipulate the data in order to adapt them to the algorithms. Three solutions might be considered:

- resample/interpolate all IMU observations (and [GNSS] measurements, if necessary) to the ideal situation.
- resample/interpolate [GNSS] measurements to IMU epochs.
- resample/interpolate IMU observations to [GNSS] measurements epochs.

Nevertheless, due to the noisy nature of the inertial observations, interpolation has to be considered a last resort and avoided if possible. It will seen that the proposed approach completely avoids the need of interpolating inertial (or external [GNSS]) measurements. The interpolation of inertial measurements is also needed when the inertial observations are not delivered at their nominal frequency due to the fact that the RK4 have constant step-size (Euler method is not considered due its poor accuracy). Remember that the proposed numerical method consisted in the combination of a prediction step (by means of an explicit method) with a correction step (an implicit method). When the navigation solution is estimated by means of the integration of the equation 1, a prediction-correction step is made at each epoch considering the integration step as the time span between two consecutive inertial measurements. These steps are depicted in figures 5 and 6. Assuming that the state at time t_{k+1} has to be estimated, and by means of the knowledge of the states $\hat{\boldsymbol{x}}_{k-3}, \hat{\boldsymbol{x}}_{k-2}, \hat{\boldsymbol{x}}_{k-1}$ and $\hat{\boldsymbol{x}}_k$, the Adams-Bashforth [prediction] step calculates a fourth-order approximation of the state $\hat{\boldsymbol{x}}_{k+1}^-$ (figure 5). Then, by means of an Adams-Moulton [correction] step, the previous states $\hat{\boldsymbol{x}}_{k-2}, \hat{\boldsymbol{x}}_{k-1}, \hat{\boldsymbol{x}}_k$ and the predicted state $\hat{\boldsymbol{x}}_{k+1}^-$, the state $\hat{\boldsymbol{x}}_{k+1}$ is estimated (figure 6).

The INS/GNSS navigation also benefits from the proposed approach to overcome the weak points of such technology. The weakness of the INS/GNSS navigation (from a computational point of view) is basically the [potential] poor synchronization between the the INS and GNSS receiver clocks. In case of not having time synchronization in such sense, difficulties to perform the Kalman filter step arise. Remember that the Kalman filter approach assumes an inertial observation and a GNSS observation to concur periodically at some epoch. This ideal situation rarely happens (at least in low-cost INS/GNSS systems). Actually, to do the Kalman filter step at a GNSS epoch, an inertial observation is not essential; what is essential is an estimation of the state at such epoch. In the absence of an inertial observation at the mentioned epoch, the state can be estimated by means of an Adams-Bashforth step. The integration step is set as the time span between the GNSS epoch and the

inertial measurement of the previous epoch. This situation is graphically represented in figure 7. This provides a fourth-order estimation of the solution, and therefore the Kalman filter can be performed. Nevertheless, once the Kalman filter step is done, the navigator has to continue integrating using the INS mechanization equations; hence, an inertial measurement at the GNSS epoch is required to initialize the predictor-corrector algorithm. Again, what is needed is not the inertial measurement but the evaluation of the field at the GNSS epoch. The strategy to estimate the field at that point is to make use of the Adams-Moulton method. Remember that the Adams-Moulton algorithm is an implicit method, this is, the unknown is both in the right and left side of the equation (see equation 9). However, what was the unknown (the state) for corrector is already known from the predictor. Furthermore, an optimal estimation has been calculated by means of a Kalman filter step. A closer look at equation 9 reveals that the evaluation of the field at the desired epoch can be easily isolated. Hence, isolating the unknown, the estimated evaluation of field at the desired epoch is:

$$\beta_{4k}\hat{\boldsymbol{f}}^{-}_{k+1} = \frac{\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_k}{h_k} - (\beta_{3k}\hat{\boldsymbol{f}}_k + \beta_{2k}\hat{\boldsymbol{f}}_{k-1} + \beta_{1k}\hat{\boldsymbol{f}}_{k-2})$$

where $\hat{\boldsymbol{f}}^{-}_{k+1}$ is the estimation of the field's value at the GNSS epoch. This procedure is depicted in the figure 8. This allows for an initialization of the predictor-corrector method and the equation 1 can be integrated until the next Kalman filter step.

