OVERVIEW OF (SOME) COMPUTATIONAL APPROACHES IN SPIN STUDIES

F. Lin, A.U. Luccio, N.D. Malitsky, W.M. Morse, Y.K. Semertzidis, BNL, Upton, NY, USA
C.J.G. Onderwater, University of Groningen, NL-9747AA Groningen, the Netherlands
Y.F. Orlov, R. Talman, Cornell University, Ithaca, NY, USA

Abstract

In the proposed electric dipole moment (EDM) experiment, with an estimated spin coherence time of 1000 s, the spin precession due to an EDM of 10^{-29} e.cm will produce a change in the vertical spin component of approximately 10 μrad during the storage time. Such high sensitivity needs an highly accurate and reliable simulation environment of the beam and spin behavior during the storage time. Therefore, several spin-related accelerator simulation programs have been considered. The paper surveys the computational algorithms of these approaches and discusses their comprehensive analysis from multiple perspectives.

INTRODUCTION

Introduced by Uhlenbeck and Goudsmit to explain the result of Stern-Gerlach experiments, spin has become a fundamental concept and plays an important role in the interactions of elementary particles. To study the various related phenomena, different experiment environments are required. For example, to study spin dependence in the interactions at the quark and gluon level, one employs collision of intense beams of polarized protons at high energy. The Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Lab provides a unique facility for this study. Here, polarized protons can be collided with 50 to 500 GeV center of mass energy. The design calls for an intensity of 2×10^{11} protons per bunch with a polarization of 70%. When the polarized beam is produced from the source, accelerated by several pre-acceleration facilities, injected and ramped in RHIC to the required energy, numerous spin resonances due to the interaction of the magnetic moment and external electromagnetic fields can deteriorate the polarization. Hence, the spin dynamics have to be understood and solutions have to be proposed to preserve the polarization during the acceleration and storage.

Recently, another quest for physics beyond the Standard Model (SM) represents a major effort in basic physics research. A non-vanishing EDM is a violation of Time-Reversal (T) and Parity (P) symmetries, and under the assumption of CPT invariance also violates the CP symmetry. Because the Electric Dipole Moment (EDM) values predicted by most extensions to the SM are many orders of magnitude larger than those of the SM itself and close to present experimental sensitivity levels, EDM experiments have become very sensitive probes for new physics, such as new sources of CP violation.

A completely new approach to EDM studies is based on a charged polarized particle storage ring [1, 2]. A non-zero EDM will affect the observed spin precession, resulting in the eventual change of the polarization. This technique promises a significant sensitivity improvement, reaching down to 10^{-29} e.cm in 10^7 s of physics running time. In such a long time running, the polarization of the beam has to be maintained, which requires the spin dynamics systematic errors need to be tightly controlled.

The most general description of spin motion under the influence of external electromagnetic fields is

$$\frac{d\vec{S}}{dt} = \mu\vec{S} \times \vec{F}_\mu(\vec{B}, \vec{\beta} \times \vec{E}) + d\vec{S} \times \vec{F}_d(\vec{E}, \vec{\beta} \times \vec{B}). \quad (1)$$

Here, the spin vector $\vec{S}$ is in the particle rest frame, $\vec{B}$ and $\vec{E}$ stand for the laboratory magnetic field and electric field, respectively. The first term, representing the spin precession due to the magnetic dipole moment $\mu = g\frac{e}{2mc}$, has been explored in previous accelerator experiments, for example at RHIC. The second term is the spin precession due to the electric dipole moment $d = \eta\frac{e}{2mc}$, which is proposed for study in the EDM experiment.

The design, optimization, and commissioning of modern accelerator complexes rely on dedicated beam studies based on advanced numerical approaches. Analysis of the spin motion required further development of conventional accelerator codes by augmenting positional coordinates with spin coordinates and combining the Lorentz and Thomas-BMT equations. However, such composite spin-orbital applications do not affect the basic computational framework, especially given that particle orbits are essentially independent of spin orientation (Stern-Gerlach forces are negligible at the high particle energies considered). The same numerical approaches are applicable and can be divided into two major categories: tracking and mapping. The list of successful implementations of these approaches is very long and this paper does not presume to cover all of them. Its primary goal is rather practical: to build an open simulation environment addressing the challenging EDM experiment. The correction of spin (g-2) frequency was selected as an initial benchmark application.

