Abstract.
We consider the numerical integration of non-autonomous differential equations by solving appropriate modified differential equations where the modified vector field is a low order polynomial in time and a linear combination of the vector field evaluated at the nodes of a quadrature rule previously chosen. We show how to find modified vector fields (time-average vector fields) such that the solution of their modified equation agree with the solution up to the desired order. We also show how to build autonomous equations whose solution also agree with the solution of the problem up to a given order. Since the modified equations can be numerically solved by any one step scheme, this procedure allows to use a given numerical method where the vector field is evaluated at the nodes of a previously chosen quadrature rule, than can be advantageous in many cases. We illustrate the procedure in different examples.

Key words. Non-autonomous differential equations, time-averaged differential equations, quadrature rules, geometric integration.

AMS subject classifications. 65L05, 65J10

1. Introduction. Non-autonomous differential equations

\[ x' = f(t, x), \quad x(t_0) = x_0 \in \mathbb{C}^d, \quad (1.1) \]

\((t \equiv \frac{d}{dt})\) appear in most branches in science. The time-dependency of the vector field can be originated from many different sources (external interactions, when a preliminary change of coordinates is considered, when solving boundary value problems using iterative methods, etc.)

Direct application of most numerical integrators to (1.1) correspond to the application of the scheme to the autonomous equation obtained by appending \( t \) to the depending variables

\[ \begin{pmatrix} x \\ t \end{pmatrix}' = \begin{pmatrix} f(t, x) \\ 1 \end{pmatrix}. \quad (1.2) \]

It is obvious that for most problems the explicitly time dependency appearing in the right hand side of (1.1) plays an important role for the evolution of the system so, to take \( t \) as a depending variable playing the same role as any component of \( x \) is, in general, not the most appropriate procedure from the computational point of view.

In the linear case, \( f(t, x) = A(t)x \), with \( A \in \mathbb{C}^{N \times N} \), i.e.

\[ x' = A(t)x, \quad (1.3) \]
a frequently used technique to numerically integrate, for one step of length \( h \), is to frozen \( A(t) \) at the midpoint. This corresponds to solve the autonomous equation (for simplicity in the presentation we will take \( t_0 = 0 \) and we will only integrate for the first time step)

\[ y' = A\left(\frac{h}{2}\right)y, \quad y(0) = x_0, \quad (1.4) \]
or to average the value of $A(t)$ from its values at the beginning and at the end of the step

$$y' = \frac{1}{2} (A(0) + A(h)) y, \quad y(0) = x_0,$$

(1.5)
or, in general, to consider the autonomous equation

$$y' = A^{(0)} y, \quad y(0) = x_0,$$

(1.6)

where

$$A^{(0)} = \frac{1}{h} \int_0^h A(\tau) d\tau.$$

(1.7)

It is well known that their solutions correspond to second order approximations, so in all cases

$$\|x(h) - y(h)\| = O(h^3).$$

(1.8)

Suppose that $A(t) = B(t) + C(t)$ then, the exact solution of

$$y' = B(\frac{h}{2}) y + \frac{1}{2} (C(0) + C(h)) y, \quad y(0) = x_0,$$

(1.9)
at $t = h$ leads to a second order approximation, so different quadrature rules can be used on different time-dependent functions of the vector field. Finally, one can solve the autonomous equation (1.4), (1.5), (1.6) or (1.9) using a Runge-Kutta, Chebyshev, Krylov, exponential, etc. scheme. If the numerical method to solve the equations is of order two or higher, the numerical solution will be a second order approximation to $x(h)$.

This result also applies to solve (1.1) as follows: the solution of

$$y' = f(\frac{h}{2}, y), \quad y(0) = x_0,$$

(1.10)
at $t = h$ satisfies (1.8), but the solution of

$$y' = \frac{1}{2} (f(0, y) + f(h, y)), \quad y(0) = x_0,$$

(1.11)

and in general the solution of

$$y' = f^{(0)}_h (y), \quad y(0) = x_0,$$

(1.12)

where

$$f^{(0)}_h (y) = \frac{1}{h} \int_0^h f(\tau, y) d\tau$$

(1.13)

(the integration has to be carried out only on the explicitly time dependent functions of the vector field) also correspond to second order approximations to (1.1).

When accurate results are required, these second order schemes turn computationally very expensive since very small time steps have to be used.

This technique has been extended to higher orders for different problems: the linear case [1, 4, 6, 13, 14, 15, 18], general Lie group methods [20, 24], Hamiltonian systems [19] and the general non-linear case [7]. Most of these results were based on technically very involved proofs.
We present a simplified and constructive proof that allows to build time-average methods at any order that can be easily adapted to be used with most one step methods.

Given $s = 1, 2, 3, \ldots$ and the coefficients $c_i, i = 1, \ldots, \hat{s}$ of a quadrature rule of order $p \geq 2s$, we show how to build a vector field

$$\tilde{f}^{(2s)}_h(t, y) = f^{(2s)}_{h,0}(y) + \cdots + t^{p-1}f^{(2s)}_{h,s-1}(y)$$

where $f^{(2s)}_{h,i}(y)$ are linear combinations of $f(c_i h, y), i = 1, \ldots, \hat{s}$ such that the solution of the associated equation

$$y' = \tilde{f}^{(2s)}_h(t, y), \quad y(0) = x_0, \quad (1.14)$$

satisfies

$$\|x(h) - y(h)\| = O(h^{2s+1}). \quad (1.15)$$

Since linear combinations of the vector field at different instants preserve its algebraic structure, the solution $y(h)$ will retain most qualitative properties of the exact solution, $x(h)$, and we can refer to the resulting methods as geometric integrators [2, 9, 16, 21].

The case $s = 1$ corresponds to (1.6)-(1.13) when the integral is approximated using a quadrature rule with nodes $c_i, i = 1, \ldots, \hat{s}$ of order two or higher.

In particular, we show how to adapt most one step methods such that in each case the explicitly time-dependency in the vector field can be treated with any desired and convenient quadrature rule (or quadrature rules when the time-dependency appears at different places in the vector field) for each particular problem.

We also show how to obtain an associated autonomous equation

$$y' = \tilde{f}^{(2s)}_h(y), \quad y(0) = x_0, \quad (1.16)$$

where now $\tilde{f}^{(2s)}_h(y)$ is a more involved non-linear autonomous function of $f(c_i h, y)$ and its derivatives that also preserve the algebraic structure of the vector field, $f(t, x)$, and such that the solution of (1.16) at $t = h$ also satisfies (1.15).

