```
1
2   /*****************************************************************************
3    * Package: GSM                                                              *
4    * Class  : DAlphaQED                                                        *
5    *                                                                           *
6    * Description:                                                              *
7    *      Auxiliary Theory of the Delta-Alpha-QED at the Z-pole                *
8    *      leptonic contribution to effective em coupling constant              *
9    *      up to three loops                                                    *
10   *                                                                           *
11   * Sources:                                                                  *
12   *      (1) M. Steinhauser, Phys. Lett. B429 (1998) 158-161, hep-ph/9803313  *
13   *      (2) J.H. Kuehn, M. Steinhauser, Phys. Lett. B437 (1998) 425-431      *
14   *                                  Hep-ph/9802241                           *
15   *                                                                           *
16   *****************************************************************************/
17  #include "TMath.h"
18
19  #include "Gfitter/GMath.h"
20  #include "Gfitter/GConstants.h"
21  #include "Gfitter/GTheory.h"
22  #include "Gfitter/GTheoryRef.h"
23  #include "Gfitter/GParameterRef.h"
24  #include "Gfitter/GReference.h"
25
26  #include "GSM/DAlphaQED.h"
27  #include "GSM/RunningAlphaQCD.h"
28
29  #include "GSM/ZMath.h"
30
31  #include "Riostream.h"
32
33  using namespace Gfitter;
34  using std::complex;
35
36  ClassImp(GSM::DAlphaQED)
37
38  GSM::DAlphaQED::DAlphaQED()
39     : Gfitter::GAuxTheory(),
40       m_isUpToDate_Update( kFALSE )
41  {
42     SetTheoryName( GetName() );
```

Hinweis:

Kommentare mit Hinweisen auf ZFitter sind grün markiert

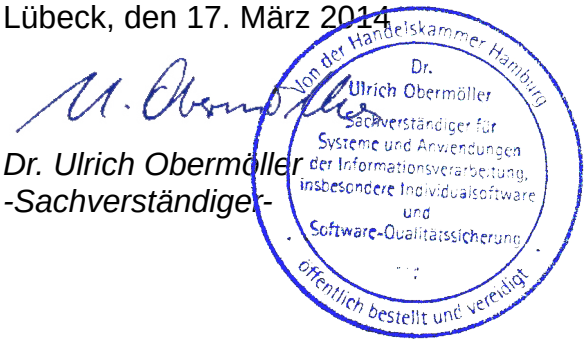
Übereinstimmungen sind gelb markiert.

