# 1 VGFFMC

Program to use the F-field formalism to solve the scattering problem for a particle of arbitrary shape. The particle is divided into an array of dipoles on a cubic lattice (by program VGFIN). The scattering is computed through a plane wave expansion of the field inside the particle. From this the external field and phase function are calculated (in program VGFPHZ)

### 1.0.1 Units:

All equations are in Gaussian units. Lengths are all relative, that is, if you input a wavelength of 10 um then all other lengths must be in um. You may use m or cm or furlongs if you wish as long as all lengths are in the same units.

### 1.0.2 Time dependence:

The VGF code uses the time dependence $e^{-iwt)}$.

### 1.0.3 Geometry:

The particle is aligned with it's symmetry axis along the Z-axis. The incident plane wave may be rotated around the particle by the angles alpha, beta, and gamma (corresponding to the Euler angles as given in Arfkin, Mathematical Methods for Physicists, 3nd Ed., Academic Press, NY, 1970). All calculations are done in this 'body frame.' The polarization is given relative to the x-axis in the lab or global frame. In this frame the incident plane wave travels in the z-hat direction and the polarization is in the x-y plane. The direction of the electric field is measured by an angle psi from the x-axis.

### 1.0.4 History

Author: Original code by Lines *
Last Modification by Lines 21 APR 97 *
List of subroutines and their signatures *

### 1.0.5 Subroutines

subroutine kvectors(NTH,khatN,count) *
complex Function GAM(d,k,EPS) *
complex function CPSI(R,KhatN,k,mm,N,b) *
complex Function Wcalc(X,d,k,EPS)
this
subroutine Gcalc(R,k,Gmn,d,EPS,m,n) *
real function delta(alpha,beta) *
real function dd(alpha,i,beta,k) *
subroutine ROT(RRR, alpha, beta, gamma) *
subroutine MV(M,V,U) subroutine to multiply a 3x3 matrix by a vector with three components

### 1.1 Algorithm

We will assume a field with complex time dependence.

$$\mathbf{E}\left(\mathbf{r},t\right)=\mathbf{E}(\mathbf{r})\exp(-i\omega t)+c.c. \tag{1}$$

and with a Green's function solution

$$\mathbf{E}(\mathbf{r})(1+\frac{4\pi}{3}\chi)=\mathbf{E}^{e}(\mathbf{r})+\int d^{3}\mathbf{r}'\mathbf{G}\left(\mathbf{r},\mathbf{r}'\right)\cdot\chi(\mathbf{r}')\mathbf{E}(\mathbf{r}') \tag{2}$$

We define the term

$$W\left(\mathbf{r}\right)=\frac{\chi(\mathbf{r})}{(1+\frac{4\pi}{3}\chi(\mathbf{r}))} \tag{3}$$

then letting

$$\mathbf{F}(\mathbf{r})=\mathbf{E}(\mathbf{r})(1+\frac{4\pi}{3}\chi)$$

and the $F$-field

$$\mathbf{F}(\mathbf{r})=\mathbf{E}^{e}(\mathbf{r})+\int d^{3}\mathbf{r}'\mathbf{G}\left(\mathbf{r},\mathbf{r}'\right)\cdot W\left(\mathbf{r}'\right)\mathbf{F}(\mathbf{r}') \tag{4}$$

to make our calculations more compact.

We can write the F-field as

$$F_{\mu i}=E_{\mu i}^{in}+\sum_{\nu}G_{\mu i\nu j}W_{\nu}F_{\nu j} \tag{5}$$

where, for convenience, the cell dimension $d_{\mu}^{3}$ is included in the $W$ term.

