ISBN: 978-3-95450-132-8 doi:10.18429/JACoW-IPAC2014-MOPME031

SOLCALC: A SUITE FOR THE CALCULATION AND THE DISPLAY OF MAGNETIC FIELDS GENERATED BY SOLENOID SYSTEMS*

M. L. Lopes[#], Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

Abstract

SolCalc is a software suite that computes and displays magnetic fields generated by a three dimensional (3D) solenoid system. Examples of such systems are the Mu2e magnet system and Helical Solenoids for muon cooling systems.

SolCalc was originally coded in Matlab, and later upgraded to a compiled version (called MEX) to improve solving speed. Matlab was chosen because its graphical capabilities represent an attractive feature over other computer languages. Solenoid geometries can be created using any text editor or spread sheets and can be displayed dynamically in 3D. Fields are computed from any given list of coordinates. The field distribution on the surfaces of the coils can be displayed as well.

SolCalc was benchmarked against a well-known commercial software for speed and accuracy and the results compared favorably.

INTRODUCTION

The fundamental field element for a solenoid is the filamentary circular current-loop. The off-axis magnetic field from a circular loop can be calculated from the vector potential using cylindrical coordinates (r, φ, z) [1]:

$$A_\varphi(r,z) = \frac{\mu_o I}{\pi k}\sqrt{\frac{a}{r}}\left[\left(1-\frac{k^2}{2}\right)K - E\right] \qquad (1)$$

where a is the radius of the circular loop, I is the current, μ_o is the magnetic permeability of the vacuum, K=K(k) and E=E(k) are, respectively, the complete elliptic integrals of first and second kinds [2], and k is defined as:

$$k^2(r,z) = \frac{4ar}{(a+r)^2+z^2} \qquad (2)$$

The radial (r) and longitudinal (z) magnetic field components can be computed by:

$$B_r(r,z) = \frac{\mu_o I}{2\pi\sqrt{(a+r)^2+z^2}}\frac{z}{r}\left[-K+\frac{a^2+r^2+z^2}{(a-r)^2+z^2}E\right],$$
$$B_z(r,z) = \frac{\mu_o I}{2\pi\sqrt{(a+r)^2+z^2}}\left[K+\frac{a^2-r^2-z^2}{(a-r)^2+z^2}E\right]. \qquad (3)$$

The field of a solenoid can be obtained from (3) by integrating a over the inner and outer radii (R_i and R_o) and by integrating z over the length of the solenoid (L), using Gaussian quadrature [3].

MAIN FEATURES

The SolCalc core was initially written in Matlab [4] language. Currently, the SolCalc package contains the following functions:

- ***FieldSolenoid***: Calculates the Cartesian components of the magnetic field (Bx, By and Bz) at a given (X, Y, Z) coordinate for a single solenoid positioned and oriented in any place in the space.
- ***FieldSolenoidSystem:*** Computes the magnetic field (Bx, By, and Bz) of a list of Cartesian coordinates (X, Y, Z).
- ***DisplaySolenoid:*** Displays a single solenoid given its geometry.
- ***DisplaySolenoidSystem:*** Displays a system of solenoids or parts of such system.
- ***FieldCoil:*** Calculates the field distribution on the surface of a coil. Maximum and minimum values are some of the output parameters.
- ***DisplayFieldCoil:*** Displays the field distribution on the surface of a coil. The user can choose to display Bx, By, or Bz as well as Bmod (the absolute field).
- ***Ptrack:*** Calculates the trajectory of a charged particle through a solenoid system.

In order to speed-up the field calculations, ***FieldSolenoid*** function was converted from Matlab to a compiled C version (referred to as MEX [4]).

EXAMPLES

SolCalc is particularly useful when the system does not have an axial symmetry. An example can be seen in figure 1; it shows a Helical Solenoid for muon cooling systems [5-7].

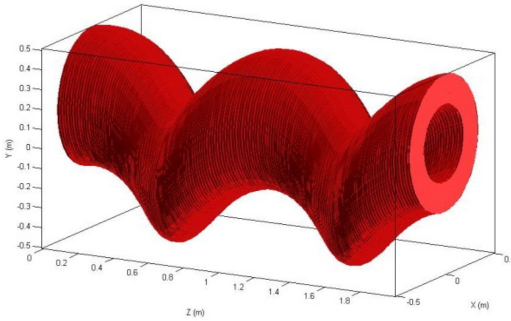


Figure 1: A helical solenoid generated with SolCalc.

Figure 2 shows a plot of the absolute field distribution for the Mu2e Transport Solenoids (TS) [8-11]. This figure illustrates capabilities to calculate and display fields using above functions. It shows 52 coils in different positions

*Work supported in part by Fermi Research Alliance under DOE Contract DE-AC02-07CH11359.
#mllopes@fnal.gov

and with different orientations as well as geometric parameters (R_i, R_o and L).