EQUATIONS OF MOTION

Present beam simulation programs have usually considered only the first term of Eq.(1) dealing with the spin procession due to the magnetic dipole moment. This term with

the magnetic field $\vec{B}$ and electric fields $\vec{E}$ defined in the laboratory frame is described by the Thomas-BMT equation [3]

$$\frac{d\vec{S}}{dt} = \vec{S} \times \vec{F}, \tag{2}$$

where

$$\vec{F} = \frac{ev_0}{p_0 c}\left[(a\gamma+1)\vec{B} - \frac{a\gamma^2}{\gamma+1}(\vec{\beta}\cdot\vec{B})\vec{\beta} - (a\gamma+\frac{\gamma}{\gamma+1})\vec{\beta}\times\vec{E}\right]. \tag{3}$$

Here $a = (g-2)/2$ (also called G in much of the accelerator literature) is the anomalous magnetic moment, $\vec{\beta} = \vec{v}/c$ and γ is the Lorentz factor.

In a circular accelerator, the particle motion is conveniently described relative to the trajectory of a reference particle with momentum p_0 since all machine magnets are positioned relative to this trajectory. In this coordinate system, called Frénet-Serret plane coordinate system, the Thomas-BMT equation can be rewritten as

$$\begin{aligned} \frac{dS_x}{ds} &= \left[\vec{S}\times\vec{F}\right]_x \frac{dt}{ds} + hS_z \\ \frac{dS_y}{ds} &= \left[\vec{S}\times\vec{F}\right]_y \frac{dt}{ds} \\ \frac{dS_z}{ds} &= \left[\vec{S}\times\vec{F}\right]_z \frac{dt}{ds} - hS_x \end{aligned} \tag{4}$$

where

$$\frac{dt}{ds} = \frac{1+hx}{\frac{p_z}{p_0}}\frac{1}{v_0}. \tag{5}$$

Similarly, the Lorentz equations describing the particle behavior in an external electromagnetic field in the Frénet-Serret coordinate system are expressed in the following form:

$$\begin{aligned} \frac{d}{ds}\left(\frac{p_x}{p_0}\right) &= \frac{e}{p_0 c}\left[\frac{\vec{E}}{\beta_0} + \frac{\vec{p}}{p_0}\times\vec{B}\right]_x \cdot \left(\frac{dt}{ds}\cdot v_0\right) + h\frac{p_z}{p_0} \\ \frac{d}{ds}\left(\frac{p_y}{p_0}\right) &= \frac{e}{p_0 c}\left[\frac{\vec{E}}{\beta_0} + \frac{\vec{p}}{p_0}\times\vec{B}\right]_y \cdot \left(\frac{dt}{ds}\cdot v_0\right) \\ \frac{d}{ds}\left(\frac{\mathcal{E}}{p_0 c}\right) &= \left(\frac{e\vec{E}}{p_0 c}\right)\cdot\frac{\vec{p}}{p_0}\cdot\left(\frac{dt}{ds}\cdot v_0\right) \end{aligned} \tag{6}$$

where the third component of momentum is directly calculated as

$$\frac{p_z}{p_0} = \sqrt{\frac{\mathcal{E}^2 - (m_0c^2)^2}{p_0^2c^2} - \left(\frac{p_x}{p_0}\right)^2 - \left(\frac{p_y}{p_0}\right)^2}. \tag{7}$$

The above equations are given in the canonical coordinates used in the MAD [4] and MAD-X program [5]. This set of coordinates is not unique and is chosen differently in different accelerator codes.

TRACKING APPROACH

Introduced in the early days of the SSC, the code TEAPOT [6] had the narrowly defined purpose of investigating proton emittance growth (and lifetime reduction) due to magnetic imperfections. Symplecticity was therefore of paramount importance. Because of the difficulty of preserving symplecticity with thick elements, the code accepts only zero length elements, making it a "kick" code, or a "finite element" code, or a "symplectic integrator".

TEAPOT, like all tracking codes, is thus a numerical differential equation solver (of the Lorentz force equation). Finite-length elements are sliced as narrowly as is required for the desired accuracy. Unlike some symplectic integrators, however, negative drift lengths are disallowed. This is consistent with the philosophy that zero length elements are "better than" finite length elements, and that the sliced accelerator, though idealized, is otherwise physical.

