AN ALGORITHM TO GENERATE MODULATION TRANSFER FUNCTION OF SEAWATER FROM SCATTERING OR BEAM ATTENUATION COEFFICIENT AT GIVEN WAVELENGTH

Vladimir I. Haltrin, Walton E. McBride III, and Alan D. Weidemann*
Naval Research Laboratory, Ocean Optics Section, Code 7333
Stennis Space Center, MS 39529-5004, USA. e-mail: <haltrin@nrlssc.navy.mil>
** Planning Systems Inc., 115 Christian Lane, Slidell, LA 70458, USA*

INTRODUCTION

Few analytical formulations for the Modulation Transfer Function (MTF) of turbid waters exist in the literature. For almost 30 years, the MTF formulation due to Wells has been the choice of many researchers. Wells' MTF uses the scattering coefficient and a parameter he called the mean scattering angle. Recently, two new analytical formulations due to McLean et al. [1] and Fournier and Jonasz [2] have been published in the literature. However, these MTFs require angular statistics of the scattering process such as averages squares of cosine and scattering angle of the scattering phase function.

We present an algorithm to generate these statistics from scattering or the beam attenuation coefficient, at a particular wavelength. The algorithm is implemented as a working FORTRAN code that outputs the angular statistics and MTFs according to [1, 2]. The basis of the proposed algorithm is a variant of chlorophyll dependent model of seawater optical properties developed by Haltrin [3] and experimental measurements of the fifteen Petzold [4] and eight Mankovsky [5, 6] phase functions. The algorithm provides a self-consistent link between the input parameters that are often available from routine measurements and the angular statistics used in underwater light propagation.

PHASE FUNCTIONS AND ANGULAR STATISTICS

Both Petzold [4] and Mankovsky [5] normalized experimental phase functions with a high level of precision may be represented as following regressions:

$$p(\vartheta) = \exp\left(\sum_{i=0}^{7} c_i \vartheta^{i/2}\right), \quad 0.5\int_0^{\pi} p(\vartheta)\sin\vartheta\, d\vartheta = 1, \quad r^2 > 0.999, \qquad (1)$$

here ϑ is a scattering angle, and coefficients c_i are given in Table 1. Because all phase functions used to generate Eq. (1) have associated values of scattering b and beam attenuation c coefficients [6], it is possible to connect computed phase function statistics with inherent properties of seawater. The resulting equations that connect these statistics with a single scattering albedo of seawater $\omega_0 = b/(a+b)$ are as follows:

Presented to the Ocean Optics XV, Musée Océanographique, Monaco, October 16-20, 2000

Table 1. Coefficients to Eq. (1). Associated values of b and c are given in Refs. [4] and [6].

phf #	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	r^2
p01	7.5549	-56.190	141.48	-213.65	178.65	-80.92	18.105	-1.4299	0.9999
p02	5.5247	-46.329	108.10	-155.16	121.18	-49.87	9.9414	-0.6985	0.9999
p03	5.9621	-48.693	112.34	-154.78	112.87	-40.915	6.0502	-0.0721	0.9999
p04	8.5668	-55.930	144.71	-236.78	221.69	-117.19	32.533	-3.6095	0.9999
p05	8.2986	-57.683	162.99	-296.38	314.16	-192.45	63.516	-8.6944	0.9999
p06	10.478	-51.226	116.96	-170.64	143.33	-67.081	15.819	-1.3336	0.9999
p07	10.668	-51.842	124.05	-198.15	191.80	-110.40	35.035	-4.6975	0.9999
p08	9.9390	-53.588	146.42	-268.75	294.30	-187.62	64.352	-9.1335	0.9999
p09	2.1180	-30.824	66.308	-96.209	66.681	-12.518	-6.1220	2.2361	0.9999
p10	6.3326	-26.145	45.989	-72.415	71.641	-43.003	14.843	-2.2221	0.9998
p11	6.4591	-27.086	50.34	-80.626	77.087	-42.462	12.987	-1.7158	0.9999
p12	9.1431	-34.948	41.352	-8.754	-38.729	42.586	-17.223	2.5005	0.9999
p13	8.7047	-50.967	119.87	-178.56	153.52	-75.886	20.375	-2.3001	0.9998
p14	6.8425	-49.294	109.37	-144.45	98.400	-30.520	2.2915	0.4647	0.9999
p15	2.3685	-53.860	210.02	-457.08	531.76	-337.96	111.52	-14.983	0.9988
m01	3.3253	29.068	-312.19	889.14	-1249.9	932.50	-352.84	53.352	0.9994
m02	2.8672	16.422	-205.20	582.16	-814.84	605.88	-228.66	34.530	0.9996
m03	2.4312	14.049	-187.34	514.22	695.87	502.83	-184.97	27.236	0.9991
m04	2.7770	26.003	-340.96	1005.0	-1413.2	1045.0	-391.24	58.554	0.9991
m05	1.8298	23.983	-302.33	885.71	-1246.9	925.02	-347.71	52.258	0.9993
m06	2.2317	19.850	-186.58	501.27	-693.17	517.0	-196.90	30.104	0.9995
m07	1.9491	11.666	-129.42	330.5	-434.53	309.12	-112.18	16.307	0.9995
m08	4.0230	5.8845	-140.44	375.5	-475.30	315.33	-105.28	13.963	0.9993