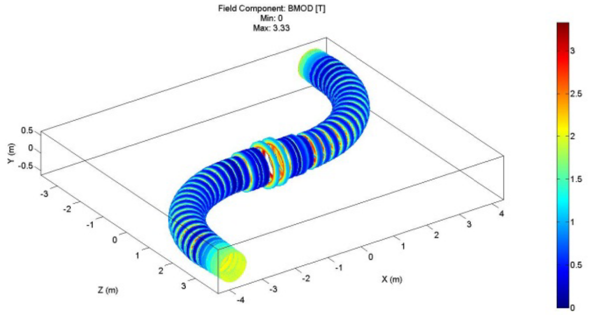


Figure 2: Magnetic field distribution on the surface of the coils of the Mu2e TS.

Another application is the particle tracking through a system of solenoids. An example can be seen in Figure 3. It shows a Guggenheim-like solenoid structure for a muon cooling scheme [12].

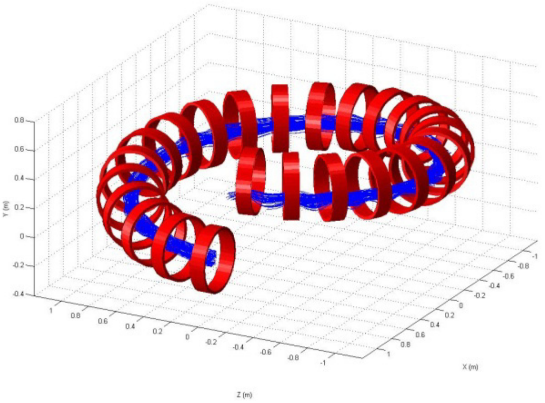


Figure 3: Particle tracking in a Guggenheim-like solenoid structure.

SolCalc takes advantage of Matlab's Parallel Computing Toolbox enabling the use of multi-core CPUs, which considerably reduces computation times. Other toolboxes, such as the Matlab Compiler, can be used to generate stand-alone applications to be distributed to a group of collaborators working on the same project. An example can be seen in Figure 4 which shows a Graphical User Interface (GUI) for the fine tuning of the Mu2e Detector Solenoid (DS) [13]. The magnetic field is computed at three different radii as well as the longitudinal gradient. The coils can be modified on-the-fly and the field is updated automatically.

BENCHMARK

The field calculation function was benchmarked in terms of computing time and precision. Results were compared to Opera3D [14]. Opera3D is a well-known commercial software that can compute magnetic fields in 3D. Opera3D currently does not take advantage of multi-core CPUs.

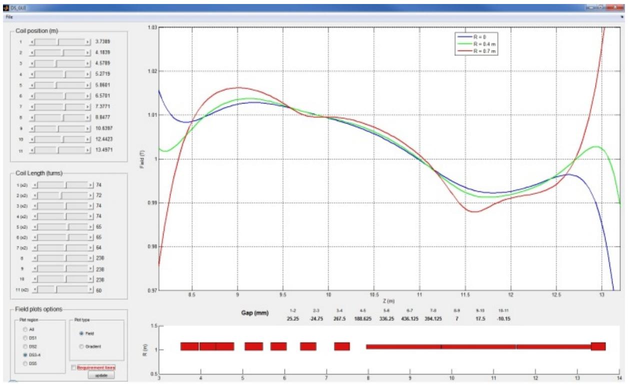


Figure 4: A Graphical User Interface powered by SolCalc.

The field at 5477 points along the TS coils centroid was computed. The Mu2e magnet system is composed of 66 coils (TS alone is composed of 52 coils). A routine was written to compute the field using SolCalc (in single-core and multi-core modes). The routine computes the field from all the coils in each point (SolCalc). In order to speed up the computation, one can modify the routine to ignore the field from coils that are far (larger than 5 times the coil radius, SolCalc-I5) from the point where the field is being calculated. The process was repeated using 10 times the radius of the coil (SolCalc-I10). The results do not include the time to prepare the model or extract the data from files. Results are summarized in Table 1.

Table 1: Comparison of the Computation Time

Software	Number of cores	Computation time (s)
SolCalc	1	152.2
SolCalc	6	27.6
SolCalc -I5	1	45.9
SolCalc -I5	6	8.1
SolCalc -I10	1	102.0
SolCalc -I10	6	18.1
Opera3D	1	128.7

As can be seen, without any approximations, SolCalc is about 18% slower as compared to Opera3D. However, by enabling multi-core processing, the calculation is considerably faster. The absolute field component was compared with results from Opera3D. Results are summarized in Table 2.

When comparing results of SolCalc with Opera3D the maximum difference is less than 1%. Using SolCalcI5 increases the difference to ~3%. With SolCalc10, the difference is of the same order as when all coils are included in the calculation. The ~3% difference can be acceptable at the design and optimization stage of a solenoid system as it yields a great gain in computational time.