Note that the fourth-order variable step-size multistep predictor-corrector method does not depend on a nominal frequency. In other terms, the proposed numerical integration scheme adapts itself to the realities of the INS/GNSS data streams, which seems to make more sense than adapting the data to the limitations of the algorithms.

5 Conclusions and ongoing work

The problem of the numerical integration of the INS mechanization equations has been presented. The classical methods —Euler, RK4 and transition matrix— have been outlined, their pitfalls exposed and an alternative to them described. The alternative is a fourth-order variable step-size multistep predictor-corrector method, derived from the classical Adams-Bashforth and Adams-Moulton formulae.

It has been also shown that the RK4 and the Euler methods are not well suited for the INS and INS/GNSS realities. Variable step-size predictor-corrector multistep methods are able to overcome this limitations in a fairly way. The described method preserves all the good properties of the Euler method and the RK4: it is of fourth order, can adapt its integration step to the inertial data rate and does not dramatically increases the computational burden. As said before, it adapts to the INS and GNSS realities, contrary to the classical approaches that adapt the reality to themselves. Further research includes the validation of the proposed numerical method.

Acknowledgements

The research reported in this paper has been funded by the Spanish Ministry of Science and Technology, through the OTEA-**g** project of the the Spanish National Space Research Programme (reference: ESP2002-03687).

References

[1] Engeln-Müllges, G., Uhlig, F., *"Numerical Algorithms with C"*, Spinger-Verlag, 1996.

[2] Jekeli, C., *"Inertial Navigation Systems with geodetic applications"*, Walter de Gruyter, 2001.

[3] Press, W.H. *et al.*, *"Numerical Recipies in C"*, Cambridge University Press, 1997.

[4] Maybeck, P.S., *"Stochastic Models, Estimation and Control. Volume 1"*, Mathematics in Science and Engineering, Volume 141-1, 1994.

[5] Stoer, J., Bulirsch, R., *"Introduction to Numerical Analysis"*, Spinger-Verlag, 1992.

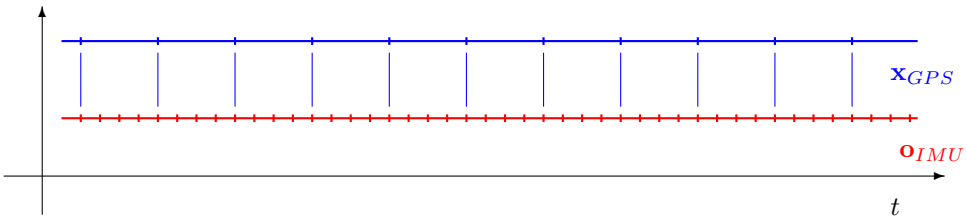

Fig. 1: Ideal IMU/GNSS scenario.

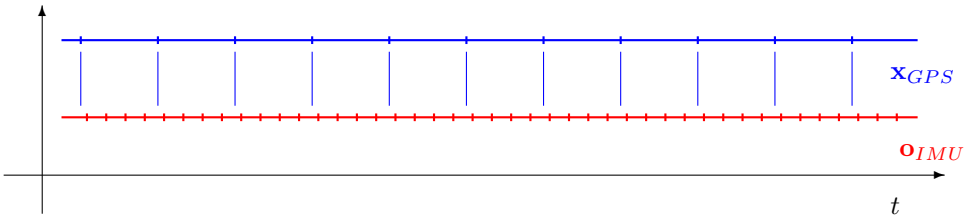

Fig. 2: Time shift between the IMU and GNSS data streams.

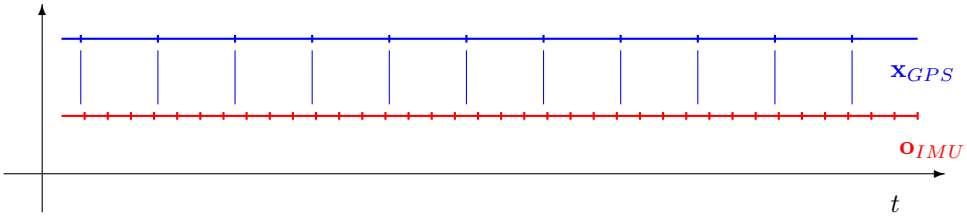

Fig. 3: IMU and GNSS data streams unsynchronizated by a scale factor and a time shift.