Conceptually the accelerator consists of a sequence of zero length elements alternating with finite element drifts. All orbits are therefore straight lines with kinks. All straight lines and kinks are calculated exactly and symplectically. One therefore calculates "exactly in an approximate accelerator", rather than "approximately in an exact accelerator".

All longitudinal evolution proceeds by straight line segments. A single slice of a magnet is represented by a "multipole plane" which includes both intended and unintended (error) fields. After solving for the intersection of an individual particle trajectory with this plane, the orbit kink is calculated and the new straight line determined. Particle tracking consists of iterating this process for all lattice elements and for each particle.

Unlike most beam dynamics codes, neither linear nor nonlinear transfer matrices are used for particle propagation. But transfer maps and lattice functions are calculated numerically and made available as optional output. This capability has been completely superceded by arbitrary order truncated Taylor series determination in the TEAPOT module of UAL [7]. Numerous other simulation features such as bunch generation, orbit smoothing, tune and chromaticity adjustment, and decoupling are also supported, along with even more specialized features.

By augmenting the six phase space coordinates with three spin coordinates, and solving also the BMT equation using the same finite element approach, it was relatively straightforward for S. Mane, in 1994, to extend the code to include spin tracking.

Another approach has been implemented in the spin tracking code SPINK [8]. This program was written for the RHIC project at Brookhaven National Laboratory and employed for years to study the behavior of polarized protons in all stages of the accelerator complex. SPINK uses a composite approach. Its orbit module is based on the first order matrices and the second order Transport maps produced by the MAD program. Spin propagation was implemented by the additional thin elements, spin kicks, which

rotate the spin in each accelerator element using the spin rotation matrix $\mathcal{M}$:

$$\mathbf{S} = \mathcal{M}\,\mathbf{S_0} \tag{8}$$

This procedure has the advantage of very high computational speed using matrices that by construction represent a rotation of the spin vector by an angle $\delta\mu$ around an axis defined by its latitude θ and longitude ϕ, with respect to the reference coordinate axes.

Eqs.(4) yield three third order formally identical linear equations for the three components of the spin

$$S''' + \omega^2 S' = 0, \tag{9}$$

with

$$\omega^2 = f_x^2 + \left(f_y - \frac{1}{\rho}\right)^2 + f_z^2, \quad \vec{f} = \vec{F}\cdot\frac{dt}{ds}. \tag{10}$$

The general integral is

$$S = C_1 + C_2\cos(\delta\mu) + C_3\sin(\delta\mu). \tag{11}$$

with

$$\delta\mu = \omega\,\delta s \tag{12}$$

the spin rotation angle (spin kick) in the machine element. δs is the path length through the element along the reference orbit. The constants of integration, $C_1,\ C_2,\ C_3$, can be found as a linear function of the initial and the final values of the spin components using $S,\ S',\ S''$ and the original system. The algebra is tedious but straightforward and the resulting linear relation between spin after and before kick, expressed in the form of the 3×3 matrix $\mathcal{M}$, is

$$\begin{pmatrix} 1-(B^2+C^2)c & ABc+Cs & ACc-Bs \\ ABc-Cs & 1-(A^2+C^2)c & BCc+As \\ ACc+Bs & BCc-As & 1-(A^2+B^2)c \end{pmatrix}$$

with

$$c = 1-\cos(\delta\mu) \quad s = \sin(\delta\mu) \tag{13}$$

and

$$A = \frac{f_x}{\omega}, \quad B = \frac{f_y - \frac{1}{\rho}}{\omega}, \quad C = \frac{f_z}{\omega} \tag{14}$$

$\mathcal{M}$ is a parametric linear transformation, whose elements are function of the values of the particle coordinates in the laboratory, and has a determinant

$$det(\mathcal{M}) = 1. \tag{15}$$

The matrix represents a rotation of the spin vector by an angle $\delta\mu$ around an axis defined by the two angles θ and lϕ. The coefficient of Eq.(14) can be expressed as a function of these angles, namely

$$A = \cos\theta\,\sin\phi, \quad B = \sin\theta, \quad C = \cos\theta\,\cos\phi \tag{16}$$

