Particle Swarm Optimization Algorithm for Improve the Gregory's formula

N. Khelil[1], L. Djerou[2], A. Khernane[3] and S. Aichouche[4]

[1]LMA Laboratory, UMKBiskra, Algeria
n.khelil@univ-biskra.dz
[2] LESIA Laboratory, UMKBiskra, Algeria
ldjerou@yahoo.fr
[3] LMA Laboratory, UMKBiskra, Algeria
khernane_a@yahoo.fr
[4]LMA Laboratory, UMKBiskra, Algeria
s.math2011@gmail.com

1 Introduction

Consider the Gregory integration formula

$$\int_0^n f(x)dx = \sum_{k=0}^{n} f(k) + a_0(f(0) + f(n)) + \sum_{k\geq 1}(\frac{a_k}{k!})(\Delta_{h_1}...\Delta_{h_k} f(0) + \Delta_{-h_1}...\Delta_{-h_k} f(n)) \qquad (1)$$

with end corrections, where Δ_{h_k} is the forward difference operator with step size h_k. In this study we prove that formula (1)[1] can be optimized by minimizing some of the coefficient a_k in the remainder term by particle swarm optimization. Experimental tests prove that formula (1) can be rendered a powerful formula for library use.
Note that for $h_1 = h_2 = ... = h_k = 1$ reduces to the classical Gregory integration formula.
To justify our formula, we shall use the umbral methods developed by Rota and his school [1]-[3] instead of classical generating function technic.
When $f(x)$ is replaced by $1, x, x^2, x^3...$ we find [2].
$a_0 = -\frac{1}{2} \quad a_1(h_1) = \frac{1}{12h_1}, \quad a_2(h_1, h_2) = \frac{1}{24h_2}, \quad a_3(h_1, h_2, h_3) = \frac{1}{720h_3h_2h_1}\Big(- 1 + 5h_1^2 + 15h_1h_2\Big)$
$a_4(h_1, \cdots, h_4) = -\frac{1}{480h_4h_3h_2h_1}\Big(5/3(h_2^2h_1 + h_3h_1^2 + h_2h_1^2) - 1/3(h_1 + h_2 + h_3) + 5h_3h_2h_1\Big)$
$a_5(h_1, \cdots, h_5) = \frac{1}{60480h_5h_4h_3h_2h_1}\Big(2 + 7(-h_1^4 + h_2^2h_1^2 + 5h_2^2h_3^2 + 15h_3h_2h_1^2 + 15h_3^2h_2h_1 + 15h_4h_2^2h_1 + 15h_3h_2^2h_1 + 15h_4h_3h_1^2 + 15h_4h_2h_1^2 + 45h_4h_3h_2h_1 - 7(h_3^2 + h_2^2 + h_1^2 + 3h_4h_3 + 3h_4h_2 + 3h_4h_1 + 3h_3h_2 + 3h_3h_1 + 3h_2h_1)\Big)$

Troncating the right member of (1) at the (5)-st term, we get the approximation

$$\begin{aligned}\int_0^n f(x)dx \approx & \sum_{k=0}^{n} f(k) + a_0\Big(f(0) + f(n)\Big) + a_1(h_1)\Big(f(h_1) - f(0) + f(n-h_1) - f(n)\Big) \\ & + a_2(h_1, h_2)\Big(f(h_1 + h_2) - f(h_2) - f(h_1) + f(0) + f(n - h_1 - h_2) \\ & + f(n - h_2) - f(n - h_1) + f(n)\Big) \\ & + a_3(h_1, h_2, h_3)\Big(f(h_1 + h_2 + h_3) - f(h_2 + h_3) - f(h_3 + h_1) \\ & - f(h_2 + h_1) + f(h_3) + f(h_2) + f(h_1) - f(0) + f(n - h_1 - h_2 - h_3)\end{aligned}$$

[1] This formula has a sense so $n \geq 1$. In the contrary case an appropriate variable change will permit us to do the integral without no difficulty.