We will also show how to build a sequence of autonomous equations such that the solution of the last equation provides an approximation to the solution. For example, for two equations it takes the form

$$z' = \tilde{f}^{(2s)}_{h,1}(z), \quad z(0) = x_0, \quad (1.17)$$

$$y' = \tilde{f}^{(2s)}_{h,2}(y), \quad y(0) = z(h), \quad (1.18)$$

where $\tilde{f}^{(2s)}_{h,1}(z), \tilde{f}^{(2s)}_{h,2}(y)$ are vector fields with a simpler structure than $\tilde{f}^{(2s)}_h(z)$. This procedure can be useful when solving each of the equations (1.17), (1.18) is simpler and/or faster than to solve (1.16) or can provide more accurate and stable results (and such that (1.15) is also satisfied).

2. Collocation methods and graded Lie algebras. In order to make the presentation self contained, we collect some well known results from the literature.

Modified equation. Given the modified differential equation

$$y' = f(t, y) + h^p f_{p+1}(t, y) + h^{p+1} f_{p+2}(t, y) + \cdots, \quad y(0) = x_0$$

with $f$ given in (1.1), the solution at $t = h$ satisfies

$$y(h) = x(h) + O(h^{p+1}). \quad (2.1)$$
Theorem 2.1 (Backward error analysis [9]). Suppose that the method \( x_1 = \Phi_h(x_0) \) is of order \( p \), i.e.,

\[
\Phi_h(x) = \psi_h(x) + h^{p+1} \delta_{p+1} + \mathcal{O}(h^{p+2}),
\]

where \( \psi_h(x) \) denotes the exact flow of \( x' = f(t, x) \), and \( h^{p+1} \delta_{p+1} \) is the leading term of the local truncation error. The modified equation such that \( \tilde{x}(h) = x_1 = x(h) + \mathcal{O}(h^{p+1}) \) satisfies

\[
\tilde{x}' = f(t, \tilde{x}) + h^p f_{p+1}(t, \tilde{x}) + h^{p+1} f_{p+2}(t, \tilde{x}) + \ldots
\]

with \( f_{p+1} = \delta_{p+1} \).

For simplicity in the presentation, we show first the results for the linear equation (1.3) and, as a simple corollary, we extend the results to the general case (1.1).

Let us consider \( c_i, i = 1, \ldots, s \), the \( s \) nodes of the Gauss quadrature of order 2


\[
a_{ij} = \int_0^{c_i} L_j(\tau) d\tau, \quad b_i = \int_0^1 L_i(\tau) d\tau
\]

(2.2)

where \( L_i(\tau) \) is the Lagrange polynomial

\[
L_i(\tau) = \prod_{j=1, j \neq i}^s \frac{\tau - c_j}{c_i - c_j}, \quad \tau \in [0, 1],
\]

(\( b_i \) are the weights of the quadrature rule).

One step of the scheme to solve (1.3) for \( t \in [0, h] \) is given by [10, 11]

\[
k_i = A(c_i h) \left( x_0 + h \sum_{j=1}^s a_{ij} k_j \right), \quad i = 1, \ldots, s
\]

\[
x_1 = x_0 + h \sum_{i=1}^s b_i k_i,
\]

(2.3)

and provides an approximate solution of order 2\( s \), i.e. \( x_1 = x(h) + \mathcal{O}(h^{2s+1}) \).

Then, by Theorem 2.1 the numerical solution \( x_1 \) coincides with the exact solution, at \( t = h \), of a perturbed differential equation given by

\[
\tilde{x}' = A(t) \tilde{x} + h^{2s} f_{2s+1}(t, \tilde{x}) + h^{2s+1} f_{2s+2}(t, \tilde{x}) + \ldots
\]

such that

\[
\tilde{x}(h) = x_1 = x(h) + \mathcal{O}(h^{2s+1}).
\]

(2.4)

Let us now consider the interpolating polynomial in \( t \) of degree \( s - 1 \) given by

\[
\tilde{A}_h^{(2s)}(t) = \sum_{i=1}^s L_i(\frac{t}{h}) A(c_i h)
\]

(2.5)

where

\[
\tilde{A}_h^{(2s)}(c_i h) = A(c_i h), \quad i = 1, \ldots, s,
\]

(2.6)

and the modified equation

\[
y' = \tilde{A}_h^{(2s)}(t)y, \quad y(0) = x_0.
\]

(2.7)
The collocation RK method applied to solve this equation provides exactly the same numerical result (2.3) so, it can be considered as the exact solution, at $t = h$, of another differential equation of the form
\[
\tilde{y}' = \tilde{A}_h^{(2s)}(t)\tilde{y} + h^2\tilde{f}_{2s+1}(t, \tilde{y}) + h^{2s+1}\tilde{f}_{2s+2}(t, \tilde{y}) + \ldots
\]  
(2.8)
As a result we have from (2.3), (2.7) and (2.8) that
\[
\tilde{y}(h) = x_1 = x(h) + \mathcal{O}(h^{2s+1})
\]  
(2.9)
so $\tilde{x}(h) = \tilde{y}(h)$ and then, from (2.4), (2.9) we have
\[
\|x(h) - y(h)\| = \|x(h) - x_1 + x_1 - y(h)\| = \|x(h) - \tilde{x}(h) + \tilde{y}(h) - y(h)\|
\leq \|x(h) - \tilde{x}(h)\| + \|\tilde{y}(h) - y(h)\| = \mathcal{O}(h^{2s+1}).
\]  
(2.10)
In other words, the exact solution of (2.7) at $t = h$ (or any numerical scheme to solve it to order $p \geq 2s$) can be seen as an approximation of order $2s$ to the original equation (1.3) at $t = h$.

The interpolating matrix polynomial $\tilde{A}_h^{(2s)}(t)$ is a polynomial function of degree $s - 1$ in $t$ and it is a linear combination of the matrices $A_i = A(c_i h), i = 1, \ldots, s$, that can be considered as the generators of a Lie algebra. Notice that [20]

\[ hA_i = \mathcal{O}(h), \quad hA_i - hA_j = \mathcal{O}(h^2), \quad \ldots, \quad \sum_{i=1}^s d_i hA_i = \mathcal{O}(h^s) \]
for appropriate choices of coefficients $d_i$. This result suggest to write $\tilde{A}_h^{(2s)}(t)$ in terms of a graded Lie algebra. For example, due to the time-symmetry of the solution and the symmetry of the nodes ($c_{s+1-i} = c_i, i = 1, 2, \ldots$) we suggest to write it as a Taylor expansion about the midpoint, i.e. to take $t = \frac{h}{2} + \tau$ and to take the Taylor expansion about $\tau = 0$ as follows

\[
\tilde{A}_h^{(2s)}(t) = \tilde{A}_h^{(2s)}\left(\frac{h}{2} + \tau\right) = \frac{1}{h^i} \sum_{i=1}^s \left(\frac{\tau}{h}\right)^{i-1} \alpha_i, \quad \tau \in \left[-\frac{h}{2}, \frac{h}{2}\right],
\]  
(2.11)
Notice that $\alpha_i, i = 1, \ldots, s$ are linear combinations of $A_j, j = 1, \ldots, s$, and $\alpha_i = \mathcal{O}(h^s), \alpha_i \alpha_j = \mathcal{O}(h^{i+j}), \text{etc.}$