$$\langle \cos\vartheta \rangle = 0.98572\omega_0^{0.12226}, \quad r^2 = 0.8431, \tag{2}$$

$$\langle \cos^2\vartheta \rangle = 0.96011\omega_0^{0.079212}, \quad r^2 = 0.8099, \tag{3}$$

$$\langle \vartheta \rangle = 0.15064\,\omega_0^{-0.52447}, \quad r^2 = 0.9067, \tag{4}$$

$$\langle \vartheta^2 \rangle = 0.113\,\omega_0^{-0.81265}, \quad r^2 = 0.9557. \tag{5}$$

The phase function statistics given by Eqs. (2)-(5) may be used to express modulation transfer function through only two inherent optical properties: scattering and beam attenuation coefficients.

A PROGRAM TO CALCULATE STATISTICS AND MODULATION TRANSFER FUNCTIONS

Appendix A presents a FORTRAN program to compute statistics given by Eqs. (2)-(5) and modulation transfer functions by McLean et. al [1] {Eq. (A5)} and Fournier and Jonasz [2] {Eq. (20). The input parameters are either scattering or attenuation coefficient and a wavelengh of light at which this coefficient is measured. To restore a single-scattering albedo at 515 nm, used in regressions (2)-(5), we employed a modification of an approach presented earlier in Refs. [3] and [7]. The output files for MTFs are formatted as input files for Transform, a 2-dimensional plotting .hdf software from Fortner Research and Research Systens Inc. (Transform is a part of Noesys software suit, see: <http://www.rsinc.com/noesys/index.cfm>).

ACKNOWLEDGMENTS

The authors from NRL thanks continuing support at the Naval Research Laboratory through the Littoral Optical Environment (LOE 6640-09) and Optical Oceanography (OO 73-5051-09) programs. This article represents NRL contribution PP/7333–00–0015.

REFERENCES

1. J. W. McLean, J. D. Freeman, and R. Walker, "Beam spread function with time dispersion," *Appl. Optics*, **37**, 4701-4711 (1998).

2. G. R. Fournier and M. Jonasz, "Computer based underwater imaging analysis," *SPIE*, **3761**,. 62-70 (1999).

3. V. I. Haltrin, "Chlorophyll-based model of seawater optical properties," *Appl. Optics,* **38**, 6826-6832 (1999).

4. T. J. Petzold, *Volume Scattering Functions for Selected Ocean Waters*, SIO Ref. 72-78, Scripps Institute of Oceanography, Visibility Laboratory, San Diego, CA, USA, pp. 79 (1972).

5. V. I. Mankovsky, *Private communication* , Marine Hydrophysical Institute, Sevastopol, Crimea (1995) [e-mail: <ocean@mhi2.sebastopol.ua>, Re.: Man'kovsky].

6. V. I. Haltrin, "Theoretical and empirical phase-functions for Monte-Carlo calculations of light scattering in sea water." in: *Proceedings of the Fourth International Conference Remote Sensing for Marine and Coastal Environments: Technology and Applications*," **I**, Publication by Environmental Research Institute of Michigan, Ann Arbor, Michigan, 509-518 (1997).[Errata: in Eq.(45) e should be replaced by 10].

7. V. I. Haltrin, "An algorithm to restore spectral signatures of all inherent optical properties of seawater using a value of one property at one wavelength," in *Proceedings of the Fourth International Airborne Remote Sensing Conference and Exhibition*, **I**, ERIM International, Ann Arbor, MI, USA, I-680-687 (1999).