MAPPING APPROACH

The transformation of spin-orbit variables after passage through given beam line, which could be a single pass transport channel or one turn in circular accelerator, can be represented in the form of the map

$$\begin{cases} \vec{z}_f &= \vec{F}(\vec{z}_i) \\ \vec{S}_f &= A(\vec{z}_i)\cdot\vec{S}_i \end{cases} \tag{17}$$

where $\vec{z}$ denotes the combined vector of six orbital variables and 3×3 matrix $A(\vec{z})$ belonging to the $SO(3)$ group. Assuming that $\vec{F}(\vec{0}) = \vec{0}$ (i.e. orbital variables are deviations from reference orbit) and elements of the vector function $\vec{F}$ and matrix A are differentiable functions, one can make an approximation of the map in Eq.(17) expanding its right hand side in Taylor series with respect to $\vec{z}$ and truncating this expansion at the certain order.

There is a variety of ways to calculate the numerical values of the coefficients of this truncated map. A.W.Chao program SLIM, the first in this field, utilizes explicit formulas and allows the treatment of the linear expansion of the matrix A and assumes linear orbit motion [11]. The code SPINLIE also uses explicit formulas obtained by formula manipulators and extends the treatment of spin-orbit motion up to third order [12]. It is clear that the complexity of this semi analytical approach grows rapidly with the truncation order making further development difficult.

Another possible way is to use some spin-orbit tracking code with subsequent application of numerical differentiation technique. This approach has its own difficulties and will not be discussed here.

It seems that the most reliable way lies in the area of analytical manipulations with polynomials. This allows obtaining spin-orbit map in a way which is independent from the truncation order. The first program with such possibilities for spin motion was FMN [10] written in 1992 under the name VasiLIE. The computer code FMN was applied for several projects, for example, to the investigation of schemes for preserving the polarization in the TRIUMF KAON Booster, to the study of the possibilities to accelerate polarized proton beams in the Nuclotron ring in Dubna, in the HERA proton ring, and others. Later on similar possibilities were included into COSY INFINITY [13].

Both codes, FMN and COSY INFINITY, use such powerful methods as numerical integration using the differential algebra techniques and the direct summing of Lie exponent series for s-independent elements. Note that Taylor map coefficients obtained by these methods accumulate two types of errors: computer rounding errors and errors defined by the size of the integration step (or by the number of terms in the Lie series taken for summation). There is an approach free from the defects of the second type in which the number of the steps does not depend on a precision (like in the Gauss method of matrix inversion, where the number of operations depends on the matrix dimension only), which is also used in the FMN code [14].

UAL FRAMEWORK

The Unified Accelerator Libraries (UAL) attempt to "manage the complexity" of accelerators by providing an environment for simulating a variety of properties of a variety of accelerators using a variety of simulation codes and methods. The intended value of the environment is to provide homogeneous access to these resources while masking their diversity yet assuring their consistency. This allows different methods to be consistently applied to the same accelerator and the same methods to be applied to different accelerators.

To facilitate such unification UAL has introduced an open architecture in which diverse accelerator codes are connected together via common accelerator objects such as $Element$, $Bunch$, $Twiss$, etc. In this architecture each accelerator code is implemented as an object-oriented library of C++ classes. There is a very natural identification of physical elements, such as magnets, with their representation by computer objects. UAL supports considerable flexibility in the attributes of all objects - certainly enough that all attributes of objects contained in modules included so far have been describable without constraint. Such flexibility has made it practical to evaluate, compare, and integrate a variety of design models and to build heterogeneous, project-specific applications.

The accumulated experience has motivated the development of the *Element-Algorithm-Probe* framework [15], a uniform mechanism for combining diverse modules to simulate complex combinations of the physical effects and dynamic processes. *Elements* are things like bending magnets, RF cavities, collimators, and so on. *Algorithms* are mathematical formulas capable of evolving quantities known at element inputs to their corresponding values at element outputs. So both element and algorithm are terms of common usage, likely to be understood unambiguously by all workers in the field. The term *probe*, because it is less standard, requires more explanation. A similar term is *observable*. As used in UAL a particle or beam bunch probes the lattice elements. Other things also probe the lattice. For example, the transfer map (identity at the origin) evolves into the transfer map from origin to element output as it evolves through an element. So a probe is anything whatsoever for which continuous evolution is meaningful and the evolution is unambiguously caused by the elements making up the lattice. It therefore makes sense to evaluate the evolution of a probe caused by the lattice. Examples of probes are 6D particle coordinates of all particles in a bunch, spin components, moments of a bunch of particles, lattice functions such as Twiss functions and dispersion functions, transfer matrices (i.e. linear maps), nonlinear maps (which are represented by truncated power series), wake fields, and so on.

