

Symbolic Computation of High-Order Exact Picard Iterates for Systems of Linear Differential Equations with Time-Periodic Coefficients

Armando G.M. Neves*

UFMG – Departamento de Matemática
Av. Antônio Carlos, 6627 – Caixa Postal 702
30123-970 – B. Horizonte – MG, BRAZIL
`aneves@mat.ufmg.br`

Abstract. In symbolic manipulation packages such as MATHEMATICA it is possible to substitute the built-in function for integration by a user-programmed specific integration function and symbolically evaluate exact high-order Picard iterates for systems of linear differential equations with time-periodic parameter-dependent coefficients. With this technique we get excellent approximations in feasible CPU times for the solutions of these differential equations explicitly dependent on the parameters. We compare our technique with the one described by Sinha and Butcher [9], in which instead of exact Picard iterates, only a certain number of terms in the expansion of these iterates in Chebyshev polynomials is obtained. Our technique is exemplified by application to the Mathieu equation and calculation of linear stability domains for simple and double inverted pendula subjected to vertical periodic motions of the pivot.

1 Introduction

In spite of many of its properties being well-known and presented in the textbooks under the name of Floquet theory [5], the explicit solution of systems of linear ordinary differential equations with time-periodic coefficients is in general not possible to find. Of course, all of the standard numerical methods such as Runge-Kutta are available, but one of their disadvantages is that in many interesting applied problems, one wants to know how properties of the solution vary as one or more parameters are continuously varied. Instead of running the method again for each set of values of the parameters one is interested in, it would be very nice if approximations to the solutions are given as functions of the parameters.

An interesting example of a physical problem involving a system of differential equations with time-periodic coefficients depending on parameters is the study of inverted simple or multiple pendula subjected to vertical periodic motion of

* Partially supported by FAPEMIG, Research Funding Foundation of the State of Minas Gerais, Brazil

their pivots. In this case we may have as parameters the frequency and amplitude of the motion of the pivot, the lengths of the pendula, their masses, damping coefficients and so on. If we are interested in small oscillations, the equations can be linearized and, as a result of the symbolic calculation of good approximations to their solution, we can determine the regions of stability and instability of the pendula as functions of the parameters.

There exist well-known methods for general problems of differential equations involving parameters, *e.g.* perturbation theory [6] and averaging [8]. But these techniques usually do not perform well when the parameters are not too small and convergence to the true solution is difficult or impossible to check. More recently, Guttalu and Flashner [4] developed a method which performs well when the parameters are not small, but CPU time expensive and also not yielding the solution as an explicit function of time.

Still more recently, Sinha and Butcher [9], [3] introduced a method based on Picard iteration, also known as Peano-Baker series in control literature. Picard iteration is the most frequently used method in proving the theorem of existence and uniqueness of solutions to initial value problems in differential equations, see for example [2]. It is known to converge locally to the true solution as the number of iteration steps goes to infinity. We found it remarkable that, besides its theoretical interest, Picard iterations are also useful as a practical method for finding approximate solutions.

Sinha and Butcher expand the coefficient matrix of the system of differential equations in shifted Chebyshev polynomials and then Picard iterates of the truncated expansion in Chebyshev polynomials are computed symbolically in the parameters. The number m of polynomials retained in the expansion can be chosen by the user. The important fact, which explains the success of Sinha and Butcher's method is that, as a consequence of the properties of Chebyshev polynomials, one can trade the integrations in the Picard method by matrix products. Their method also performs well when the parameters are not small, runs in smaller CPU times than Guttalu and Flashner's (see comments at page 7 of [3]) and, in opposition to Guttalu and Flashner's, yields the solutions explicitly as functions of time.

The method we explain in this paper shows that it is possible to get rid of the Chebyshev expansion in the evaluation of Picard iterates, obtaining thus *exact* Picard iterates. With a Computer Algebra System (CAS) such as MATHEMATICA, we are able to perform large numbers of exact Picard iterates, as many as Sinha and Butcher in [3], in feasible CPU times. This is possible because we prove that all necessary integrals are of functions in a small definite class. We can then avoid using the built-in function for integration, which is too slow, and program instead a specific integration function which is able to evaluate the symbolic integral of any function in the class.