Table 2: Field Differences

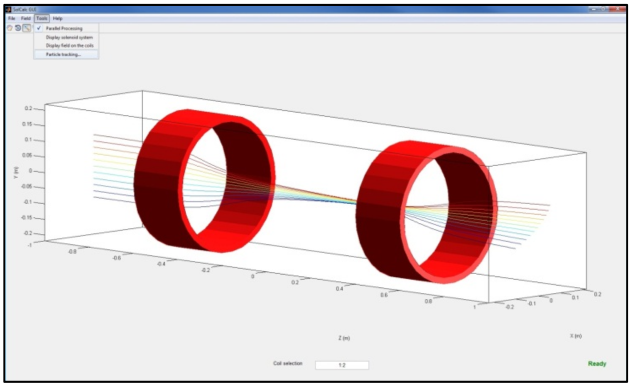
Case	Max. difference
SolCalc – Opera 3D	0.6%
SolCalc-I5 – Opera 3D	3.1%
SolCalc-I10 – Opera 3D	0.4%

STAND-ALONE VERSION

A stand-alone version of SolCalc is also available. This version of SolCalc has a GUI. Figure 4 shows the main screen for SolCalc GUI.

This version does not need Matlab to run however one needs to install the Matlab Compiler Runtime (MCR) [15] package in order to run. This package can be obtained on the Mathworks website and is free to use and a copy of it is distributed with SolCalc GUI.

SolCalc GUI has basically the same capabilities of the Matlab suite. The SolCalc GUI main interface can be seen in figure 5. The user can load files containing the geometry from the top menu; the geometry is displayed in the central part automatically. All the coils will be displayed by default; the user can choose what to be displayed. Modifications to the geometry or the creation of a new one can be done in the built-in coil editor; the geometry can be saved. The fields can be computed on single point, along a line or on a map with the coordinates loaded from a file. The fields can be computed using parallel processing option. The field distribution on the coils and particle tracking are options also available to the user.



Coil selection 1:2

Ready

Figure 5: SolCalc GUI main interface.

DISTRIBUTION

SolCalc package is distributed free of charge and follows the BSD license. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

SUMMARY

SolCalc is a useful set of Matlab functions that helps magnet designers and collaborators in the development of solenoid systems. The graphical capabilities of Matlab represent an attractive feature over other computer languages. Combined with the power of the Matlab Language, this considerably simplifies the optimization process of complex solenoid systems without any additional software.

SolCalc results show very good accuracy when compared to other commercial softwares with reasonable computing times. The use of multi-core CPUs can decrease computational time significantly. By ignoring fields from faraway coils, one sacrifices precision in exchange for speed. The sacrifice in precision is, however, as low as 3% according to the benchmark.

REFERENCES

[1] H. E. Knoepfel – “Magnetic Fields” – John Wiley & Sons, 2000.

[2] G. B. Arfken, H. J. Weber – “Mathematical Methods for Physicists”. Academic Press, 1996.

[3] W. H. Press et al., “Numerical Recipes”, Cambridge University Press, 2007.

[4] MATLAB 2013b, The MathWorks Inc., Natick, MA, 2012.

[5] M. L. Lopes et al., “Studies of the High-Field Section for a Muon Helical Cooling Channel”, Proc. of PAC2009, Vancouver, BC, Canada, 2009.

[6] Y. Derbenev and R. Johnson, “Six-Dimensional Muon Cooling Using a Homogeneous Absorber”, Phys. Rev. ST AB, 8, 041002 -2005.

[7] M. L. Lopes et al., “Magnetic Design Constrains of Helical Solenoids”, Proc. of IPAC2014, Dresden, Germany, 2014.

[8] “Mu2e Conceptual Design Report” - http://mu2e.fnal.gov/ - 2012.

[9] G. Ambrosio et al., “Challenges and Design of the Transport Solenoid for the Mu2e experiment”, IEEE Transactions on Applied Superconductivity, Vol. 24 No 3, 2014.

[10] M. Lopes et al., “Tolerance Studies of the Mu2e Solenoid System”, IEEE Transactions on Applied Superconductivity, Vol. 24 No 3, 2014.

[11] M. Lopes et al., “Studies on the Magnetic Center of the Mu2e Solenoid System”, IEEE Transactions on Applied Superconductivity, Vol. 24 No 3, 2014.

[12] R. Palmer et al., “Ionization Cooling Ring for Muons”, Phys. Rev. ST Accel. Beams 8, 061003, 2005.

[13] S. Feher et al., “Reference Design of the Mu2e Detector Solenoid”, IEEE Transactions on Applied Superconductivity, Vol. 24 No 3, 2014.

[14] Cobham/Vector Fields, Opera 3D Version 15 R13 http://cobham.vectorfields.com/ - 2012.

[15] Matlab Compiler Runtime www.mathworks.com/products/compiler/mcr/