We now define the moment integrals associated to $\tilde{A}_h^{(2s)}(t)$

\[
\tilde{A}_h^{(2s)}_{i,2s}(t) = \int_0^h \left(\frac{t}{h} - \frac{1}{2}\right)^i \tilde{A}_h^{(2s)}(t)dt = \int_{-h/2}^{h/2} \left(\frac{\tau}{h}\right)^i \tilde{A}_h^{(2s)}\left(\frac{h}{2} + \tau\right)d\tau.
\]  
(2.12)
Substituting (2.11) into (2.12) and evaluating the integrals (they are polynomial functions of $t$ or $\tau$) we have

\[
\tilde{A}_h^{(2s)}_{i,2s} = \sum_{j=1}^s \frac{1 - (-1)^{i+j}}{(i + j)^{2s+j+1}} \alpha_j \left(\sum_{j=1}^s \frac{(i + j)^{2s+j+1}}{T_{i,j}^{(2s)}} \alpha_j\right)
\]  
(2.13)
or, alternatively

\[
\alpha_i = \sum_{j=1}^s R_{ij}^{(2s)} \tilde{A}_h^{(2s)}_{i,2s}
\]  
(2.14)
where \( R^{(2s)} = (T^{(2s)})^{-1} \), and then

\[
\tilde{A}^{(2s)}_h(t) = \frac{1}{h} \sum_{i,j=1}^{s} \left( \frac{t - h/2}{h} \right)^{i-1} R_{ij}^{(s)} A^{(j)}_{h,2s}.
\]  

(2.15)

Taking into account that

\[
R^{(2)} = 1, \quad R^{(4)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 180 \end{pmatrix}, \quad R^{(6)} = \begin{pmatrix} 0 & 0 & -15 \\ 0 & 12 & 0 \\ -15 & 0 & 180 \end{pmatrix}
\]  

(2.16)

we have that for \( 2s = 2, 4, 6 \)

\[
\tilde{A}^{(2)}_h(\tau + h/2) = \frac{1}{h} \omega_1 = A^{(0)}_h, \\
\tilde{A}^{(4)}_h(\tau + h/2) = \frac{1}{h} \omega_1 + \frac{1}{h^2} \omega_2 \tau = A^{(0)}_h + \frac{12}{h^2} A^{(1)}_h \tau, \\
\tilde{A}^{(6)}_h(\tau + h/2) = \frac{1}{h} \omega_1 + \frac{1}{h^2} \omega_2 \tau + \frac{1}{h^3} \omega_3 \tau^2 \\
= \frac{1}{h} \left( 9 A^{(0)}_h - 15 A^{(2)}_{h,6} \right) + \frac{12}{h^2} A^{(1)}_h \tau + \frac{1}{h^3} \left( -15 A^{(0)}_{h,6} + 180 A^{(2)}_{h,6} \right) \tau^2,
\]  

(2.17, 2.18, 2.19)

\( \tau \in [-h/2, h/2] \), or to take \( \tau = t - h/2, \quad t \in [0, h] \). Then, a method of order \( 2s \) is obtained if we solve (2.7), where \( \tilde{A}^{(2s)}_h \) is given by (2.17), (2.18), (2.19) for \( 2s = 2, 4, 6 \), respectively.

Taking into account that \( b_i, c_i, i = 1, \ldots, s \) are the \( s \) weights and nodes of the Gauss quadrature rule of order \( 2s \) and that they provide the exact solution for polynomial functions of degree lower than \( 2s \) we have that (2.12) is given by (2.13), but also by

\[
\tilde{A}^{(i)}_{h,2s} = \sum_{j=1}^{s} b_j \left( c_j - \frac{1}{2} \right)^i A_j. 
\]  

(2.20)

Let us now define the moment matrices associated to \( A(t) \)

\[
A^{(i)}_{h,2s} = \int_0^h \left( \frac{t}{h} - \frac{1}{2} \right)^i A(t) dt.
\]  

(2.21)

If we evaluate these integrals using the Gaussian quadrature rule we have

\[
A^{(i)}_{h,2s} = \sum_{j=1}^{s} b_j \left( c_j - \frac{1}{2} \right)^i A_j + O(h^{2s+1}),
\]

and then

\[
\tilde{A}^{(i)}_{h,2s} = A^{(i)}_{h,2s} + O(h^{2s+1}).
\]

As a result, if in (2.15) we replace \( \tilde{A}^{(i)}_{h,2s} \) by \( A^{(i)}_{h,2s} \) and we denote by

\[
A^{(2s)}_h(t) = \frac{1}{h} \sum_{i,j=1}^{s} \left( \frac{t - h/2}{h} \right)^{i-1} R_{ij}^{(2s)} A^{(j)}_{h,2s}
\]

then, from Theorem 2.1, the exact solution at \( t = h \) of the equation

\[
z' = A^{(2s)}_h(t)z, \quad z(0) = x_0
\]  

(2.22)
satisfies \( z(h) = x(h) + O(h^{2s+1}) \). For \( 2s = 2, 4, 6 \) we have

\[
A_h^{(2)}(\tau + \frac{h}{2}) = \frac{1}{h} A_{h,2}^{(0)},
\]

(2.23)

\[
A_h^{(4)}(\tau + \frac{h}{2}) = \frac{1}{h} A_{h,4}^{(0)} + \frac{12}{h^2} A_{h,4}^{(1)} \tau,
\]

(2.24)

\[
A_h^{(6)}(\tau + \frac{h}{2}) = \frac{1}{h} \left( \frac{9}{4} A_{h,6}^{(0)} - 15 A_{h,6}^{(2)} \right) + \frac{12}{h^2} A_{h,6}^{(1)} \tau + \frac{1}{h^3} \left( -15 A_{h,6}^{(0)} + 180 A_{h,6}^{(2)} \right) \tau^2.
\]

(2.25)

Writing the equation in this form allows us to evaluate the moment matrices (2.21) using any quadrature rule of order \( 2s \) or higher e.g. given \( \{\hat{b}_i, \hat{c}_i\}_{i=1}^s \) the weights and nodes of a quadrature rule of order \( p \geq 2s \) then we have

\[
A_{h,2s}^{(i)} = \sum_{j=1}^s b_j \left( \hat{c}_j - \frac{1}{2} \right) \hat{A}_j + O(h^{p+1}),
\]

where \( \hat{A}_j = A(\hat{c}_j h) \). If we replace \( A_{h,2s}^{(i)} \) by \( \hat{A}_{h,2s}^{(i)} \) in (2.22) and solve the resulting equation with an integrator of order \( 2s \) or higher the final approximation will remain of order \( 2s \). Notice in addition that each component of the matrix \( A(t) \), say \( A_{i,j}(t) \), \( i, j = 1, \ldots, d \), can be computed using different quadrature rules.