Our method is exemplified in this paper by applying it to the calculation of stability domains for the Mathieu equation and for inverted simple and multiple pendula driven by vertical periodic motion of their pivots. Our results were

obtained using MATHEMATICA. A package for the method was developed and may be downloaded from our web site at

<http://www.mat.ufmg.br/~aneves/diffusion/picard.htm>

In [7] we explain thoroughly the implementation details of the package and provide examples of its usage. Our specific integration function relies heavily on the strong pattern matching ability of MATHEMATICA, as exemplified in the code at the appendix. We think the same method can be implemented in other CAS, or even in some traditional programming language. Probably this will require extra programming work to mimic MATHEMATICA pattern matching constructs.

Of course the expressions for exact high-order Picard iterates are very complicated, containing hundreds or thousands of terms. Although not useful to be written in blackboard, we show that the exactness of our results yields visible differences when compared to results in [3] *calculated with the same number of Picard iterates*. In their method, because of the finite number m of terms retained in the Chebyshev expansion in each iteration step, the neglected coefficients in the series would give contributions to the retained ones. Even if one performs a very large number p of steps, the solution might not improve as a consequence of accumulated errors in the retained coefficients.

As a consequence of the proof of the existence-uniqueness theorem in differential equations, it is also possible to give rigorous bounds on the committed errors when one calculates only a finite number p of Picard iteration steps. Of course, these estimates are not available for the method by Sinha and Butcher. In a further paper we intend to provide such estimates.

2 Picard Iteration and Floquet Theory

Let $t \in \mathbb{R}$ and $x(t)$ and $f(t, x)$ be functions with values in \mathbb{R}^n . Consider then the problem of finding the solution to the initial value problem for the system of differential equations

$$\begin{cases} \frac{dx(t)}{dt} = f(t, x) \\ x(t_0) = x_0 \end{cases}, \quad (1)$$

where $t_0 \in \mathbb{R}$ is the initial “time” and $x_0 \in \mathbb{R}^n$ is the initial “position”.

The Picard iterate of a function $y(t)$ with initial condition x_0 is defined as

$$(T_{x_0}y)(t) = x_0 + \int_{t_0}^t f(\tau, y(\tau)) d\tau. \quad (2)$$

Let $(T_{x_0}^2y)(t) = (T_{x_0}(T_{x_0}y))(t)$, $(T_{x_0}^3y)(t) = (T_{x_0}(T_{x_0}^2y))(t)$ and so on. If f and $\frac{\partial f}{\partial x}$ are continuous, then it can be shown that given any continuous $y(t)$, its m th Picard iterate $T_{x_0}^m y(t)$ converges to the solution of (1) as $m \rightarrow \infty$, if t is in a suitable interval of values close to t_0 . The proof of this result is in fact

the standard proof of the existence and uniqueness theorem for solutions to differential equations, see for example [2]. If the system is linear, *i.e.* $f(t, x)$ is of the form

$$f(t, x) = A(t) x(t) , \quad (3)$$

where $A(t)$ is an $n \times n$ matrix, then convergence holds for all $t \in \mathbb{R}$.

Let e_j , $j = 1, 2, \dots, n$ denote the vector in \mathbb{R}^n with all coordinates equal to 0, except for the j th coordinate, which is equal to 1. Let $\phi_j(t)$ be the solution to the linear system (1) with $f(t, x)$ given by (3) and initial condition

$$x(t_0) = e_j . \quad (4)$$

If $\Phi(t_0, t)$ is the $n \times n$ matrix whose j th column is ϕ_j , then $x(t) = \Phi(t_0, t) x_0$ is the solution to the same system with initial condition $x(t_0) = x_0$. Matrix $\Phi(t_0, t)$ is then called a fundamental matrix solution. As in what follows we will be always taking the initial time $t_0 = 0$, we abbreviate $\Phi(t_0, t)$ as $\Phi(t)$.

As $\Phi(t)$ is in general not possible to calculate, our aim in this paper is to calculate approximations to it. We will call $\Phi^{(p)}(t)$ the approximation to $\Phi(t)$ having as j th column the p th Picard iterate of e_j with initial condition e_j , *i.e.* $T_{e_j}^p e_j$.

We will do that not in the general linear case, but in the special one when $A(t)$ is periodic of period L , *i.e.*, $A(t + L) = A(t)$. We restrict a bit more by considering matrices of the form

$$A(t) = A_0 + \sum_{k=1}^N (A_k^c \cos(\frac{2\pi k}{L} t) + A_k^s \sin(\frac{2\pi k}{L} t)) . \quad (5)$$

All our examples already have this form with $N = 1$, but in principle any reasonable periodic matrix can be approximated by one of the above form by retaining a finite number of terms in its Fourier series.