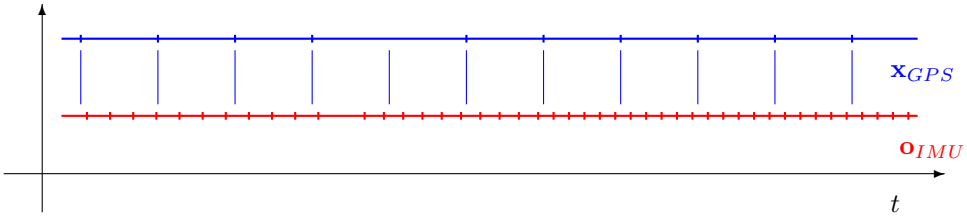

Fig. 4: Data streams with gaps and non-constant IMU frequency and data gaps.

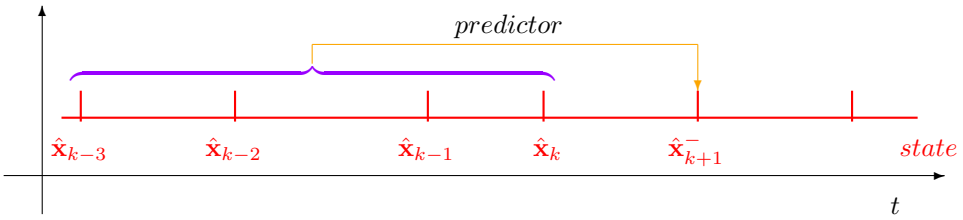

Fig. 5: INS integration. Predictor step.

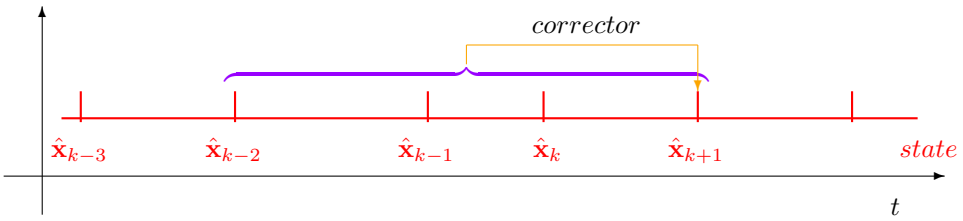

Fig. 6: INS integration. Corrector step.

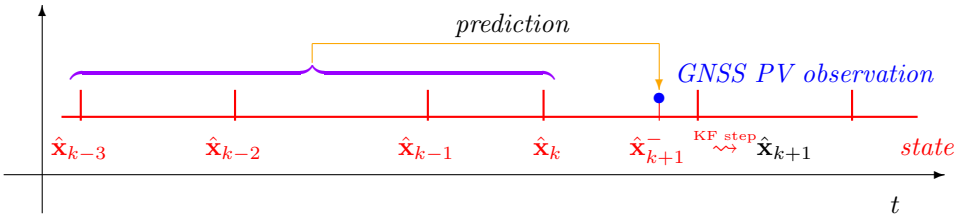

Fig. 7: INS/GNSS integration. Prediction and Kalman filter step.

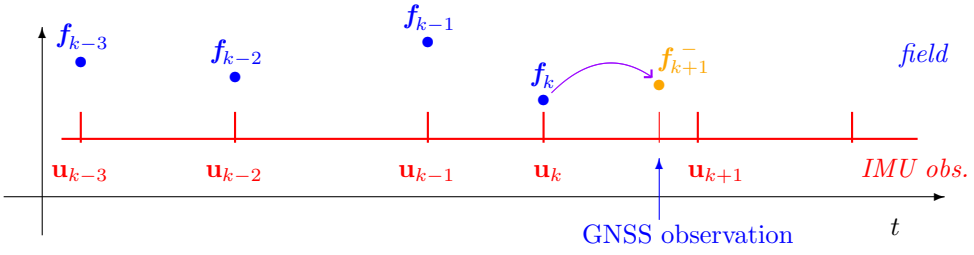

Fig. 8: INS/GNSS integration. Initialization after a Kalman filter step.