Auffällige Stellen bzw. Anmerkungen sind violett markiert

Anhang 1

zum Gutachten DESY ZFitter_GFitter vom 17.03.2014

Lübeck, den 17. März 2014



Dr. Ulrich Obermöller
-Sachverständiger-

```
43      SetExistDerivative( kFALSE );
44
45      BookParameter( "MZ"        , &p_MZ );
46      BookParameter( "mt"        , &p_mt );
47      BookParameter( "DAlphaHad" , &p_DAlphaHad );
48      //   BookParameter( "GSM::MWTree"   , &p_MW );
49
50      BookTheory   ( "GSM::RunningAlphaQCD" , &t_AlphasRun );
51    }
52
53    void GSM::DAlphaQED::Initialise()
54    {
55      // use calculation from Steinhauser and Kuehn
56      m_Zfitter = kFALSE;
57
58      // lepton masses
59      m_mLep2[0] = GMath::IPow( GConstants::me()  , 2 );
60      m_mLep2[1] = GMath::IPow( GConstants::mmu() , 2 );
61      m_mLep2[2] = GMath::IPow( GConstants::mtau(), 2 );
62
63      m_alp1pi   = GConstants::alphaQED()/TMath::Pi();
64    }
65
66    void GSM::DAlphaQED::UpdateLocalFlags( GReference& /* ref */ )
67    {
68      m_isUpToDate_Update = kFALSE;
69    }
70
71    // Update parmeters and check if something has changed
72    void GSM::DAlphaQED::Update()
73    {
74      if (m_isUpToDate_Update) return;
75
76      // now, it is uptodate (I mean... it will be)
77      m_isUpToDate_Update = kTRUE;
78
79      Double_t MZ2   = p_MZ*p_MZ;
80      Double_t mt2   = p_mt*p_mt;
81
82      Double_t chQ21 = DAlphaLep(MZ2);
83
84      // excluded top-contribution
```

```
85      m_DAlphaQED    = chQ21 + p_DAlphaHad;
86
87      // included top-contribution
88      // ZFitter method
89      if ( m_Zfitter ) {
90        chQ21        += 2.0*m_alp1pi*3.0*4/9.0*real( GSM::ZMath::I3(mt2, -MZ2 ,mt2, mt2) ); || ~ keine Übereinstimmung in ZFitter gefunden
91        m_DAlphaQEDtop = chQ21 + p_DAlphaHad + AlQCD()*m_alp1pi/4.0;
92      }
93      // Steinhuser and Kuehn
94      else  m_DAlphaQEDtop = chQ21 + p_DAlphaHad + DAlphatop() ;
95
96      SetUpToDate();
97    }
98
99    // leptonic contribution
100   // eq. (1) of hep-ph/9803313
101   // and sub-equation (5) - (10)
102   Double_t GSM::DAlphaQED::DAlphaLep( Double_t s )
103   {
104     Double_t dalpha = 0;
105     Double_t sum    = 0;
106     Double_t Const  = 0;
107
108     for (Int_t i = 0; i < 3; i++) { || ~ line 5564
109       Double_t Ratio = m_mLep2[i]/s;
110       Double_t LogR  = -TMath::Log( Ratio );
111
112       // ZFitter method
113       // comming from dizet_6_42.f (ZFitter) line 5667
114       if ( m_Zfitter ) {
115         dalpha += ( m_alp1pi*( 1/9.0 + 1/3.0*(1.0 + 2.0*Ratio)
116                               *real( GSM::ZMath::I0(m_mLep2[i], -s , m_mLep2[i], m_mLep2[i]) ) ) ); || ~ line 5568/5569
117       }
118       else {
119         // PI(0) eq. (5) + some 1-loop corrections,
120         dalpha += -0.25*m_alp1pi*(20/9.0 - 4/3.0*LogR + 8.0*Ratio);
121       }
122
123       // PI(1) eq. (6)
124       dalpha += ( m_alp1pi*m_alp1pi*((1.0 + 12.0*Ratio)/4.0*LogR + GMath::Zeta3() - 5/24.0) ); || ~ line 5570
125       // PI(2)A eq. (7)
126       dalpha += 2.0*( GMath::IPow(m_alp1pi, 3)*(-0.25*(-121/48.0 + (-5.0 + 8.0*TMath::Log(2.0))*GMath::Zeta2()
```

```
127                                  -99/16.0*GMath::Zeta3() + 10.0*GMath::Zeta5() + 1/8.0*LogR)) ); || ~ line 5572/5573
128
129       for (Int_t j = 0; j < 3; j++) {
130         Double_t Logi = TMath::Log( s/m_mLep2[i] );
131         Double_t Logj = TMath::Log( s/m_mLep2[j] );
132
133         if (i == j) {
134           // PI(2)F eq. (9)
135           sum += ( GMath::IPow(m_alp1pi, 3)
136                  *(-0.25*(- 307/216.0 - 8/3.0*GMath::Zeta2() + 545/144.0*GMath::Zeta3()
137                         +(11/6.0 - 4/3.0*GMath::Zeta3())*Logi - 1/6.0*Logi*Logi + Const)) );
138         }
139         else if( i > j) {
140           // PI(2)l eq. (8)
141           sum += ( GMath::IPow(m_alp1pi, 3)
142                  *(-0.25*( -116/27.0 + 4/3.0*GMath::Zeta2() + 38/9.0*GMath::Zeta3()
143                          + 14/9.0*Logi + (5/18.0 - 4/3.0*GMath::Zeta3())*Logj
144                          + 1/6.0*Logi*Logi - 1/3.0*Logi*Logj + Const)) );
145         }
146         else {
147           // PI(2)h eq. (10)
148           sum += ( GMath::IPow(m_alp1pi, 3)
149                  *(-0.25*(- 37/6.0 + 38/9.0*GMath::Zeta3()
150                          + (11/6.0 - 4/3.0*GMath::Zeta3())*Logj - 1/6.0*Logj*Logj + Const)) );
151         }
152       }
153     }
154
155     return dalpha + sum;
156   }
157
158
159   // for DAlphaLept
160   Double_t GSM::DAlphaQED::AlQCD()
161   {
162     // sw2 = 1 - MW^2/MZ^2
163     // Attention: MW depends on DAlphaQED
164     // => DAlphaQED -> MW -> DAlphaQED !!!
165     // We use the measured value from the steering card
166     // The obtained error is small because AlQCD is multiplied by a small value ( line 93 )
167
168     if (TMath::IsNaN(p_mt)) m_logger << kFATAL << "<DAlphaQED::AlQCD> p_mt is NaN !" << GEndl;
```