APPENDIX A:

A FORTRAN PROGRAM GETMTF.F TO CALCULATE STATISTICS AND MTFS.

```
c ====================================================================
c                        program getMTF
c        written by Vladimir I. Haltrin, NRL SSC Code 7333
c                   <haltrin@nrlssc.navy.mil>
c ====================================================================

      implicit   none
      integer    Nlm,Nfm,Nl,Nf,i,j,scat
      parameter  (Nlm=100, Nfm=100)
      real       lam,b,c,omeg,cb,taul,avth,avth2,avcos,avcos2,ssalb
      real       Cc, fCcb, fCcc, k,mu, fSsalb, FourJonMTF,McLeanoMTF
      real       fjMTF(Nlm,Nfm),mlMTF(Nlm,Nfm),tau(Nlm),fr(Nfm)
      character  tb
      open(11,file='getMTF.in',status='old')
        read(11,*) scat      ! read flag: = 1 if cb=b; = 2 if cb=c
        read(11,*) cb        ! read b or c depending on scat flag
        read(11,*) lam       ! read wavelength in nm
        read(11,*) Nl        ! read a number of viewing lengths for MTF
        read(11,*)
        read(11,*) (tau(i),i=1,Nl)  ! read array of optical lengths
        read(11,*) Nf               ! read a number of frequencies
        read(11,*)
        read(11,*) (fr(i),i=1,Nf)   ! read an array of MTF frequences
      close(11)
      if (scat .eq. 1) then
         b = cb
         Cc = fCcb(lam,b)
         ssalb = fSsalb(Cc)
         c = b/ssalb
      else
         c = cb
         Cc = fCcc(lam,c)
         ssalb = fSsalb(Cc)
         b = c*ssalb
      end if
      call getStats(ssalb, avth,avth2,avcos,avcos2)  ! calc. statistics
      k  = 0.0        ! special case for McLean MTF
      mu = 0.0        ! special case for McLean MTF
      do i=1, Nl
         taul = tau(i)
         do j=1, Nf
            omeg = fr(j)
            fjMTF(i,j) = FourJonMTF(b,c,taul,omeg)
            mlMTF(i,j) = McLeanoMTF(b,c,taul,omeg,k,mu)
         end do
      end do
      tb = CHAR(9)
      open(21,file='Stats.out',status='new')  ! output with statistics
         write(21,33) 'Scattering coefficient:    ', b,' 1/m'
         write(21,33) 'Attenuation coefficient:   ', c,' 1/m'
         write(21,55) 'Single scattering albedo:    ', ssalb
         write(21,44) 'Wavelength of light:        ', lam,' nm'
         write(21,33) 'Chlorophyll concentration: ', Cc,' mg/m^3'
         write(21,55) 'Average cosine:              ', avcos
```

```
      write(21,55) 'Average square of cosine:   ', avcos2
      write(21,55) 'Average angle:              ', avth
      write(21,55) 'Average square of angle:    ', avth2
   close(21)
   open(22,file='FourJonMTF.hdf',status='new',recl=1300)
      write(22,77) Nl,tb,Nf
      write(22,77) 0,tb,0
      write(22,88) tau(1),(tb,tau(i),i=2,Nl)
      write(22,88) fr(1), (tb,fr(j), j=2,Nf)
      do i = 1, Nl
         write(22,88) fjMTF(i,1),(tb,fjMTF(i,j),j=2,Nf)
      enddo
   close (22)
   open(23,file='McLeanoMTF.hdf',status='new',recl=1300)
      write(23,77) Nl,tb,Nf
      write(23,77) 0,tb,0
      write(23,88) tau(1),(tb,tau(i),i=2,Nl)
      write(23,88) fr(1), (tb,fr(j), j=2,Nf)
      do i = 1, Nl
         write(23,88) mlMTF(i,1),(tb,mlMTF(i,j),j=2,Nf)
      enddo
   close (23)
33 format(a,f7.3,a)
44 format(a,f7.2,a)
55 format(a,g12.6)
77 format(i5,a1,i5)
88 format(g12.5,99(a1,g12.5))
   end

   real function fCcb(lam,b)   ! calculates chlorophyll conc. via b
   real      lam,b, x,c1,c2,c3          ! This is a modification of
   x = (lam-500.)/500.                  ! subroutine of Ref. [7]
   c1 = (( 3.743707E-1*x-6.845814E-1)*x+ 1.4658230)*x+3.824251