$$W_{\nu}=Wd_{\nu}^{3} \tag{6}$$

Symbolically, this may be written as

$$\mathbf{F}=\mathbf{E}^{in}+\mathbf{GWF} \tag{7}$$

Proceeding directly to a numerical solution at this point would be equivalent to the DGF. Instead, we are going to try a variational solution. First, an error term $\boldsymbol{\varepsilon}$ is defined such that

$$\boldsymbol{\varepsilon}=\mathbf{F}-\mathbf{E}^{in}-\mathbf{GWF} \tag{8}$$

Writing components yields

$$\varepsilon_{\alpha i}=\sum_{\beta}\sum_{j}\left(1_{\alpha i\beta j}-G_{\alpha i\beta j}W_{\beta}\right)F_{\beta j}-E_{\alpha i}^{in} \tag{9}$$

Here the factor

$$F_{\alpha i}^{st}=-W_{\alpha}G_{\alpha i\alpha j}F_{\alpha j} \tag{10}$$

is the specially defined self term discussed below as the “self term.” Note that

$$1_{\alpha i\beta j} = \delta_{\alpha\beta}\delta_{ij} \tag{11}$$

Equation (62), gives a definition of the diagonal elements of the dyadic Green function taken from the self term,

$$F^{st}_{\alpha i} = \delta_{\alpha\beta}\left(\frac{4\pi}{3}\Gamma_\alpha\right)W_\alpha F_{\alpha i} \tag{12}$$

Equation (**??**) gives a discretized form of the dyadic Green function diagonal terms.

$$G_{\alpha i\alpha j} = \frac{4\pi}{3d^3_\alpha}\Gamma_\alpha\delta_{ij} \tag{13}$$

Now, for convenience, define the quantity

$$M_{\alpha i\beta j} = (1_{\alpha i\beta j} - W_\beta G_{\alpha i\beta j}) \tag{14}$$

where the $\alpha$, $i$, $\beta$,and $j$ are not summed over. Then

$$\varepsilon_{\alpha i} = M_{\alpha i\beta j}F_{\beta j} - E^o_{\alpha i} \tag{15}$$

where in equation (15) the summation convention has been employed throughout, on both Latin and Greek indices.

The function $\Phi$ to extremize is defined as the squared magnitude of the error $\boldsymbol{\varepsilon}$,

$$\Phi = \boldsymbol{\varepsilon}_{\alpha i}\boldsymbol{\varepsilon}^*_{\alpha i} = F_{\beta j}M^T_{\beta j\alpha i}M^*_{\alpha i\gamma l}F^*_{\gamma l} - E^{o*}_{\alpha i}M_{\alpha i\beta j}F_{\beta j} - E^o_{\alpha i}M^*_{\alpha i\beta j}F^*_{\beta j} + E^o_{\alpha i}E^{o*}_{\alpha i} \tag{16}$$

Again for convenience two additional quantities are defined which will shorten the notation

$$K_{\gamma l\beta j} = M^{*T}_{\gamma l\alpha i}M_{\alpha i\beta j} \tag{17}$$

$$X^*_{\beta j} = E^{o*}_{\alpha i}M_{\alpha i\beta j} \tag{18}$$

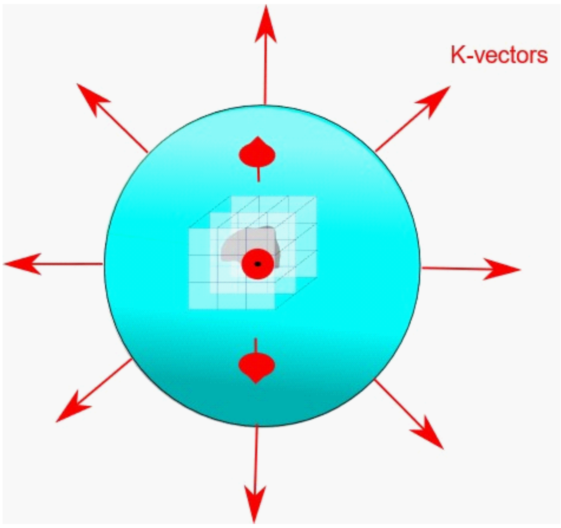
Then equation (16) can be written compactly as

$$\Phi = F^*_{\gamma l}K_{\gamma l\beta j}F_{\beta j} - \left[X^*_{\beta j}F_{\beta j} + X_{\beta j}F^*_{\beta j}\right] + E^o_{\alpha i}E^{o*}_{\alpha i} \tag{19}$$