From Floquet's theory of linear systems with periodic coefficient matrix [5], stability properties of the solutions can be obtained from the fundamental matrix solution evaluated at the period, *i.e.* $\Phi(L)$ in our case. This matrix is called the Floquet transition matrix (FTM). The relation between stability and the FTM is that the equilibrium solution $x(t) = 0$ is stable (*i.e.* any solution remains close to the equilibrium solution for all times) if and only if all the eigenvalues λ_i of the FTM are such that $|\lambda_i| \leq 1$, [5].

Another result from Floquet's theory [5] we will need in the following is that the product of the eigenvalues of the FTM is

$$\prod_i \lambda_i = \exp \left(\int_0^L \text{tr} A(s) ds \right) . \quad (6)$$

3 Implementation Details

If the reader performs (by hand or with a symbolic software) some of the first Picard iterates for a linear system with matrix in the form (5) and uses easy

trigonometric identities to replace products of trigonometric functions by sums, he will be able to prove by induction that the only functions that need to be integrated are finite linear combinations of functions of the form

$$\tau^r \sin q\tau \quad \text{or} \quad \tau^r \cos q\tau, \quad (7)$$

where r and q are integers, $r, q \geq 0$. As these integrals with $r = 0$ or $q = 0$ are easily computed and integration by parts can be used recursively to reduce the values of r , we can create a new function that overrides the built-in integration function of the software and calculates any integral that can possibly come out of the iterations. As recursion is necessary, our new integration function also memorizes the values of already calculated integrals, so that these are not calculated anew every time they are needed.

The reader may convince himself that a new integration function is necessary if he computes $\Phi^{(p)}$ by using the built-in function of his software with p more than 10 iterates. We will need 20 or more iterations in our examples. In [7], it is shown that integrations are much quicker with our method.

In the typical example of approximating the solution of the Mathieu equation (10), the 15th Picard iterate of one initial condition had 1480 terms in its expanded form and only 164 terms after grouping similar ones. In using the expanded result to calculate the next iterate, each kind of integral would have to be evaluated in average 9 times, unless we first group similar terms and then perform each integration only once. Results show also that as the number of iterations grows, the computational effort necessary to group terms will eventually compensate for the smaller number of integrations to be performed. So, we provided grouping of terms before integration in our function.

The complete MATHEMATICA code for defining our integration function is given in the appendix.

4 Inverted Driven Pendula

As an example of an interesting physical system whose solution depends on a set of parameters and can be approximated by our method, consider n uniform rods, *i.e. physical pendula*, all with the same length l , mounted one over the other. The system is under influence of gravity with acceleration g and the suspension point is subjected to external periodic driving in the vertical direction, its position being given by

$$f(t) = \varepsilon \cos \omega_0 t. \quad (8)$$

It is almost incredible, but if the amplitude ε and the frequency ω_0 are carefully chosen, such an arrangement of pendula may be stabilized, see [1].

Let $\theta_i(t)$ denotes the angular coordinate of the i th pendulum with respect to an upward vertical and $\theta(t) = (\theta_1(t), \dots, \theta_n(t))$. The equations of motion of such a system can be deduced in the Lagrangian formalism, see [7]. Linearizing them in the neighborhood of the upward pointing equilibrium $\theta(t) = 0$, one finds they can be written as

$$B \frac{d^2 \theta}{d\tau^2} + \rho D \theta(\tau) = 0, \quad (9)$$

where $\rho = -\frac{g}{\omega_0^2 l} + \frac{\epsilon}{l} \cos \tau$ and $\tau = \omega_0 t$ is the rescaled time.

In the case of a single rod ($n = 1$), we have $B = \frac{1}{3}$ and $D = \frac{1}{2}$. In the case $n = 2$, we have $D = \text{diag}\{3/2, 1/2\}$ and

$$B = \begin{pmatrix} 4/3 & 1/2 \\ 1/2 & 1/3 \end{pmatrix}.$$

In order to apply our method to this system, multiply (9) by B^{-1} and then put it in form of a first-order linear system by defining the $2n$ -dimensional vector $x(t) = (\theta(t), \theta'(t))$.