$$
\begin{aligned}
& - f(n-h_2-h_3) - f(n-h_3-h_1) - f(n-h_3) - f(n-h_2-h_1) \\
& + f(n-h_2) + f(n-h_1) - f(n)\Big) \\
& + a_4(h_1,\cdots,h_4)\Big(\Delta_{h_1}\cdots\Delta_{h_4}f(0) + \Delta_{-h_1}\cdots\Delta_{-h_4}f(n)\Big) \\
& + a_5(h_1,\cdots,h_5)\Big(\Delta_{h_1}\cdots\Delta_{h_5}f(0) + \Delta_{-h_1}\cdots\Delta_{-h_5}f(n)\Big) \qquad (2)
\end{aligned}
$$

The problem we consider here is to optimize the remainder. For do it, we try to determine h_1, h_2, h_3, h_4 and h_5 that gives us a_3, a_4 and a_5 as closer as possible to zero (0).
a_3, a_4 and a_5 is a nonlinear system of 3 equations to 5 unknowns h_1, h_2, h_3, h_4 and h_5; we take h_4, $h_5(=1$, in this study) as parameters and let's solve this nonlineair system by PSO.
The PSO method provides us the solution: $h_1 = 0.09, h_2 = 0.3$ and $h_3 = 0.08$
The particle swarm treatment supposes a population of individuals designed as real valued vectors particles, and some iterative sequences of their domain of adaptation must be established. It is assumed that these individuals have a social behavior, which implies that the ability of social conditions, for instance, the interaction with the neighborhood, is an important process in successfully finding good solutions to agiven problem.
The strategy of the PSO algorithm is summarized as follows: We assume that each agent (particle) i can be represented in a N dimension space by its current position $X_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$ and its corresponding velocity $v_i = (v_{i1}, v_{i2}, \cdots, v_{iN})$. Also a memory of its personal (previous) best position is represented by $p_i = (p_{i1}, p_{i2}, \cdots, p_{iN})$, called (pbest), the subscript i range from 1 to s, where s indicates the size of the swarm. Commonly, each particle localizes its best value so far (pbest) and its position and consequently identifies its best value in the group (swarm), called also (sbest) among the set of values (pbest). The velocity and position are updated as:

$$v_{ij}^{k+1} = w_{ij}v_{ij}^k + c_1 r_1^k[(pbest)_{ij}^k - x_{ij}^k] + c_2 r_2^k[(sbest)_{ij}^k - x_{ij}^k] \ . \qquad (3)$$

$$x_{ij}^{k+1} = v_{ij}^{k+1} + x_{ij}^k \ . \qquad (4)$$

where x_i^{k+1}, v_i^{k+1} are the position and the velocity vector of particle i respectively at iteration $k+1$, c_1 and c_2 are acceleration coefficients for each term exclusively situated in the range of $2--4$, w_j is the inertia weight with its value that ranges from 0.9 to 1.2, where as r_1, r_2 are uniform random numbers between zero and one. For more details, the double subscript in the relations (3) and (4) means that the first subscript is for the particle i and the second one is for the dimension j.

References

1. G.C.ROTA, finite operator calculus, Academic press, Inc., (1975)
2. N. Khelil, Intégration de Gregory avec correction paramtrisée, thèse de Magister , Université de Batna, (1992).
3. M.K.Belbahri, Generalized Gregory formula, Doctoral Thesis, Stevens Institute of Technology (1982).
4. K. E.Parsopoulos and M. N. Vrahatis , Modification of the Particle Swarm Optimizer for Locating all the Global Minima.
5. J.Kennedy and R.C.Eberhart, Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, (2001).
6. Y. H.Shi and R.C.Eberhart, Fuzzy adaptive particle swarm optimization. IEEE Int. Conf. on Evolutionary Computation, pp. 101-106, (2001).
7. Y. H.Shi and R.C.Eberhart, A modified particle swarm optimizer, Proc. of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9 (1998).
8. Y. H.Shi and R.C.Eberhart, Parameter selection in particle swarm optimization. Evolutionary Programming VII, Lecture Notes in Computer Science, pp. 591-600, (1998).
9. A.Zerarka and N. Khelil, A generalized integral quadratic method: improvement of the solution for one dimensional Volterra integral equation using particle swarm optimization. Int. J. Simulation and Process Modelling 2(1-2), 152-163, (2006).
10. L.Djerou et al. Numerical Integration Method Based on Particle Swarm Optimization. Y. Tan et al. (Eds.) : Part I, LNCS 6728, Springer-Verlag Berlin Heidelberg, pp. 221-226, (2011).