Because the form of $\mathbf{F}$ is not known, a trial function $\tilde{\mathbf{F}}$ can be used which is an expansion in some basis function set $\Psi_{\beta N}$. We define the trial function $\tilde{\mathbf{F}}$ by

$$\tilde{F}_{\beta j} = \sum_{N=1}^{N_k} a_{Nj}\Psi_{\beta N} \tag{20}$$

where the sum has been written explicitly. The function $\tilde{\mathbf{F}}$ is substituted for $\mathbf{F}$. The functions $\Psi_{\beta N}$ we will assume to be of the form



$$\Psi_{\beta N} = e^{imk\hat{\mathbf{k}}_N \cdot \mathbf{r}_\beta} \tag{21}$$

which depend on the $\hat{\mathbf{k}}_N$

These are read in from a file. That file is created by vgfkv.f. And it looks like it makes a $k_N$ in equally spaced increments in the $\theta$ and $\phi$ directions. Then equation (16) is approximately given by

$$\Phi = \Psi^*_{\gamma M} a^*_{Ml} K_{\gamma l \beta j} a_{Nj} \Psi_{\beta N} - \left[X^*_{\beta j} \Psi_{\beta N} a_{Nj} + X_{\beta j} \Psi^*_{\beta N} a^*_{Nj}\right] + E^o_{\alpha i} E^{o*}_{\alpha i} \tag{22}$$

Now the notation can again be simplified by defining two functions

$$\begin{aligned} H_{MlNj} &= \Psi^*_{\gamma M} K_{\gamma l \beta j} \Psi_{\beta N} && (23) \\ Y^*_{Nj} &= \Psi^T_{N\beta} X^*_{\beta j} && (24) \end{aligned}$$

Then the final form of equation (16) is

$$\Phi = a^*_{Ml} H_{MlNj} a_{Nj} - \left[Y^*_{Nj} a_{Nj} + Y_{Nj} a^*_{Nj}\right] + E^o_{\alpha i} E^{o*}_{\alpha i} \tag{25}$$

To extremize $\Phi$, equation (25) is differentiated with respect to $a^*_{Li}$, yielding

$$\frac{\partial \Phi}{\partial a^*_{Li}} = H_{LiNj} a_{Nj} - Y_{Li} \tag{26}$$

The result is set equal to zero. Equation (26) can be written in matrix form as

$$(\mathbf{H})\,(\mathbf{a}) = (\mathbf{Y}) \tag{27}$$

The solution for $(\mathbf{a})$ is given by

$$(\mathbf{a}) = (\mathbf{H})^{-1}\,(\mathbf{Y}) \tag{28}$$

Writing out the $(\mathbf{H})$and $(\mathbf{Y})$ matrix explicitly in components we have

$$\begin{aligned} H_{MlNj} &= \Psi^*_{\gamma M}\left(1_{\alpha i\gamma l}{-}W^*_\beta G^*_{\alpha i\gamma l}\right)\left(1_{\alpha i\beta j}{-}W_\beta G_{\alpha i\beta j}\right)\Psi_{\beta N} \\ Y_{Nj} &= \left(1_{\alpha i\beta j}{-}W_\beta G_{\alpha i\beta j}\right)^* E^o_{\alpha i}\Psi^*_{\beta N} \end{aligned} \tag{29}$$

The functions $\Psi_{\beta N}$ could be any set of functions. A simple choice for larger particles with refractive indices that are not very different from unity is a set of plane waves:

$$\Psi_{\beta N} = e^{imk\hat{\mathbf{k}}_N\cdot\mathbf{r}_\beta} \tag{30}$$

where as before $m$ is the complex refractive index of the particle. These functions obey the correct wave equation inside the particle. The trial functions $\tilde{F}_{\beta j}$ are given by

$$\tilde{F}_{\beta j}{=}\sum_{N=1}^{N_k} a_{Nj}e^{imk\hat{\mathbf{k}}_N\cdot\mathbf{r}_\beta} \tag{31}$$