In the case of one pendulum, (9) is in the form of the well-known Mathieu equation

$$\theta''(\tau) + (a + b \cos \tau) \theta(\tau) = 0, \quad (10)$$

where parameters a and b are given by

$$a = -\frac{3g}{2\omega_0^2 l} \quad (11)$$

and

$$b = \frac{3\epsilon}{2l} \quad (12)$$

and the primes now refer to differentiation with respect to τ .

In the case of n pendula, $n > 1$, besides solving directly by our method the system of $2n$ equations, it is possible to give a more clever solution [1]. It can be seen that the matrix $B^{-1}D$ is always diagonalizable with distinct positive eigenvalues μ_1, \dots, μ_n . Let $v_i, i = 1, \dots, n$ be the corresponding eigenvectors and U the matrix whose columns are v_1, \dots, v_n . Thus $U^{-1}B^{-1}DU$ is diagonal and by defining "normal" coordinates $\varphi_i = \sum_{j=1}^n (U^{-1})_{ij} \theta_j$ we have that (9) decouples to n independent Mathieu equations

$$\varphi_i''(\tau) + \mu_i \left(-\frac{g}{\omega_0^2 l} + \frac{\epsilon}{l} \cos \tau \right) \varphi_i(\tau) = 0, \quad (13)$$

$i = 1, 2, \dots, n$.

In the next section we are going to show the results obtained by our method both in the case of the Mathieu equation and in the case of a double pendulum.

5 Results

5.1 Mathieu Equation

In order to compare with the results of [9], we calculated $\Phi^{(24)}(a, b, t)$, *i.e.* the 24th exact Picard iterate for the fundamental solution matrix of (10) as an explicit function of the parameters a and b and time t . Table 1 shows the eigenvalues

Table 1. Eigenvalues of the FTM for the Mathieu equation for several values of a and b and different methods. For each set of values of a and b the top row is the “exact” result obtained by a fourth order Runge-Kutta, the middle row is our result with $p = 24$ Picard iterations and the bottom row is the result by Sinha and Butcher with $p = 24$ and $m = 15$ terms retained in the Chebyshev expansion.

		$b = 0.01$	$b = 0.75$	$b = 1.5$
$a = -0.75$	exact	0.00434 ; 230.754	0.00577 ; 173.295	0.05810 ; 17.2120
	our result	0.00434 ; 230.754	0.00577 ; 173.295	0.05810 ; 17.2120
	Sinha-Butcher	0.00433 ; 230.754	0.00577 ; 173.297	0.05801 ; 17.2129
$a = 0.0$	exact	0.99901 \pm 0.04442 <i>i</i>	-8.47371 ; -0.11801	-34.3546 ; -0.02911
	our result	0.99901 \pm 0.04442 <i>i</i>	-8.47371 ; -0.11801	-34.3546 ; -0.02911
	Sinha-Butcher	0.99901 \pm 0.04442 <i>i</i>	-8.47367 ; -0.11800	-34.3551 ; -0.02909
$a = 0.75$	exact	0.66606 \pm 0.74590 <i>i</i>	0.36764 \pm 0.92997 <i>i</i>	0.41935 \pm 0.90782 <i>i</i>
	our result	0.66606 \pm 0.74590 <i>i</i>	0.36764 \pm 0.92997 <i>i</i>	0.41935 \pm 0.90782 <i>i</i>
	Sinha-Butcher	0.66606 \pm 0.74590 <i>i</i>	0.36764 \pm 0.92997 <i>i</i>	0.41935 \pm 0.90784 <i>i</i>
$a = 1.50$	exact	0.15804 \pm 0.98743 <i>i</i>	0.32108 \pm 0.94705 <i>i</i>	1.33050 ; 0.75160
	our result	0.15830 \pm 0.98658 <i>i</i>	0.32124 \pm 0.94648 <i>i</i>	1.33078 ; 0.75136
	Sinha-Butcher	0.15829 \pm 0.98659 <i>i</i>	0.32169 \pm 0.94611 <i>i</i>	1.33493 ; 0.74892

of the 24th approximation to the FTM (*i.e.* $\Phi^{(24)}(2\pi, a, b)$) for some chosen values of a and b obtained by our method alongside with the values reported in [9] both by their method and by a fourth order Runge-Kutta with a tolerance of 10^{-8} .