   c2 = ((-3.948394E-2*x+7.330175E-2)*x-0.26272660)*x-0.4134296
   c3 = (( 1.172611E-3*x-2.296292E-3)*x+0.02819422)*x+0.02742180
    x = 400./lam
    x = b - (5.826E-3)*(x*x*x*x)*(x**0.322)        ! x = b-bw
   fCcb = x*(c1+x*(c2+x*c3))
   return
   end

   real function fCcc(lam,c)   ! calculates chlorophyll conc. via c
   implicit  none                       ! This is a modification of
   real      lam,c, x,c1,c2,c3, faw     ! subroutine of Ref.[7]
   x = (lam-500.)/500.
   if (lam.lt.670.) then
      c1 = (((-1.494882E+1*x-1.026426E+1)*x-1.061336E+0)*x
  &                         +3.875227E+0)*x+3.199760
      c2 = ((( 1.075787E+1*x+2.933227E+0)*x-7.986360E-1)*x
  &                         -5.943025E-1)*x-2.823301E-1
      c3 = (((-1.583553E+0*x-2.482009E-1)*x+1.312316E-1)*x
  &                         +4.842813E-2)*x+1.312129E-2
   else
      c1 = ((( 1.795351E+4*x-2.841010E+4)*x+1.678696E+4)*x
  &                         -4.384514E+3)*x+4.306481E+2
      c2 = (((-7.477734E+3*x+1.184332E+4)*x-7.005140E+3)*x
  &                         +1.832100E+3)*x-1.789934E+2
      c3 = ((( 9.321093E+2*x-1.477465E+3)*x+8.746930E+2)*x
  &                         -2.289986E+2)*x+2.237217E+1
   end if
     x = 400./lam
```

```
  x = (5.826E-3)*(x*x*x*x)*(x**0.322)          ! x = bw
  x = c-x-faw(lam)                             ! x = c-cw
fCcc = x*(c1+x*(c2+x*c3))
return
end

real function faw(lam)  ! calculates pure water absorption coef.
integer   k             ! via simple interpolation
real      lam, aw0,lmk,aw(35)
data      aw /0.01137,0.00851,0.00663,0.00473,0.00454,0.00495,
&             0.00635,0.00922,0.00979,0.0106,0.0127,0.015,0.0204,
&             0.0325,0.0409,0.0434,0.0474,0.0565,0.0619,0.0695,
&             0.0896,0.1351,0.2224,0.2644,0.2755,0.2916,0.3108,
&             0.34,0.41,0.439,0.465,0.516,0.624,0.827,1.231/
k = 0.1*(lam-370.)
lmk = 370.+10.*k
aw0 = aw(k)
faw = aw0 + 0.1*(lam-lmk)*(aw(k+1)-aw0)
return
end

real function fSsalb(Cc)   ! calculates single scattering albedo
real      Cc,Cf,Ch,Cs,Cl,Achl,lam,ay,bs,bl,aw,a,b
data      lam /515.0/, aw /0.0367/, Achl /0.573/
Cf = 1.74098*Cc*EXP(0.12327*Cc)
Ch = 0.19334*Cc*EXP(0.12343*Cc)
Cs = 0.01739*Cc*EXP(0.11631*Cc)
Cl = 0.76284*Cc*EXP(0.03092*Cc)
ay = Cf*35.959*EXP(-0.0189*lam)+Ch*18.828*EXP(-0.01105*lam)
a = aw + Achl*0.06*(Cc**0.602) + ay
  bs = 1.151302*Cs*lr*((400./lam)**0.7)
  bl = 0.3410739*Cl*((400./lam)**0.3)
b = 5.826E-3*(400./lam)**4.322 + bs+bl ! total scattering coeff.
fSsalb = b/(a+b)                       ! single-scattering albedo
return
end

subroutine getStats(ssalb, avth,avth2,avcos,avcos2)  ! calculates
real      ssalb, avth,avth2,avcos,avcos2             ! statistics
avth   = 0.15064/(ssalb**0.52447)          ! given by Eqs.(2)-(5)
avth2  = 0.11300/(ssalb**0.81265)
avcos  = 0.98572*(ssalb**0.12226)
avcos2 = 0.96011*(ssalb**0.079212)
return
end

real function McLeanoMTF(b,c,tauz,q,k,mu)  ! calculates McLean et
real*8    b,c, ssalb,tauz, k, q, mu, avth2 ! al. MTF from Ref.[1]
ssalb = b/c
avth2  = 0.11300/(ssalb**0.81265)
McLeanoMTF = EXP(-(0.25*ssalb*tauz*avth2)
&            *(q*q+tauz*k*(q*mu+tauz*k/3.0)))
return
end

real function FourJonMTF(b,c,taul,omeg)    ! calculates Fournier
real      b,c,taul, omeg, ssalb,taua,d,h,p,q  !  -Jonasz MTF [2]
ssalb = b/c
taua = 1.0 - ssalb
h = 0.11300*taul*taul*ssalb**0.18735
d = taua*h
```

```
p = 12.0 + d*(2.6 + 0.0375*d)
q = h*(7.0 + 0.191666667*d)/(taul*p)
p = (12.0/p)*(EXP(-taua)-EXP(-taul))*EXP(-q*omeg*omeg)
FourJonMTF = p + EXP(-taul)
return
end
```