Again, the summation has been written explicitly. The $\hat{\mathbf{k}}_N$ or "$k$-vectors" must be chosen to represent the internal wave well. A "goodness of choice" criterion for the number and direction of the $\hat{\mathbf{k}}_N$ vectors results from evaluating $\Phi$ explicitly. Observing the values of the $a_{Nj}$ can determine which of the $k$-vectors are not contributing well. If the magnitude of an $a_N$ is small, then the $\hat{\mathbf{k}}_N$ is not well placed or may not be needed.

For the sake of computation, a temporary variable is employed in the calculations that follow.

$$T_{\alpha iNj} = \sum_\beta \left(1_{\alpha i\beta j}{-}G_{\alpha i\beta j}W_\beta\right)\Psi_{\beta N} \tag{32}$$

Using $T_{\alpha iNj}$, $H_{MlNj}$ and $Y_{Nj}$ can be expressed as

$$\begin{aligned} H_{MlNj} &= \sum_\alpha\sum_i T^*_{\alpha iMl}T_{\alpha iNj} \\ Y_{Nj} &= \sum_\alpha\sum_i T^*_{\alpha iNj}E^o_{\alpha i} \end{aligned} \tag{33}$$

This is the form of the quantities used in the actual algorithms.

Then since we found the $a_{Nj}$ with the matrix inversion we can calculate the

$$\tilde{F}_{\beta j}{=}\sum_{N=1}^{N_k} a_{Nj}e^{imk\hat{\mathbf{k}}_N\cdot\mathbf{r}_\beta} \tag{34}$$

and these are related to the electric field like this

$$\mathbf{F}(\mathbf{r}) = \mathbf{E}(\mathbf{r})(1+\frac{4\pi}{3}\chi)$$

so

$$\frac{\mathbf{F}(\mathbf{r})}{(1+\frac{4\pi}{3}\chi)}=\mathbf{E}(\mathbf{r})$$

and we have our electric field.

This is what the code is supposed to do.

## 1.2 Green's function and Self Term

Equation (5) can be written as a sum where the infinitesimal volume $d^3r'$ is now a finite volume $d_\nu^3$:

$$\begin{aligned} F_i(\mathbf{r}_\mu) &= E_i^{in}\left(\mathbf{r}_\mu\right)+\sum_{\nu\neq\mu} d_\nu^3 G_{ij}\left(\mathbf{r}_{\mu\nu}\right)WF_j(\mathbf{r}'_\nu)+I_i(\mathbf{r}_\mu) \\ &= E_{\mu,i}^{in}+W\sum_{\nu\neq\mu} d_\nu^3 G_{\mu i\nu j}F_{\nu j}+I_{\mu i} \end{aligned} \tag{35}$$

Here $I_{\mu i}$ is the self term, and Greek indices run and sum over dipole cell number and the Latin indices run and sum over vector components. Also, $\mathbf{r}_\mu$ is at the center of cell number $\mu$. Of course, this only works well if $d_\mu^3$ is small. Experience shows that $|m|\, kd<1$ should be used, where $m$ is the complex refractive index of the homogenous particle.

The Green's function is discretized as

$$G_{\mu i\nu j}=\exp(ikR_{\mu\nu})\left[\frac{k^2}{R_{\mu\nu}}\left(\delta_{ij}-\widehat{R_{\mu\nu i}}\widehat{R_{\mu\nu j}}\right)+\left(\frac{ik}{R_{\mu\nu}^2}-\frac{1}{R_{\mu\nu}^3}\right)\left(\delta_{ij}-3\widehat{R_{\mu\nu i}}\widehat{R_{\mu\nu j}}\right)\right] \tag{36}$$

where $\mathbf{R}_{\mu\nu}=\mathbf{r}_\mu-\mathbf{r}_\nu$. This equation is very similar to that used by Goedecke and O'Brien.[**?**]

