CLAS NOTE 95-004

27 April 1995

Approximations to the CLAS Magnetic Field

M.V. Kossov, CNU/CEBAF/ITEP

1. Introduction

Historically, the magnetic field of the CLAS detector has been calculated by the subroutine TORUS which subdivides conductors into small straight elements. The goal of this note is to compare two approximation methods for the magnetic field. Both approximation algorithms use a data base consisting of the node values of the magnetic field, calculated by the TORUS subroutine. In the SDA package, a 2nd order approximation algorithm is used, and in the IFRAME package a linear approximation algorithm is used. These two algorithms will be compared to make a decision which algorithm should be used in CLAS GEANT simulation.

For the comparison three parameters were used:

1. The number of nodes in the data base for the approximation (the size);
2. The speed of the field approximation;
3. The accuracy of the $\int Bdl$ calculation in comparison with exact TORUS values.

2. 2-nd order approximation of CLAS magnetic field in SDA.

The SDA package uses the subroutine SNAKE, which performs a 3-dimensional 2nd order interpolation for each component of the magnetic field using Taylor expansion. For the data base the node's values of the cubic 3-dimensional table are:

1. 61 nodes along the X-axis (along a mid.plane perpendicular to the beam direction) with 7 cm step (X_{min} = -7 cm, X_{max} = 420 cm);
2. 41 nodes along the Y-axis (along the wires of the CLAS Drift Chamber) with 7 cm step (Y_{min} = -7 cm, Y_{max} = 280 cm);
3. 87 nodes along the Z-axis (along the ideal beam direction) with 10 cm step (Z_{min} = -350 cm, Z_{max} = 520 cm).

The SNAKE algorithm uses 8 points, which define a cube around the space point, where the field value is needed: (i,i,i), (i+1,i,i), (i,i+1,i), (i,i,i+1), (i,i+1,i+1), (i+1,i,i+1), (i+1,i+1,i), (i+1,i+1,i+1). In addition, three upstream points (i-1,i,i), (i,i-1,i), (i,i,i-1) are used. Because of these upstream points the data base needs additional points with negative values of X and Y. The algorithm is as follows:

xx=x-$x_{i,i,i}$
yy=y-$y_{i,i,i}$
zz=z-$z_{i,i,i}$
bx=$f_{i+1,i,i}$-$f_{i-1,i,i}$
by=$f_{i,i+1,i}$-$f_{i,i-1,i}$
bz=$f_{i,i,i+1}$-$f_{i,i,i-1}$
ff=$f_{i,i,i}+f_{i,i,i}$
cx=$f_{i+1,i,i}+f_{i-1,i,i}$-ff
cy=$f_{i,i+1,i}+f_{i,i-1,i}$-ff
cz=$f_{i,i,i+1}+f_{i,i,i-1}$-ff
xy=$f_{i,i,i}$-$f_{i+1,i,i}+f_{i+1,i+1,i}$-$f_{i,i+1,i}$
yz=$f_{i,i,i}$-$f_{i,i+1,i}+f_{i,i+1,i+1}$-$f_{i,i,i+1}$
zx=$f_{i,i,i}$-$f_{i,i,i+1}+f_{i+1,i,i+1}$-$f_{i+1,i,i}$
dd=$f_{i+1,i+1,i+1}$-$f_{i+1,i,i+1}+f_{i,i,i+1}$-$f_{i,i+1,i+1}$-$f_{i,i,i}+f_{i+1,i,i}$ -$f_{i+1,i+1,i}+f_{i,i+1,i}$
f=$f_{i,i,i}$+(xx· (bx+xx· cx)+yy· (by+yy· cy)+zz· (bz+zz· cz) +xx·yy·xy+yy·zz·yz+zz·xx·zx+xx·yy·zz·dd)/2

An obvious disadvantage of this algorithm is the large number of dummy nodes in case of the 60 degrees sector. The cross section of the grid in a plane perpendicular to the beam is a rectangle, while a cross section of the sector is a triangle. That is why at least half of nodes are dummy.

3. Linear approximation of CLAS magnetic field in IFRAME.

The IFRAME package uses the algorithm of the FINT function (CERNLIB). The existing FINT algorithm is optimized for the constant step (it is arbitrary in the original function) and for 3 dimensions (the original function can work with up to 5 dimensions). The FINT algorithm performs a simple first order interpolation. For the data base the following nodes were used:

1. 17 nodes for the $tg(\phi)$ subdivision with a step 0.03625 ($tg(\phi)_{min} = 0$, $tg(\phi)_{max} = 0.58$);
2. 129 nodes along the Z-axis (along the ideal beam direction) with 7 cm step (Z_{min} = -360 cm, Z_{max} = 535 cm).
3. 65 nodes along the X-axis (along a middle plane perpendicular to the beam direction) with 7 cm step ($X_{min} = 0$ cm, $X_{max} = 448$ cm).

The IFRAME data base covers a 15% larger volume than the SDA data base, while the number of nodes is 1.53 times smaller than in SDA data base.