Another interesting result is the picture of the stability boundaries for the solutions to the Mathieu equation. Solutions are stable for the (a, b) values such that the eigenvalues $\lambda_i, i = 1, 2$ of the FTM $\Phi(2\pi)$ satisfy $|\lambda_i| \leq 1$. Because the particular form taken by matrix A in the case of the Matheu equation is such that $\text{tr } A = 0$, it follows by (6) that $\prod_{i=1}^2 \lambda_i = 1$. As the sum of the eigenvalues is always a real number, it can be seen that solutions change from stable to unstable at the points where $\text{tr } \Phi(2\pi, a, b) = \pm 2$. The stability boundaries are the curves where this condition is fulfilled. We show in figure 1 the stability boundaries of the Mathieu equation obtained by using the 50th Picard iterate to approximate the FTM as a function of a and b and the condition of its trace being equal to ± 2 . This figure should be compared with figure 1C in [3], produced by using the same number of Picard iterates and 32 terms in the Chebyshev expansion in the same range of values for a and b (we accounted for the fact that the authors use a different form for Picard equation). Our result does not produce the spurious points seen at the far right end of their picture. This proves that exact Picard iterates, as opposed to the approximate ones calculated in [3], do produce better results. As [3] does not display numerical results such as those in table 1, we cannot numerically compare accuracy of our results with theirs.

Concerning CPU times, we cannot make exact comparisons with results by Sinha and Butcher, because we produced our results with more modern machines. The cumulative CPU time to produce their figure similar to our figure 1

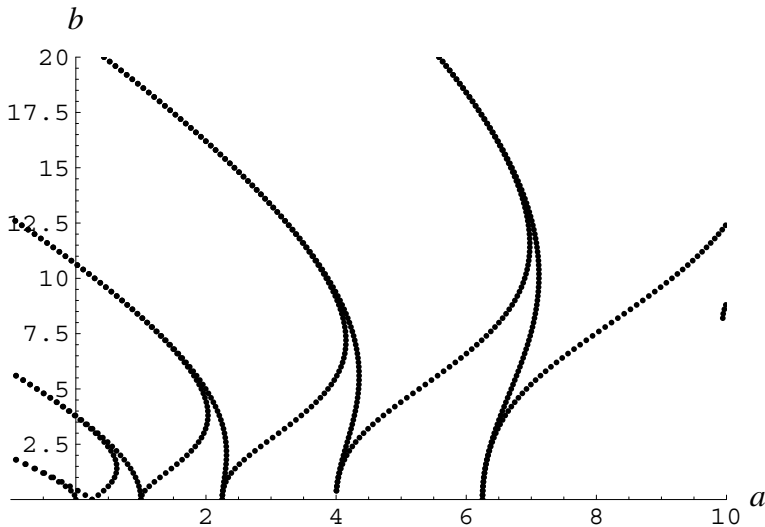


Fig. 1. Stability boundaries for the Mathieu equation calculated using 24 Picard iterates.

was 114 **hours** with a SUN SPARC 20 with 128 MB of RAM. Our figure 1 was produced by a 1 GByte Pentium 4 with 512 MB of RAM in only 52 **minutes**.

5.2 Vertically Driven Double Pendulum

As a further check for the accuracy of our method, we calculate the linear stability boundaries for a system of 2 pendula in 2 different ways. It is natural to present the stability boundary for such a double pendulum in a graph where the horizontal axis is the amplitude of the driving given in units of the length l of the pendulum and the vertical axis the frequency of the driving in units of the frequency of small oscillations of the undriven pendulum. Such a graph is given in figure 2. The full line in the figure was calculated by the normal coordinates method, as explained in the previous section, whereas the dots were produced by directly calculating the Floquet transition matrix for the system of 4 differential equations. In both instances, we used 20 Picard iterates. Notice how the results calculated by both methods agree.

6 Conclusions

By following the idea of Sinha and Butcher [9], we used Picard iterates to approximate the solutions of systems of linear ordinary differential equations with parameter-dependent periodic coefficients. We showed that in CPU times similar to theirs, or even smaller, we are able to obtain exact Picard iterates symbolically in parameters, whereas they obtain approximate Picard iterates.