But a problem occurs when the summation index $\nu$ is equal to $\mu$ : Terms with $R_{\mu\mu}$ in the denominator would diverge (blow up to infinity). Thus in equation (51) these terms have been separated. Because such a term results from the action of the cell field on itself, it is known as the self-term. The self-term results from letting $\mathbf{r}=\mathbf{r}_\mu$, and $\mathbf{r}'=\mathbf{r}_\mu+\boldsymbol{\xi}$, in equation (4) , and taking $\chi\left(\mathbf{r}'\right)\mathbf{F}_j\left(\mathbf{r}'\right)=\chi\left(\mathbf{r}_\mu\right)\mathbf{F}_j\left(\mathbf{r}_\mu\right)$. One gets

$$I_{\mu,i}=\left[\int_\square d^3\xi G_{ij}\left(\boldsymbol{\xi}\right)\right]WF_j(\mathbf{r}_\mu) \tag{37}$$

The $\square$ indicates that the integral is over the small dipole cell, of volume $d_\mu^3$. The integral in brackets in equation (53) can be written as

$$\int_\square d^3\xi G_{ij}\left(\boldsymbol{\xi}\right)=\int_\square d^3\xi e^{ik\xi}\left[\frac{k^2}{\xi}\left(\delta_{ij}-\hat{\xi}_i\hat{\xi}_j\right)+\left(\frac{ik}{\xi^2}-\frac{1}{\xi^3}\right)\left(\delta_{ij}-3\hat{\xi}_i\hat{\xi}_j\right)\right] \tag{38}$$

Employing the identity

$$\int_{\Box} d\Omega_{\xi}\hat{\xi}_i\hat{\xi}_j = \delta_{ij}\frac{1}{3}\int_{\Box} d\Omega_{\xi} \tag{39}$$

the last term vanishes due to symmetry of cubical cells. Also, the remaining terms can be combined to yield

$$\int_{\Box} d^3\xi G_{ij}\left(\boldsymbol{\xi}\right) = \delta_{ij}\frac{2k^2}{3}\int_{\Box} d^3\xi\frac{k^2 e^{ik\xi}}{\xi} \tag{40}$$

A series expansion of the integrand on the right hand side yields

$$\frac{\exp(ikx)}{x} = \left[\frac{1}{x} + ik - \frac{1}{2}k^2 x - \frac{1}{6}ik^3x^2 + \frac{1}{24}k^4x^3 + Ox^4\right] \tag{41}$$

which can be substituted into the previous expression to yield

$$\begin{aligned}\int_{\Box} d^3\xi G_{ij}\left(\boldsymbol{\xi}\right) &= \delta_{ij}\frac{2k^2}{3}\int_{\Box} d^3\xi\left(\frac{1}{\xi} + ik - \frac{1}{2}k^2\xi - \frac{1}{6}ik^3\xi^2 + \frac{1}{24}k^4\xi^3 + O\left(\xi^4\right)\right)\\ &\equiv \delta_{ij}\frac{4\pi}{3}\Gamma \end{aligned}\tag{42}$$

where this equation defines $\Gamma$ as

$$\Gamma = \left((ka)^2 + \frac{2}{3}i\left(ka\right)^3 - \frac{1}{4}\left(ka\right)^4 - \frac{1}{15}i\left(ka\right)^5 + O\left(ka\right)^6\right) \tag{43}$$

and the integral has been done by using an equivalent volume sphere with radius $a_\mu$

$$a_\mu = d_\mu\left(\frac{3}{4\pi}\right)^{\frac{1}{3}} \tag{44}$$

thus

$$\begin{aligned}\Gamma_\mu &= \left(\frac{3}{4\pi}\right)^{\frac{2}{3}}\left(kd_\mu\right)^2 + \left(\frac{i}{2\pi}\right)\left(kd_\mu\right)^3 - \frac{1}{4}\left(\frac{3}{4\pi}\right)^{\frac{4}{3}}\left(kd_\mu\right)^4\\ &\quad -\frac{1}{15}i\left(\frac{3}{4\pi}\right)^{\frac{5}{3}}\left(kd_\mu\right)^5 + O\left(\left(kd_\mu\right)^6\right)\end{aligned}\tag{45}$$