The FINT algorithm uses only 8 points, which define a cube around the space point, where the field value is needed. The algorithm for the calculation is as follows:
xx=x-$x_{i,i,i}$
yy=y-$y_{i,i,i}$
zz=z-$z_{i,i,i}$

dx=$x_{i+1,i,i}$-$x_{i,i,i}$
dy=$y_{i,i+1,i}$-$y_{i,i,i}$
dz=$z_{i,i,i+1}$-$z_{i,i,i}$
xn=xx/dx
yn=yy/dy
zn=zz/dz
xm=1.-xn
ym=1.-yn
zm=1.-zn
fy$_{i,i}$=$f_{i,i,i}$·xm+$f_{i+1,i,i}$·xn
fy$_{i,i+1}$=$f_{i,i,i+1}$·xm+$f_{i+1,i,i+1}$·xn
fy$_{i+1,i}$=$f_{i,i+1,i}$·xm+$f_{i+1,i+1,i}$·xn
fy$_{i+1,i+1}$=$f_{i,i+1,i+1}$·xm+$f_{i+1,i+1,i+1}$·xn
fz$_i$=fy$_{i,i}$·ym+fy$_{i+1,i}$·yn
fz$_{i+1}$=fy$_{i,i+1}$·ym+fy$_{i+1,i+1}$·yn
f=fz$_i$·zm+fz$_{i+1}$·zn

3. Speed and accuracy.

The linear IFRAME algorithm is obviously faster than the 2-nd order SDA algorithm. Measurements on a VAX computer show that the field extraction by the SNAKE subroutine requires 2 msec while the extraction by the modified FINT subroutine requires 1.25 msec. That is a linear algorithm is factor of 1.6 faster than the 2nd order algorithm.

The accuracy of the calculations was compared using $\int Bdl$ calculations for the exact TORUS magnetic field values and for approximated values. The $\int Bdl$ integrations were made along directions with random θ and ϕ. For most of the points (67%) the accuracy of the $\int Bdl$ integration is better then 1% for both algorithms. In case the accuracy is worse than 1% for at least one projection, the resulting relative differences are listed in the table.

One can see, that only at very small angles (θ=11.48 in Table 1) and at very large angles (θ=137.31 in Table 1) IFRAME algorithm gives worse results. In all other cases it seems to be even more accurate than the SDA algorithm.

It should be noted, that if the same magnetic field approximation is used in a simulation and in a reconstruction, the relative systematic errors in the acceptances will be much smaller than the relative errors in $\int Bdl$ integrals.

Table 1. **Fractional accuracies of $\int Bdl$ for the SDA and IFRAME methods**

θ(deg)	ϕ(deg)	Bx(SDA)	Bx(IFR)	By(SDA)	By(IFR)	Bz(SDA)	Bz(IFR)
11.48	7.97	.0601	.0701	.0008	-.0073	-.0031	-.0090
11.48	9.03	.0426	.0714	.0005	-.0072	-.0007	-.0095
11.48	9.54	.0282	.0780	.0004	-.0072	.0020	-.0096
11.48	12.13	.1029	.1748	.0020	-.0077	.0084	-.0101
11.48	24.18	.1435	.0211	-.0058	-.0091	.0124	.0119
62.23	1.89	-.0073	.0001	-.0012	.0002	.0177	.0001
62.23	2.41	-.0063	-.0002	-.0012	.0001	.0172	-.0002
71.88	7.46	-.0218	-.0069	-.0022	-.0006	.0166	.0050
71.88	10.98	-.0176	-.0057	-.0008	-.0002	.0139	.0042
71.88	14.57	-.0122	-.0040	.0008	.0004	.0102	.0033
72.72	6.28	-.0232	-.0082	-.0027	-.0008	.0154	.0060
72.72	15.70	-.0110	-.0043	.0011	.0004	.0085	.0031
73.20	2.86	-.0247	-.0096	-.0037	-.0013	.0159	.0055
73.20	11.86	-.0166	-.0060	-.0005	-.0001	.0109	-.0039
73.20	12.02	-.0163	-.0058	-.0004	-.0001	.0108	.0039
79.28	12.73	-.0197	-.0080	-.0001	.0000	.0027	.0013
79.28	16.49	-.0125	-.0050	.0016	.0008	.0017	.0010
79.28	16.75	-.0120	-.0050	.0017	.0008	.0017	.0008
90.00	1.03	-.0397	-.0173	-.0052	-.0022	-.0159	-.0063
90.00	7.24	-.0333	-.0147	-.0032	-.0013	-.0132	-.0054
90.00	12.19	-.0237	-.0099	-.0004	.0002	-.0100	-.0036
94.30	0.92	-.0370	-.0165	-.0047	-.0021	-.0188	-.0085
94.30	3.26	-.0358	-.0160	-.0044	-.0020	-.0186	-.0082
94.30	14.09	-.0185	-.0081	.0007	.0003	-.0111	-.0042
94.30	15.11	-.0163	-.0077	.0011	.0004	-.0101	-.0043
95.62	16.86	-.0124	-.0047	.0018	.0007	-.0094	-.0032
96.26	4.69	-.0295	-.0127	-.0034	-.0014	-.0174	-.0073
96.26	12.79	-.0177	-.0078	-.0001	.0001	-.0120	-.0049
96.26	15.16	-.0136	-.0062	.0009	.0005	-.0099	-.0042
96.89	13.98	-.0162	-.0071	.0004	.0003	-.0113	-.0045
110.18	8.87	-.0105	-.0032	-.0007	-.0003	-.0096	-.0020
111.96	26.89	.0390	-.0029	.0358	-.0002	.0066	.0046
122.28	2.12	-.0150	-.0052	.0000	-.0005	.0016	.0017
132.22	17.22	.0122	-.0083	-.0011	-.0024	.0038	-.0014
132.22	20.41	.0218	-.0005	-.0003	-.0043	.0066	-.0003
132.22	25.69	-.0863	-.0522	.0154	.0051	.0154	.0151
137.31	25.64	.0098	-.0164	-.0018	-.0123	-.0035	-.0154