2.1. The non-linear case. In a similar way as for the linear case we get the following result:

**Theorem 2.2.** Given \( x(h) \), the solution at \( t = h \) of (1.1), the solution of the differential equation

\[
y'(t, y) = \hat{f}_h^{(2s)}(t, y), \quad y(0) = x_0,
\]

(2.26)

with

\[
\hat{f}_h^{(2s)}(t, y) = \sum_{i=1}^s L_i \left( \frac{t}{h} \right) f(c_i h, y)
\]

(2.27)

where \( c_i \) are the nodes of the Gaussian quadrature rule of order \( 2s \), satisfies that

\[
\| x(h) - y(h) \| = O(h^{2s+1}).
\]

The proof is identical to the linear case, the RKGL collocation method of order \( 2s \) to solve (1.1) is given by

\[
k_i = f(c_i h, x_0 + h \sum_{j=1}^s a_{ij} k_j), \quad i = 1, \ldots, s
\]

\[
x_1 = x_0 + h \sum_{i=1}^s b_i k_i.
\]

(2.28)

From the definition (2.27) we have that

\[
\hat{f}_h^{(2s)}(c_i h, y) = f(c_i h, y), \quad i = 1, \ldots, s
\]

so the numerical scheme (2.28) applied to solve (2.26) provides exactly the same numerical solution and a similar inequality to (2.10) can be obtained.
We can define graded vector fields similarly to \( \alpha_i \) for the linear case and write the equation (2.26) in terms of one-dimensional moment integrals over the explicitly time dependent functions of the vector field, i.e. in terms of

\[
\int_0^h \left( \frac{t}{h} - \frac{1}{2} \right)^i f(t,y) dt,
\]

(2.29)
i.e. the exact solution of

\[
z' = f_h^{(2s)}(t,z), \quad z(0) = x_0
\]

(2.30)
where for \( 2s = 2, 4, 6 \)

\[
f_h^{(2)}(\tau + \frac{h}{2}, z) = \frac{1}{h} f_h^{(0)}
\]

(2.31)
\[
f_h^{(4)}(\tau + \frac{h}{2}, z) = \frac{1}{h} f_h^{(0)} + \frac{12}{h^2} f_h^{(1)} \tau
\]

(2.32)
\[
f_h^{(6)}(\tau + \frac{h}{2}, z) = \frac{1}{h} \left[ \frac{9}{4} f_h^{(0)} - 15 f_h^{(2)} \right] + \frac{12}{h^2} f_h^{(1)} \tau + \frac{1}{h^3} \left( -15 f_h^{(0)} + 180 f_h^{(2)} \right) \tau^2
\]

(2.33)
satisfies that

\[\|x(h) - z(h)\| = \mathcal{O}(h^{2s+1}).\]

**Example 1.** The exact solution of the equation

\[
z' = \frac{1}{2} (f_1(z) + f_2(z)) + \frac{\sqrt{2}}{h} (f_2(z) - f_1(z)) \left( t - \frac{h}{2} \right)
\]

where \( f_i(z) = f(c_i h, z), i = 1, 2 \) and \( c_1, c_2 \) the nodes of the 4th-order GL quadrature rule, agrees with the solution of (1.1) at \( t = h \) up to order four. Notice that a linear combinations of the vector field at different intants preserve its algebraic structure and then, the exact solution, \( z(t) \) will preserve most qualitative properties of the exact solution \( x(t) \).

3. To construct associated autonomous equations. Let us see how to build autonomous equations where the exact solution at \( t = h \) coincides with the desired solution at a given order while preserving most qualitative properties of the exact solution.

As in the previous section, we first consider the linear case (1.3) that has been extensively studied in the literature. As we have seen, up to order \( 2s \) this is equivalent to solve (2.22) with \( A_h^{(2s)} \) given by (2.23) for \( 2s = 2 \), by (2.24) for \( 2s = 4 \), etc.

3.1. The Magnus expansion. The Magnus expansion [17] expresses the solution to (1.3), \( x(h) = \Phi(h)x_0 \), as a single exponential

\[
\Phi(h) = \exp \left( \Omega(h) \right), \quad \Omega(h) = \sum_{k=1}^{\infty} \Omega_k(h),
\]

(3.1)

where the first terms of the Magnus series \( \{\Omega_k\} \) are given by

\[
\Omega_1(h) = \int_0^h A(\tau_1) d\tau_1, \quad \Omega_2(h) = \frac{1}{2} \int_0^h \int_0^{\tau_1} [A(\tau_1), A(\tau_2)] d\tau_2 d\tau_1, \ldots
\]

(3.2)
\( [P, Q] = PQ - QP \) is the matrix commutator of \( P \) and \( Q \). Backward error analysis tells us that the solution at \( t = h \) coincides with the solution at \( t = h \) of

\[
y' = \frac{1}{h} \Omega(h) y, \quad (3.3)
\]

for sufficiently small \( h \) to guarantee the convergence of the series.

Since (2.22) is an approximation to order \( 2s \) to the solution \( x(h) \) and \( \tilde{A}^{(2s)}(t) \) is a polynomial function in \( t \) of degree \( s - 1 \), we can solve (2.22) using the Magnus expansion where the integrals can be exactly evaluated.

To take the necessary number of terms to truncate the Magnus series to obtain an approximation to order \( 2s \) we consider the following property [4, 15, 20] (see also [3] and references therein): each term in \( \Omega_k \) is an odd function of \( h \) and, in particular

\[
\Omega_1 = O(h), \quad \Omega_2 = O(h^3), \quad \Omega_{2m-1}, \Omega_{2m} = O(h^{2m+1}), \quad m > 1.
\]

Let us denote by \( \exp(\tilde{\Omega}(2s)) \) the fundamental matrix solution of

\[
y' = h^{-1} \Omega(h) y;
\]

Then, a second order method is obtained taking \( 2s = 2 \) and

\[
\tilde{\Omega}^{(2)} = \alpha_1 = \tilde{A}^{(0)}_{h,2}.
\]