The expression for $I_{\mu i}$ can then be written as

$$I_{\mu i} = \delta_{ij}\frac{4\pi}{3}W\Gamma_\mu F_{\mu j} = \frac{4\pi}{3}W\Gamma_\mu F_{\mu i} \tag{46}$$

The first two terms in $\Gamma_\mu$ were used by Goedecke and O'Brien in their formulation of the DGF scattering code [**?**]. Draine and Goodman obtained a somewhat different expression for $\Gamma_\mu$ for placement of the dipole cells on a cubic lattice[**?**], i.e. the coefficients of the various powers of $kd_\mu$ are different than in equation(61). However, for ease in comparison with the DGF method, the form given by Goedecke and O'Brien is used here. The corrections of order $(kd_\mu)^4$and higher are insignificant; and even the term in $(kd_\mu)^2$ is not important if $kd_\mu \ll 1$. But, as shown by Goedecke and O'Brien, the lowest order imaginary term proportional to $(kd_\mu)^3$ is essential for agreement with the optical theorem, and this term is the same in the Draine and Goodman expression.

## 1.3 Initialized variables

The code initializes variables in lines 51-86. ForTran requires variables to be declared like in C++. It uses parenthesis to indicate an array.

We start with

| | |
|---|---|
| the maximum number of possible dipole cells | $NMAX$ |
| Number of $k$-vectors | $KNMAX$ |

| | |
|---|---|
| | $IPVT(3 * KNMAX)$ |
| | $z\,(3 * KNMAX)$ |
| | $rcond$ |
| | $NUSE$ |
| | $I$ |
| | $J$ |
| | $mcount$ |
| | $kcount$ |

| | |
|---|---|
| | $N$ |
| | $M$ |
| | $l$ |
| | $a$ |
| | $b$ |
| | $NK$ |
| | $np$ |
| | $mp$ |
| | $iseed$ |
| | $ikcount$ |
| | $itemp$ |

| | |
|---|---|
| | $N$ |
| | $M$ |
| | $l$ |
| | $a$ |
| | $b$ |
| | $NK$ |
| | $np$ |
| | $mp$ |
| | $iseed$ |
| | $ikcount$ |
| | $itemp$ |

| | |
|---|---|
| wavelength | $Wave$ |
| number of dipoles in along the major axis | $NSID$ |
| fine structure divisions per dipole cell side | $NLSID$ |
| Particle symmetry semi-axis | $RAD$ |
| Incident direction | $(\alpha lpha, \beta eta, gamma)$ |
| polarization angle | $psi$ |
| real and imaginary parts of index of refraction | $MR$, $MI$ |
| index of refraction (complex) | $M$ |
| $\pi$ | $PI$ |
| Aspect ratio, ration of major to minor axis | $AR$ |

| | |
|---|---|
| Dipole weighting factors | $D(NMAX)$ |
| Locations of dipoles, $x$, $y$, and $z$ components. | $R(NMAX, 3)$ |
| real and imaginary part of the permittivity | $ER$, $EI$ |
| permittivity of the particle material | $EPS$ |
| $r \cdot k$ | $RDK$ |
| | $TD$ |
| wave number $2\pi/\lambda$ | $k$ |
| | $PI$ |
| Convert from degrees to radians | $DEG$ |
| | $Khat\,(3)$ |
| Rotation matrix | $RRR\,(3, 3)$ |
| | $V\,(3)$ |
| I think this is the input electric field direction | $E0hat\,(3)$ |

| | |
|---|---|
| | $KhatN\,(KNMAX, 3)$ |
| | $dtemp$ |
| | $dsum$ |
| | $divd$ |
| | $kth\,(KNMAX)$ |
| | $kph\,(KNMAX)$ |
| | $ERR$ |
| | $ERRlast$ |
| | $ERR0$ |