Several years ago, The BNL team extended UAL with the space charge module SIMBAD [16] as a part of the European Union contract. At this time, addressing the special requirements of the EDM experiment, UAL accommodated another BNL code, SPINK, described in the previous section on tracking approaches. This development solved three major tasks. First, UAL allowed the upgrade of the original SPINK version with the TEAPOT symplectic tracking engine and provided access to other modules, for example, the differential algebra-based integrator and others. Second, the SPINK Fortran code has been refactored and transformed into a few dedicated C++ classes. As a result, it added flexibility for developing new extensions, such as new types of elements and enhancing the accuracy of the original approach with the slicing technique. Finally, the SPINK algorithm became a part of the simulation infrastructure which can be applied to other projects.

(G-2) EXPERIMENT

In the studies of spin motion, the (g-2) frequency is defined as the frequency of the spin precession relative to the momentum vector, describing the rate of change of the component of spin $\vec{S}$ parallel to the velocity. For a particle with momentum p_0 moving in a plane perpendicular to the magnetic field $\vec{B}$, the (g-2) frequency is given by Thomas-BMT equation as

$$\omega_0 = \frac{e\mathcal{B}}{p_0 c}\gamma v_0 \cdot \frac{g-2}{2}, \qquad (18)$$

where $\mathcal{B}$ is the magnitude of the field. If the orbit is not exactly perpendicular to the magnetic field, a small correction to the (g-2) frequency will appear. This pitch correction has been addressed and examined in a few papers [17, 18] when the pitch angle (the angle with respect to the plane perpendicular to the magnetic field) is varying due to axial focusing force, presented by an almost uniform and only particle linear position dependent magnetic field.

As part of a calibration effort of particle tracking codes for the storage ring of electric dipole moment (EDM) experiments (proton and deuteron), a study of the pitch correction has been carried out by simulating a circulating particle in a continuous ring with weak magnetic focusing. Given the field focusing index $n = -[\frac{\rho_0}{B_0}\frac{\partial B_y}{\partial x}]$, the pitch correction is extended [19] because of the quadratic dependence on y^2 in the curvilinear coordinate system. In the limit of fast pitch change ($\omega_y \gg \omega_0$, with ω_y the frequency of the vertical betatron oscillation), the corrected (g-2) frequency is given as

$$\omega_a = \omega_0 \left[1 - \frac{a}{2}\psi_0^2 + \frac{n}{2(1-n)}\psi_0^2\right]. \qquad (19)$$

Here, ω_0 is the uncorrected (g-2) frequency given in Eq.(18), $a = (g-2)/2$ and ψ_0 is the amplitude of the pitch angle with respect to the xz-plane perpendicular to the y direction. This analytical formula has been confirmed quantitatively by the integration of the differential equations of motion using the subroutine TDHPCG [20] based on the fourth order corrector-predictor method of Runge-Kutta.

We applied this result to a simple muon ring defined by the parameters in table 1. For n equals 0.13, $\omega_y/\omega_0 \approx 225$

Table 1: Testing muon particle and ring parameters.

Muon momentum p (GeV/c)	0.1
Focusing Index n	0.13
Radius ρ (m)	5
Anomalous magnetic moment a	0.00116592
Initial pitch angle ψ_0 (mrad)	1
Initial spin vector Sx, Sy, Sz	0, 0, 1

close to the limit of fast pitch change. The results from the analytical formula Eq.(19) and the UAL simulation are listed in table 2. The number of slices each element broken up into in UAL is listed in the first column. The last column shows the CPU time of per particle per turn.

Table 2: List of pitch correction from analytical formula and simulation of UAL.

Approach	$\frac{\omega_a-\omega_0}{\omega_0}(\times 10^{-8})$	CPU time (ms)
Analytical formula	7.42	
UAL, slices = 8	6.48 ± 0.01	0.86
UAL, slices = 32	7.34 ± 0.02	2.97
UAL, slices = 64	7.38 ± 0.02	5.81
UAL, slices = 128	7.39 ± 0.02	10.66
UAL, slices = 256	7.40 ± 0.02	21.72

SUMMARY

This paper has given a brief summary of the most prominent sorts of accelerator simulation codes. The motivation for this is the need for extraordinarily precise spin tracking calculations required for interpreting high precision experimental measurements of magnetic and, especially, electric dipole moments. The presentation has attempted to be neutral as regards physical methods, but partial as regards establishing an environment in which diverse methods of calculation can be applied to multiple physical effects.