A fourth order method is obtained taking \( 2s = 4 \) and [4]

\[
\tilde{\Omega}^{(4)} = \tilde{\Omega}^{(4)}_1 + \tilde{\Omega}^{(4)}_2 = \alpha_1 - \frac{1}{12} [\alpha_1, \alpha_2] + O(h^5) = \tilde{A}^{(0)}_{h,4} + [\tilde{A}^{(1)}_{h,4}, \tilde{A}^{(0)}_{h,4}] + O(h^5).
\]

A sixth order method is obtained taking \( \tilde{\Omega}^{(6)}_1, \ldots, \tilde{\Omega}^{(6)}_4 \) and \( \tilde{A}^{(6)}_h \) in (2.19). The algebra turns now more involved and we find simpler to consider \( \tilde{A}^{(6)}_h \) in terms of the graded elements \( \alpha_1, \alpha_2, \alpha_3 \) to obtain

\[
\begin{align*}
\tilde{\Omega}^{(6)}_1 &= \alpha_1 + \frac{1}{12} \alpha_3 \\
\tilde{\Omega}^{(6)}_2 &= -\frac{1}{12} [12] + \frac{1}{240} [23] \\
\tilde{\Omega}^{(6)}_3 &= \frac{1}{360} [113] - \frac{1}{240} [212] + O(h^7) \\
\tilde{\Omega}^{(6)}_4 &= \frac{1}{720} [1112] + O(h^7)
\end{align*}
\]

so

\[
\tilde{\Omega}^{(6)} = \alpha_1 + \frac{1}{12} [12] + \frac{1}{240} [23] + \frac{1}{360} [113] - \frac{1}{240} [212] + \frac{1}{720} [1112] + O(h^7)
\]

where \( [ij \ldots kl] \) represents the nested commutator \([\alpha_i, [\alpha_j, \ldots, [\alpha_k, \alpha_l]]]]\). Finally, if we replace in the previous equations \( \tilde{A}^{(j)}_{h,2} \) by \( A^{(j)}_{h,2} \), \( j = 0, \ldots, s - 1 \) we still have a scheme of order \( 2s \) so

\[
z' = A^{(0)}_{h,2} z
\]

provides a second order approximation as already known,

\[
z' = \left( A^{(0)}_{h,4} + [A^{(1)}_{h,4}, A^{(0)}_{h,4}] \right) z
\]

provides a fourth order approximation, etc.

In some cases the numerical solution of (3.3) turns very involved or computationally costly due to the presence of commutators. In these cases it can be preferable
to approximate the solution as a sequence of simpler equations. For example, to fourth-order, if we take into account that

\[ e^{\frac{1}{2} \alpha_1 + \frac{1}{4} \alpha_2} e^{\frac{1}{2} \alpha_1 - \frac{1}{4} \alpha_2} = e^{\alpha_1 - \frac{1}{4} [\alpha_1, \alpha_2]} + O(h^5) \]

or equivalently, in terms of \( \tilde{A}^{(j)} \), \( j = 0, 1 \)

\[ e^{\frac{1}{2} \tilde{A}^{(0)} + 2 \tilde{A}^{(1)}} e^{\frac{1}{2} \tilde{A}^{(0)} - 2 \tilde{A}^{(1)}} = e^\Omega + O(h^5) \]

then the sequence of equations, if we replace in the previous equations \( \tilde{A}^{(j)} \) by \( A^{(j)} \)

\[
\begin{align*}
    z' &= \frac{1}{h} \left( \frac{1}{2} A_{h,4}^{(0)} - 2 A_{h,4}^{(1)} \right) z, \quad z(0) = x_0, \\
y' &= \frac{1}{h} \left( \frac{1}{2} A_{h,4}^{(0)} + 2 A_{h,4}^{(1)} \right) y, \quad y(0) = z(h),
\end{align*}
\]

provide a 4th-order scheme. If we use the 4th-order Gauss quadrature to evaluate the integrals we get the method [6, 23]

\[
\begin{align*}
    z' &= (a_1 A_1 + a_2 A_2) z, \quad z(0) = x_0, \quad (3.4) \\
y' &= (a_2 A_1 + a_1 A_2) y, \quad y(0) = z(h). \quad (3.5)
\end{align*}
\]

with

\[
a_1 = \frac{1}{4} - \frac{\sqrt{3}}{6}, \quad a_2 = \frac{1}{4} + \frac{\sqrt{3}}{6}. \quad (3.6)
\]

or equivalently the method is given by

\[
x_1 = e^{h(a_2 A_1 + a_1 A_2)} e^{h(a_1 A_1 + a_2 A_2)} x_0.
\]

Since \( a_1 + a_2 = \frac{1}{2} \), roughly we advance half step with the variable \( z \) and we continue the second half step with the variable \( y \). This class of methods is usually referred as commutator-free exponential or Magnus integrators due to the close relation to exponential integrators when the solution of both equations is written in exponential form or to the Magnus expansion since the composition is usually deduced taking the Magnus expansion as the formal solution. Optimized fourth-order schemes as well as higher order methods are considered in [1, 5, 8].

In general we have the sequence

\[
z[i]' = \left( \sum_{j=1}^{s} a_{i,j} A_j \right) z[i], \quad z[i](0) = z[i-1](h), \quad (3.7)
\]

\( i = 1, \ldots, k, \) with \( z[0](h) = x_0 \), and such that \( z[k](h) = x(h) + O(h^{2s+1}) \).

**The Magnus expansion for non-linear non-autonomous differential equations.**

The Magnus expansion was originally built to solve the linear non-autonomous equation, but the expansion can be generalised to nonlinear problems as follows. Given the vector fields, \( f(x), g(x) \), we denote by \( (f, g) \) their Lie bracket, i.e.

\[
(f, g)_i = \sum_{j=1}^{d} \left( f_j \frac{\partial g_i}{\partial x_j} - g_j \frac{\partial f_i}{\partial x_j} \right).
\]

Then, we notice the following property: given the linear vector fields \( f(x) = Fx, \ g(x) = Gx \) their Lie bracket is

\[
(f, g) = [G, F] x.
\]

We observe that the vector fields in the Lie bracket appear in the reverse order with respect to the commutator of their associated matrices\(^1\).