| | |
|---|---|
| | $aa\,(3 * KNMAX, 3 * KNMAX)$ |
| | $bb\,(3 * KNMAX)$ |
| complex index of refraction (21) | $mm$ |
| complex permittivity (20) | $EPS$ |
| complex susceptibility (20) | $X$ |
| W-factor (39) $\chi\,(r)\,/(1 + \frac{4}{3}\pi\chi\,(r))$ | $W$ |
| | $C$ |
| | $CI$ |
| | $temp$ |

| | |
|---|---|
| Incident field (line 138-148) | $E0\,(3*KNMAX)$ |
| | $E\,(3*KNMAX)$ |
| | $T1\,(NMAX, 3*KNMAX, 3)$ |
| | $F\,(NMAX, 3)$ |
| | $H(KNMAX, 3, KNMAX, 3)$ |
| | $Y(KNMAX, 3)$ |
| | $An(KNMAX, 3)$ |
| | $PHI$ |

$PI = 3.141592654$
$DEG = PI/180.0$
$CI = (0.0, 1.0)$
$iseed = 234564$

## 2 Main Program

The program defines many variables at the start. After defining $\pi$ and $i$ in line 86, it defines the particle permittivity $\epsilon$ in line 107, the complex index of refraction $mm$ in line 108, and the complex susceptibility $\chi$ in line 109.

It inputs the positions and weights for our dipoles in lines 110 to 116.

It then find the wave number, $k$ in line 119 and it rotates the direction of the incident electric field into the position we asked for in vgfin in lines 125-135.

It then calculates the $W$ factor (137)

$$W\left(\nu\right) = \frac{\chi}{1 + \frac{4\pi}{3}\chi}$$

and the incident field (147)

$$E_o\left(r\right) = e^{-ik\mathbf{r}\cdot\hat{\mathbf{k}}}$$

for every $r$ where we have placed a dipole.

The program then calls a subroutine to bring in our direction angles for each of the $k-vectors$. (152)

The program then calls kvector3 to turn the $k$-vector direction angles into the components of the $\hat{k}$ vectors.

The program then starts the monte Carlo loop. In this loop it calculates where

$$\Psi_{\beta N} = e^{imk\hat{\mathbf{k}}_N\cdot\mathbf{r}_\beta} \tag{47}$$

$$\psi = e^{(i*k*khatN.R(b))}$$

equation 183

We want to find the $a_{Nj}$ in

$$\tilde{F}_{\beta j}=\sum_{N=1}^{N_k} a_{Nj}\Psi_{\beta N} \tag{48}$$

and use this to find our $E$ field

$$\frac{\mathbf{F}(\mathbf{r})}{(1+\frac{4\pi}{3}\chi)}=\mathbf{E}(\mathbf{r})$$

The code calculates

$$T_{\alpha iNj}=\sum_{\beta}\left(1_{\alpha i\beta j}-G_{\alpha i\beta j}W_{\beta}\right)\Psi_{\beta N}$$

$$T1_{(a,i,N,j)}=T1_{(a,i,N,j)}+(dd_{(a,i,b,j)}-d_{(b)}^{3}*W*GG)\psi$$

$$T1_{(a,i,N,j)}=T1_{(a,i,N,j)}+(dd_{(a,i,b,j)}-d_{(b)}^{3}*W*GG_{(R,k,dtemp,EPS,a,i,b,j)})e^{(i*k*khatN.R(b))}$$

which is equation 185 and happens between lines (166) and (181).

@@@I don't know what $d\,(b)$ is yet.

Now it calculates Y (191)

$$Y_{Nj}=\sum_{\alpha}\sum_{i}T_{\alpha iNj}^{*}E_{\alpha i}^{o}$$

and then (199)

$$H_{MlNj}=\sum_{\alpha}\sum_{i}T_{\alpha iMl}^{*}T_{\alpha iNj}$$

Then it does a matrix solve. It first must reshape our multidimensional matrices into two-dimensional matrices because that is what the inversion routine can do.