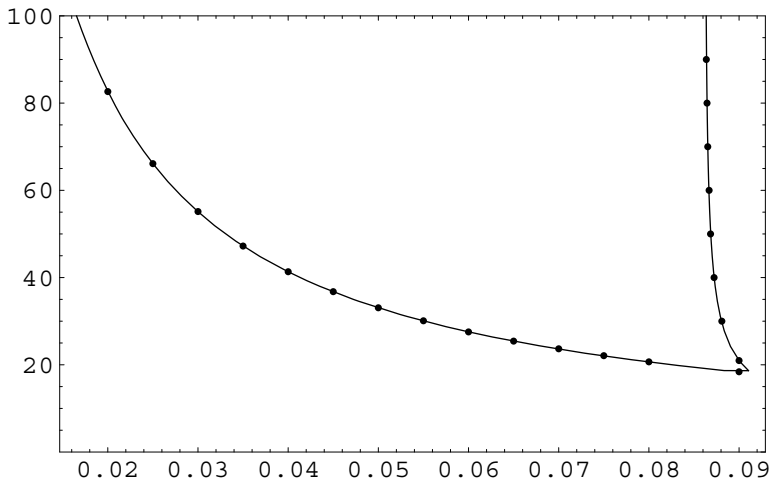


Fig. 2. Stability boundaries for a vertically driven linearized inverted double pendulum calculated by 2 methods, both with 20 Picard iterates. The full line result was calculated by the normal coordinates method. The dots were produced by directly approximating the solution for the system of 4 differential equations. For small enough amplitude, the stable region is the one above the corresponding curve, *i.e.* the high-frequency region.

Our method was tested on the Mathieu equation and also in systems of 2 and 3 linearized vertically driven pendula. Excellent results can be obtained in feasible CPU times using around 20 Picard iterates.

References

1. D. J. Acheson, A pendulum theorem, *Proceedings of the Royal Society of London A* **443**, 239–245 (1993).
2. M. Braun, *Differential Equations and their Applications*, 2nd edition. Springer-Verlag, New York (1978).
3. E. A. Butcher and S. C. Sinha, Symbolic Computation of Local Stability and Bifurcation Surfaces for Nonlinear Time-Periodic Systems, *Nonlinear Dynamics* **17**, 1–21 (1998).
4. R. S. Guttalu and H. Flashner, Stability Analysis of Periodic Systems by Truncated Point Mappings, *Journal of Sound and Vibration* **189**, 33–54 (1996).
5. D. W. Jordan, P. Smith *Nonlinear Ordinary Differential Equations*, 2nd edition, Clarendon Press, Oxford (1987).
6. A. H. Nayfeh, *Perturbation Methods*, John Wiley, New York (1973).
7. Armando G. M. Neves, Approximating Solutions of Linear Ordinary Differential Equations with Periodic Coefficients by Exact Picard Iterates, submitted to *The Mathematica Journal* (2002).
8. J. A. Sanders and F. Verhulst, *Averaging Methods in Nonlinear Dynamical Systems*, Springer-Verlag, New York (1985).
9. S. C. Sinha and E. A. Butcher, Symbolic Computation of Fundamental Solution Matrices for Linear Time-Periodic Dynamical Systems, *Journal of Sound and Vibration* **206**, 61–85 (1997).

Appendix

```

NewIntegrate[x_Plus, var_Symbol]:=
Map[NewIntegrate[#,var]&, x]

NewIntegrate[c_ x_, var_Symbol]:= c NewIntegrate[x, var]/;
FreeQ[c,var]

NewIntegrate[c_, var_Symbol]:= c var /; FreeQ[c,var]

NewIntegrate[var_^n_. , var_Symbol] := var^(n + 1)/(n + 1)

NewIntegrate[Cos[m_. var_Symbol], var_Symbol] := 1/m Sin[m var]

NewIntegrate[Sin[m_. var_Symbol], var_Symbol] :=
1/m - 1/m Cos[m var]

NewIntegrate[var_^n_. Cos[m_. var_Symbol],
var_Symbol] := NewIntegrate[var^n Cos[m var],
var] = 1/m var^n Sin[m var] -
n/m NewIntegrate[var^(n - 1) Sin[m var], var]

NewIntegrate[var_^n_. Sin[m_. var_Symbol],
var_Symbol] := NewIntegrate[var^n Sin[m var],
var] = -1/m var^n Cos[m var] +
n/m NewIntegrate[var^(n - 1) Cos[m var], var]

NewIntegrate[function_, {var_, a_ , b_}]:=
Module[{int},
int= NewIntegrate[function, var];
(int/.var->b)- (int/.var->a)
]

```