\(^1\)See the appendix of [2] (and references therein) for a review on the Lie algebra of vector fields, the Lie algebra of the associated Lie operators and the order in which they appear.
If we take
\[ f(t, x) = A(t) x \] (3.8)
we can write the autonomous equation (3.3) in terms of the vector field \( f(t, x) \),
given by (3.8) as follows
\[ y' = \frac{1}{h} f_\Omega(h, y) \] (3.9)
where
\[ f_\Omega(h, y) = \int_0^h f(\tau_1, y) \, d\tau_1 + \frac{1}{2} \int_0^h \int_0^{\tau_1} \left( f(\tau_2, y), f(\tau_1, y) \right) \, d\tau_2 \, d\tau_1 + \ldots \] (3.10)
Notice the commutators have been replaced by Lie brackets, but in the reverse order. Similarly,
\[ y' = \frac{1}{h} f^{(0)}_h(y) \] (3.11)
leads to a second order approximation, a fourth order approximation is given by
the exact solution of
\[ y' = \frac{1}{h} \left( f^{(0)}_h(y) + (f^{(0)}_h(y), f^{(1)}_h(y)) \right) \] (3.12)
etc., and the sequence of equations
\[ z' = \frac{1}{h} \left( \frac{1}{2} f^{(0)}_h(z) - 2 f^{(1)}_h(z) \right), \quad z(0) = x_0, \] (3.13)
\[ y' = \frac{1}{h} \left( \frac{1}{2} f^{(0)}_h(y) + 2 f^{(1)}_h(y) \right), \quad y(0) = z(h), \] (3.14)
provides a 4th-order scheme.

The equations (3.9)-(3.14) are obtained for the particular case in which \( f(t, x) \) is the linear vector field given by (3.8). However, the following theorem proves that the schemes remain valid for any non-linear vector field.

**Theorem 3.1.** Given the general non-autonomous equation
\[ x' = f(t, x), \quad x(0) = x_0, \] (3.15)
the solution at \( t = h \), \( x(h) \), coincides with the solution of (3.8) at \( t = h \), \( y(h) \), where \( f_\Omega(h, y) \) is given by the series expansion (3.10) in the region where it converges.

In general, the series (3.10) diverges for non-linear vector fields, but the truncated series gives us approximations to different orders, and it provides us a tool to obtain methods where the time-dependency in the vector field is averaged.

The proof of the theorem is left to the appendix where, for the convenience of the reader, we previously review some basic concepts of Lie operators [2].

**4. Examples.** The goal of this section is to show how to use the time-average techniques presented in this work in order to adapt different one-step methods involving the vector field evaluated at a given quadrature rule.

*The steps to build time-averaged methods.*
- To choose the order of the method, \( 2s \).
- To consider the graded Lie algebra with generators, \( \alpha_1, \ldots, \alpha_s \), and to write the equation in terms of them, and whose exact solution corresponds to an approximate solution up to order \( 2s \).
• to write $\alpha_i$, $i = 1, \ldots, s$ in terms of $\tilde{A}^{(j)}_{h,2s}$, $j = 1, \ldots, s$ or $\tilde{f}^{(j)}_{h,2s}(y)$, $j = 1, \ldots, s$ for non-linear problems (see (2.17)-(2.19) for $2s = 2, 4, 6$).
• To replace $\tilde{A}^{(j)}_{h,2s}$ by $A^{(j)}_{h,2s}$, $j = 1, \ldots, s$.
• To use the desired quadrature rule or quadrature rules to compute $A^{(j)}_{h,2s}$ (different quadrature rules can be used for different time-dependent functions appearing in $A(t)$ or $f(t,x)$).
• To choose the most appropriate method to solve the modified equation.

For convenience of the reader, we write the exact values of $\tilde{A}^{(i)}_{h,2s}$, $i = 0, \ldots, s - 1$ to obtain methods of order $p = 2s = 4, 6$ using both the Gauss-Legendre $\{b_i, c_i\}_{i=1}^{s}$ (GL$p$) and the Newton-Cotes quadrature rules $\{b_i, c_i\}_{i=1}^{s}$, with $s = 3, 5$ (NC$p$), respectively:

**GL4:**
\[
(b_1, b_2) = \frac{1}{2}(1, 1), \quad (c_1, c_2) = \left(\frac{1}{2}, \frac{\sqrt{3}}{6}, \frac{1}{2} + \frac{\sqrt{3}}{6}\right).
\]

**NC4:**
\[
(b_1, b_2, b_3) = \frac{1}{6}(1, 4, 1), \quad (c_1, c_2, c_3) = \frac{1}{2}(0, 1, 2).
\]

**GL6:**
\[
(b_1, b_2, b_3) = \frac{1}{18}(5, 8, 5), \quad (c_1, c_2, c_3) = \left(\frac{1}{2}, \frac{\sqrt{15}}{10}, \frac{1}{2} + \frac{\sqrt{15}}{10}\right).
\]

**NC6:**
\[
(b_1, b_2, b_3, b_4, b_5) = \frac{1}{90}(7, 32, 12, 32, 7), \quad (c_1, c_2, c_3, c_4, c_5) = \frac{1}{4}(0, 1, 2, 3, 4)
\]

It is then immediate to check that using the GL4 coefficients
\[
\tilde{A}^{(0)}_{h,4} = \frac{h}{2}(A_1 + A_2), \quad \tilde{A}^{(1)}_{h,4} = \frac{h\sqrt{3}}{12}(A_2 - A_1)
\]
while using the NC4 coefficients we have
\[
\tilde{A}^{(0)}_{h,4} = \frac{h}{6}(\hat{A}_1 + 4\hat{A}_2 + \hat{A}_3), \quad \tilde{A}^{(1)}_{h,4} = \frac{h}{12}(\hat{A}_3 - \hat{A}_1).
\]

To build sixth-order methods using the GL6 coefficients we have
\[
\tilde{A}^{(0)}_{h,6} = \frac{h}{18}(5A_1 + 8A_2 + 5A_3) \quad (4.1)
\]
\[
\tilde{A}^{(1)}_{h,6} = \frac{h\sqrt{15}}{36}(A_3 - A_1) \quad (4.2)
\]
\[
\tilde{A}^{(2)}_{h,6} = \frac{h}{24}(A_3 + A_1) \quad (4.3)
\]
while using the NC6 coefficients we have
\[
\tilde{A}^{(0)}_{h,6} = \frac{h}{90}(7\hat{A}_1 + 32\hat{A}_2 + 12\hat{A}_3 + 32\hat{A}_4 + 7\hat{A}_5), \quad (4.4)
\]
\[
\tilde{A}^{(1)}_{h,6} = \frac{h}{90}\left(\frac{7}{4}(\hat{A}_5 - \hat{A}_1) + 8(\hat{A}_4 - \hat{A}_2)\right) \quad (4.5)
\]
\[
\tilde{A}^{(2)}_{h,6} = \frac{h}{90}\left(\frac{7}{4}(\hat{A}_1 + \hat{A}_5) + 2(\hat{A}_2 + \hat{A}_4)\right) \quad (4.6)
\]
where $A_i = A(c_i h)$, $\hat{A}_j = A(\hat{c}_j h)$ for the corresponding rules.

Obviously, similar expressions have to be used for non-linear problems.

**Example 2.** To numerically solve the linear system (1.3) using the standard 4-stage 4th-order explicit RK (RK4) method and the 4th-order implicit RK Gauss-Legendre (RKGL4) method. Apply the averaging technique in each case using the following quadrature rules to evaluate the matrix $A(t)$: GL4, GL6, NC4 and NC6.