Match 1

```
5564        DO I=1,3
5565        AML2=(AML(2*I))**2
5566        RML2=AML2/S
5567        ALG1=LOG(S/AML2)
5568        PI0(I)=1D0/ALFAI/PI*(1D0/9+1D0/3*(1D0+2D0*RML2)
5569       &      *DREAL(XI0(AML2,-S,AML2,AML2)))
5570        PI1(I) =(1D0/ALFAI/PI)**2*((1D0+12D0*RML2)/4*ALG1+D3-5D0/24)
5571        PI1L(I)=(1D0/ALFAI/PI)**2*( 1D0/4*ALG1)
5572        PI2A(I)=(1D0/ALFAI/PI)**3*(-1D0/4*(-121D0/48
5573       &       +(-5D0+8D0*LOG(2D0))*D2-99D0/16*D3+10D0*D5+1D0/8*ALG1))
5574        ENDDO
5575      *
5576        DO I=1,3
5577        SUM=0D0
5578        DO J=1,3
5579        ALG1=LOG(S/(AML(2*I))**2)
5580        ALG2=LOG(S/(AML(2*J))**2)
5581        PI2F(I)=(1D0/ALFAI/PI)**3*(-1D0/4*(-307D0/216-8D0/3*D2
5582       &       +545D0/144*D3+(11D0/6-4D0/3*D3)*ALG1-1D0/6*ALG1**2+CONST))
5583        PI2L(I,J)=(1D0/ALFAI/PI)**3*(-1D0/4*(-116D0/27+4D0/3*D2
5584       &          +38D0/9*D3+14D0/9*ALG1+(5D0/18-4D0/3*D3)*ALG2
5585       &          +1D0/6*ALG1**2-1D0/3*ALG1*ALG2+CONST))
5586        PI2H(J)=(1D0/ALFAI/PI)**3*(-1D0/4*(-37D0/6+38D0/9*D3
5587       &       +(11D0/6-4D0/3*D3)*ALG2-1D0/6*ALG2**2+CONST))
5588        IF(I.LT.J) THEN
5589          PI2B(I,J)=PI2H(J)
5590        ELSEIF(I.EQ.J) THEN
5591          PI2B(I,J)=PI2F(I)
5592        ELSEIF(I.GT.J) THEN
5593          PI2B(I,J)=PI2L(I,J)
5594        ENDIF
5595        SUM=SUM+PI2B(I,J)
5596        ENDDO
5597        PI2S(I)=PI2A(I)+SUM
5598      *
5599        IF(IALE2.EQ.0) THEN
5600          PISU(I)=PI0(I)+PI1L(I)
5601        ELSEIF(IALE2.EQ.1) THEN
5602          PISU(I)=PI0(I)
5603        ELSEIF(IALE2.EQ.2) THEN
5604          PISU(I)=PI0(I)+PI1(I)
5605        ELSEIF(IALE2.EQ.3) THEN
5606          PISU(I)=PI0(I)+PI1(I)+PI2A(I)+PI2S(I)
5607        ENDIF
5608      *
5609        ENDDO
5610      *
5611        DALPHL=PISU(1)+PISU(2)+PISU(3)
5612      *
5613        END
```

```
169
170     Double_t sw2   = 1.0 - p_MW->GetValue()*p_MW->GetValue()/(p_MZ*p_MZ);
171     Double_t pvftz = -p_mt*p_mt/(p_MZ*p_MZ)*real( GSM::ZMath::PolVfi(-p_mt*p_mt/(p_MZ*p_MZ)) );
172     Double_t alqcd = ( -GetAlphasRun().EvolveAlphas(p_mt)/(3.0*TMath::Pi()*sw2)
173                        *(4.0*sw2*4/9.0*( pvftz - 45/4.0 )) );
174     return alqcd;
175   }
176
177
178   // eq. (12) from hep-ph/9802241
179   Double_t GSM::DAlphaQED::DAlphatop()
180   {
181     Double_t alphaspi = GetAlphasRun().EvolveAlphas(p_MZ)/TMath::Pi();
182     Double_t L        = TMath::Log((p_MZ*p_MZ)/(p_mt*p_mt));
183
184     Double_t atop = ( -4/45.0*m_alp1pi*(p_MZ*p_MZ)/(p_mt*p_mt)*
185                       (1.0 + 5.062*alphaspi + (28.220 + 9.702*L)*alphaspi*alphaspi
186                        + (p_MZ*p_MZ)/(p_mt*p_mt)*(0.1071 + 0.8315*alphaspi + (6.924 + 1.594*L)*alphaspi*alphaspi)) );
187     return atop;
188   }
189
```