Many component errors and imperfections are capable of producing behavior that can be misinterpreted as being due to the particular dipole moment being measured. Examples are fringe fields, errors, nonlinearities, etc. In this report just one (particularly simple)such effect has been investigated and compared with an analytic calculation. This has been intended only as a prototype of the sorts of calculations and comparisons that will be required. This initial step needs to be extended in many directions: new benchmarks, new programs, especially from the mapping category (like FMN and COSY), and new extensions encompassing all relevant physical effects.

ACKNOWLEDGEMENT

One of the authors (N.Malitsky) would like to thank V.Balandin, N.Golubeva for their invaluable input.

REFERENCES

[1] F.J.M.Farley et al. Phys.Rev.Lett. 93, 052001 (2004).

[2] "AGS Proposal: Search for a permanent electric dipole moment of the deuteron nucleus at the 10^{29} e.cm level." Spokesperson Y.K.Semertzidis (2008)

[3] L.H.Thomas, Phil. Mag. 3, 1 (1927) ; V.Bargmann, L.Michel, V.L.Telegdi, Phys. Rev. Lett. 2, 435 (1956).

[4] H.Grote and F.Iselin, The MAD Program (Methodical Accelerator Design), User's Reference Manual, CERN/SL/SO-l3(A:P) (1990).

[5] P.K.Skowronski, F.Schmidt, E.Forest. "Advances in MAD-X using PTC", Proceedings of PAC07, p.3381, Albuquerque, New Mexico, USA (2007).

[6] L.Schachinger and R. Talman, "TEAPOT: A Thin-Element Accelerator Program For Optics and Tracking", Particle Accelerators, Vol.22, pp.35-56 (1987).

[7] N.Malitsky and R.Talman, Unified Accelerator Libraries. AIP 391 http://code.google.com/p/ual (1996).

[8] A.U.Luccio, Proceedings of the Adriatico Research Conference on Trends in Collider Spin Physics p.235 Trieste, Italy (1995).

[9] V.Balandin and N.Golubeva, *Hamiltonian Methods for the Study of Polarized Proton Beam Dynamics in Accelerators and Storage Rings*, DESY 98-016, February 1998.

[10] V.Balandin and N.Golubeva, *Nonlinear Spin Dynamics*, Proc. XV Int. Conf. High Energy Accelerators, Hamburg, 1992, Int. J. Mod. Phys. A, **2B**, 998, (1992).

[11] A.W.Chao, Nucl. Instr. Meth., **180** (1981) 29.

[12] Yu. Eidelman and V. Yakimenko, Proc 1991 IEEE Part. Accel. Conf., 1991, San-Francisco.

[13] V.Balandin, M.Berz and N.Golubeva, *Computation and Analysis of Spin Dynamics*, Computational Accelerator Physics Conference, Williamsburg, Virginia, USA, (1996).

[14] V.Balandin, *One of Methods to Extract Truncated Taylor Map for Orbital and Spin Motion in Proton Storage Rings*, Particle Accelerator Conference, Washington, USA, (1993).

[15] N.Malitsky and R.Talman, Framework of Unified Accelerator Libraries. ICAP'98 (1998).

[16] N.L.D'Imperio, A.U.Luccio, N.Malitsky, O.Boine-Frankenheim, Parallel 3-D Space Charge Calculations in the Unified Accelerator Library. EPAC'04 (2004).

[17] S.Granger, G.W.Ford, "Electron Spin Motion in a Magnetic Mirror Trap", Phys. Rev. Lett. 28, 1479 (1972).

[18] F.J.N.Farley, "Pitch Correction in (g-2) Experiments", Phys. Lett. 42B, 1, 66 (1972).

[19] F.Lin, etc "Extended Pitch Correction with Weak Magnetic Focusing", EDM internal note, (2009)

[20] This program is the differential equation solver-DHPCG which uses the fourth order corrector-predictor method. This is a modification of subroutine DHPCG from the scientific subroutine package (SSP) published by IBM.