In all cases, the equation to solve corresponds to

$$y'(t) = (B_0 + B_1 \tau)y, \quad y(0) = x_0$$

where

- **GL4**: $B_0 = \frac{1}{2} (A_1 + A_2)$, $B_1 = \frac{\sqrt{3}}{h} (A_2 - A_1)$.
- **GL6**: $B_0 = \frac{1}{18} (5A_1 + 8A_2 + 5A_3)$, $B_1 = \frac{\sqrt{15}}{36} (A_3 - A_1)$.
- **NC4**: $B_0 = \frac{1}{6} (A_1 + 4A_2 + A_3)$, $B_1 = \frac{1}{h} (A_3 - A_1)$.
- **NC6**: $B_0 = \frac{1}{90} (7A_1 + 32A_2 + 12A_3 + 32A_4 + 7A_5)$, $B_1 = \frac{1}{90} \left( \frac{7}{4} (A_5 - A_1) + 8 (A_4 - A_2) \right)$.

where by $A_i$ we denote the matrix evaluated at the nodes of the corresponding quadrature rule. Taking into account that $\tau = t - h = 2$ we have that the RK4 method is given by

$$\begin{align*}
k_1 &= (B_0 - \frac{h}{2} B_1)x_0 \\
k_2 &= B_0 \left( x_0 + \frac{h}{2} k_1 \right) \\
k_3 &= B_0 \left( x_0 + \frac{h}{2} k_2 \right) \\
k_4 &= (B_0 + \frac{h}{2} B_1) \left( x_0 + hk_3 \right)
\end{align*}$$

$$y_1 = x_0 + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4),$$

while the RKGL4 is given by

$$\begin{align*}
k_1 &= (B_0 - h \frac{\sqrt{3}}{6} B_1) \left( x_0 + h \left( \frac{1}{4} k_1 + \frac{3 - 2\sqrt{3}}{12} k_2 \right) \right) \\
k_2 &= (B_0 + h \frac{\sqrt{3}}{6} B_1) \left( x_0 + h \left( \frac{3 + 2\sqrt{3}}{12} k_1 + \frac{1}{4} k_2 \right) \right)
\end{align*}$$

$$y_1 = x_0 + \frac{h}{2} (k_1 + k_2).$$

We observe that in all cases the resulting methods are different to the results obtained if the schemes were used directly to solve (1.3). Notice that the methods are of order four even in the case when higher order quadrature rules are used. The stability of the methods as well as their relative performance will then depend on the particular problem they are used.

**Example 3.** Let us consider the separable non-autonomous Hamiltonian

$$H(q,p,t) = T(p,t) + V(q,t),$$

and $\{a_i, b_i\}_{i=1}^m$ the coefficients of a splitting integrator of order $p$ such that

$$\prod_{i=1}^m e^{ba_i X} e^{hb_i Y} = e^{h(X+Y)} + O(h^{p+1}),$$
with $X, Y$ two non-commuting matrices. To solve the non-autonomous problem is equivalent to solve the following autonomous one in the extended phase space where two new coordinates and associated momenta are considered [22]

$$\tilde{H}(q,p,t) = T(p,p_{t,2}) + p_{t,1} + V(q,q_{t,1}) - q_{t,2},$$

that can be solved using the splitting (and symplectic when applied to this Hamiltonian problem) integrator as follows

$$\prod_{i=1}^{m} e^{hV(d_i,h)} e^{ha_i T(e_i,h)}$$

with $d_i = \sum_{j=1}^{i} a_j$, $e_i = \sum_{j=0}^{i-1} b_j$, $b_0 = 0$ and $e^{hT(e_i,h)}$ denotes the map that advances the kinetic part $T(p,t)$ with $t$ frozen at $t = e_i h$ and similarly for the potential part, i.e.

$$\text{do } i = 1, m$$

$$q_i = q_{i-1} + h a_i T_p(p_{i-1}, e_i h)$$

$$p_i = p_{i-1} - h b_i V_q(q_i, d_i h)$$

(4.9)

enddo

with $T_p \equiv \frac{\partial T}{\partial p}$, $V_q \equiv \frac{\partial V}{\partial q}$.

Splitting methods of order greater than two necessarily involve negative coefficients, $a_i, b_i$, and it can happen that some coefficients $d_i$ or $e_i$ do not belong to the interval $[0,1]$, and the time-dependent functions in $T$ and $V$ could be not well defined. There are also splitting methods with complex coefficients (pseudosymplectic integrators) and the evaluations of $T, V$ at complex times could be problematic. In addition, these schemes require $m$ different evaluations of such functions.

The schemes we propose require the evaluation of $T$ and $V$ at the nodes one chooses in advance. For example, a 4th-order scheme that uses the GL4 coefficients to evaluate $T(q,t)$ and the NC4 coefficients to evaluate $V(q,t)$ is given by

$$\text{do } i = 1, m$$

$$q_i = q_{i-1} + h a_i T_p^{(4)}(p_{i-1}, e_i h)$$

$$p_i = p_{i-1} - h b_i V_q^{(4)}(q_i, d_i h)$$

(4.10)

enddo

where

$$T_p^{(4)}(p,t) = \frac{1}{2} (T_1(p) + T_2(p)) + \sqrt{3}(T_3(p) - T_1(p)) \left( \frac{t}{h} - \frac{1}{2} \right)$$

$$V_q^{(4)}(q,t) = \frac{1}{6} (V_1(q) + 4V_2(q) + V_3(q)) + (V_5(q) - V_1(q)) \left( \frac{t}{h} - \frac{1}{2} \right)$$

with $T_i(p) = T(p, e_i h)$, $i = 1, 2$, and $V_j(p) = V(p, e_j h)$, $j = 1, 2, 3$.

Example 4. Let us consider the Lotka-Volterra problem with a time-dependent equilibrium point given by $(u,v > 0)$

$$u' = u(v - r(t)), \quad v' = v(s(t) - u)$$

with $r(t), s(t)$ positive definite functions. This problem is equivalent to the non-autonomous Hamiltonian problem with Hamiltonian

$$H(q,p,t) = e^p - r(t)p + e^q - s(t)q$$
\[(q = \log u, p = \log v) \text{. The algorithm (4.11) applied to this problem corresponds to} \]

\[
\begin{aligned}
do & \ i = 1, m \\
q_i &= q_{i-1} + h a_i (e^{p_i-1} - r^{(4)}(c_i h)) \\
p_i &= p_{i-1} - h b_i (e^{q_i} - s^{(4)}(d_i h)) \\
\end{aligned}
\]

\[\text{enddo}\]

where

\[
\begin{aligned}
r^{(4)}(t) &= \frac{1}{2}(r_1 + r_2) + \sqrt{3}(r_2 - r_1) \left( \frac{1}{6} - \frac{1}{2} \right) \\
s^{(4)}(t) &= \frac{1}{6}(s_1 + 4s_2 + s_3) + (s_3 - s_1) \left( \frac{1}{6} - \frac{1}{2} \right) \\
\end{aligned}
\]

with \(r_i = r(c_i h), \ i = 1, 2, \) and \(s_j = s(c_j h), \ j = 1, 2, 3.\)

\[\text{Example 5. If } A(t) = B + C(t) \text{ with } B \text{ an unbounded nonreversible operator then the scheme (3.7) reads} \]

\[
z[0]' = \left( a_i B + \sum_{j=1}^{n} a_{i,j} C_j \right) z, \quad z[0] = z[-1] h,
\]

\[\text{with } C_j = C(c_j h) \text{ and } a_i = \sum_{j=1}^{n} a_{i,j}. \] Obviously a necessary condition for the scheme to be well defined for this problem is that \(a_i > 0, i = 1, \ldots, k.\) Unfortunately, this condition is not satisfied for methods of order greater than four [1, 8] unless complex coefficients \(a_{i,j}\) are considered for this class of schemes [5]. The analysis considered in this work allowed to find well defined schemes for numerically solving this problem.

\textbf{Appendix A. Proof of Theorem 3.1.}

\textbf{A.1. Lie algebra of Lie operators.} Let \(\mathcal{A}\) the set of all infinitely differentiable vector functions defined in a given region, \(f : D \subset \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d.\) This set, with the addition and multiplication by a scalar, and the Lie bracket as the internal product form a Lie algebra of vector fields. Given \(f \in \mathcal{A}\) we define the Lie operator, or Lie derivative, associated to \(f,\) and denoted by \(L_f,\) as the operator \(\mathcal{L}_f : \mathcal{A} \to \mathcal{A},\) given by

\[
L_f = \sum_{i=1}^{d} f_i \frac{\partial}{\partial x_i},
\]

(derivatives with respect to coordinates, but not with respect to \(t\)) which acts on \(g \in \mathcal{A}\) as

\[
L_f g = w, \quad \text{where} \quad w_i = \sum_{j=1}^{d} f_j \frac{\partial g_j}{\partial x_i}.
\]

Since \(L_f\) acts on each component of the vector field, \(g,\) we will use it both acting on vector functions as well as on scalar functions. For example, we denote \(L_f \text{Id}(x) = L_f x = f(t, x)\)

We denote by \(\hat{\mathcal{A}}\) the set of Lie operators that also have the structure of a vector space with the addition and product by a scalar

\[
\left( L_f + L_g \right) w = L_{f+g} w, \quad \alpha L_f g = L_{\alpha f g}, \quad \alpha \in \mathbb{R}.
\]

Let us now introduce the internal product, \([\cdot, \cdot] : \hat{\mathcal{A}} \times \hat{\mathcal{A}} \to \hat{\mathcal{A}}\) as follows

\[
[L_f, L_g] = L_f L_g - L_g L_f
\]
i.e. the commutator of Lie operators. This internal product is bilinear and skew-symmetric and, given $f, g, h \in \mathcal{A}$

$$[L_f, L_g]h = L_{(f,g)}h.$$  \hfill (A.5)

Then, the set $\mathcal{A}$ with the commutator as the internal product is a Lie algebra that is isomorphic to the Lie algebra of vector fields.

**Definition A.1. (Lie transformation):** Given $f(t, x) \in \mathcal{A}$, we define the associated Lie transformation, $e^{L_f} : \mathcal{A} \rightarrow \mathcal{A}$ as

$$e^{L_f} = \sum_{k=0}^{\infty} \frac{L_f^k}{k!}.$$  

Here $L_f^1 = L_f$, $L_f^0 = I$ where $I$ can be either a scalar or a vector function.

Let $x = \psi(t; x_0)$ be the solution of the autonomous equation, $x' = f(x)$. It can be written as the action of an operator on the identity function as follows

$$x = \psi(t; x_0) = \Psi_t \circ \text{Id}(x_0).$$  \hfill (A.6)

The equation to be satisfied by the operator $\Psi_t$ is

$$\frac{d}{dt} \Psi_t \circ \text{Id}(x_0) = L_f \circ \text{Id}(x_0) = L_f(\Psi_t \circ \text{Id}(x_0)) \circ \text{Id}(x_0) = \Psi_t L_f(x_0) \circ \text{Id}(x_0).$$  \hfill (A.7)

so, we have the operator differential equation

$$\frac{d}{dt} \Psi_t = \Psi_t L_f(x_0)$$

and the solution can be formally be written as

$$x = \psi(t; x_0) = e^{t L_f(x_0)} \circ \text{Id}(x_0)$$

where the derivatives act on $x_0$, that can be seen as an independent set of coordinates.

Similarly, the solution of the non-autonomous equation can formally be written as a Lie transform of the (unknown) vector field $f_\Omega(t, x_0)$, i.e.

$$\Psi_t = \exp(L_{f_\Omega(t;x_0)}).$$  \hfill (A.8)

The equation to be satisfied by $\Psi_t$ is now

$$\frac{d}{dt} \Psi_t = \Psi_t L_{f(t;x_0)}.$$  \hfill (A.9)

The linear operator $L_{f(t;x_0)}$ appears on the right hand side so we can use the property

$$0 = \frac{d}{dt} (\Psi_t^{-1} \Psi_t) = -\frac{d}{dt} \Psi_t^{-1} + \Psi_t^{-1} \frac{d}{dt} \Psi_t = \frac{d}{dt} \Psi_t^{-1} + \Psi_t^{-1} \Psi_t L_{f(t;x_0)}$$

so

$$\frac{d}{dt} \Psi_t^{-1} = -L_{f(t;x_0)} \Psi_t^{-1},$$  \hfill (A.10)

and we can solve it using the Magnus series expansion similarly to the linear case. Notice that, from (A.8) $\Psi_t^{-1} = e^{-L_{f_\Omega}}$ and then

$$-L_{f_\Omega}(h, x_0) = \int_0^h -L_{f_\Omega(\tau_1, x_0)} d\tau_1 + \frac{1}{2} \int_0^h \int_0^{\tau_1} [-L_{f_\Omega(\tau_1, x_0)}, -L_{f_\Omega(\tau_2, x_0)}] d\tau_2 d\tau_1 + \ldots$$

and taking into account the properties (A.3) and (A.5) we find that $f_\Omega(h, x_0)$ is given by (3.10), as we wanted to prove.
REFERENCES