So $Y_{Nj}$ is turned into $bb\,(m)$ and $H_{MlNj}$ is turned into $aa(m,mp)$

The matrix inversion routine comes in two parts. It first calls CGECO and then CGESL. The call to cgeco Factors a COMPLEX matrix by Gaussian elimination and estimates the condition of the matrix. The call to cgefa solves the COMPLEX system A*X=B or CTRANS(A)*X=B using the factors computed by CGECO or CGEFA. The input matrix is $aa$ $(H)$ and the subroutine uses two variables ipvt and z that come in as zeros. the variable $aa$ leaves cgeco row reduced and enters cgesl to finish the inversion and multiply. The input matrix is still aa, and the vector is now called $bb$. After cgesl, the variable $bb$ has the solution. What we did was

$$(\mathbf{a})=(\mathbf{H})^{-1}(\mathbf{Y}) \tag{49}$$

The result should be the $a_{Nj}$ values which we then use to calculate

$$\tilde{F}_{\beta j}= \sum_{N=1}^{N_k} a_{Nj}\Psi_{\beta N} \tag{50}$$

but our *bb* vector is in the wrong form. We have to turn it back into an array of vectors. And that is what happens next. The *bb* output is turned into $A_n$. Next we want to calculate the $F$ fields and then the $E$ fields.

$$\mathbf{E}(\mathbf{r}) = \frac{\mathbf{F}(\mathbf{r})}{(1+\frac{4\pi}{3}\chi)}$$

That happens in a call to Ecalc. That first finds $F$ and then finds $E$.

There is still the test to see if our field has converged. That is all the calculation of ERR from $\Phi$ in lines 243 and 245. If the error is small enough then the $F$ and $E$ are calculated. $If$ not, we move the k-vectors around (line 271) to try to find a better set that represent the field better. Then we try the calculation again.

In the end, the fields are output by a function called printout and then the code vgfphz will make fancy graphs for us.

# 3 Subroutines

Subroutines are just functions. In vgffmc there are the following subroutines.

subroutine Gcalc(R,k,Gmn,d,EPS,m,n) * This is a lie, it is called GG (see below)

subroutine Ecalc(NUSE,R,E,F,An,khatN,X,NK,K,mm)

subroutine printout(COMMENTS,ERR,ERR0,NUSE,wave,alpha,beta,gamma,

& psi, RAD, EPS,R,D,E,NK,An,mcount,OUTFILE,khatN,divd)

complex function CPHI(An,H,Y,E0,NK,NUSE)

subroutine printkv(kth,kph,NK,KFILE,ERR,ERRlast,mcount,kcount)

subroutine getkv(kth,kph,NK,KFILE,ERR,ERRlast,mcount,kcount)

subroutine move1kv(mcount,khatN,kth,kph,NK,iseed,divd)

real function rannum(iseed)

subroutine kvectors4(N,khatN,kth,kph,NK)

subroutine kvectors3(khatN,kth,kph,NK)

complex Function GAM(d,k,EPS)

complex function CPSI(R,KhatN,k,mm,N,b)

complex Function Wcalc(X)

complex function GG(R,k,d,EPS,a,i,b,j)

real function delta(alpha,beta)

real function dd(alpha,i,beta, j)

subroutine ROT(RRR, alpha, beta, gamma)

subroutine MV(M,V,U) subroutine to multiply a 3x3 matrix by a vector with three components

What follows is a description of each function.

### 3.1 subroutine Ecalc(NUSE,R,E,F,An,khatN,X,NK,K,mm)

Not done yet

### 3.2 subroutine printout(COMMENTS,ERR,ERR0,NUSE,wave,alpha,beta,gamm

### 3.3 & psi, RAD, EPS,R,D,E,NK,An,mcount,OUTFILE,khatN,divd)

Not done yet

### 3.4 Subroutine getkvectors(kth,kph,NK,KFILE,ERR,ERRlast,mcount,kcount)

subroutine to read in the kvectkors from the file KFILE
subroutine kvectors(NTH,khatN,count) *
complex Function GAM(d,k,EPS) *
complex function CPSI(R,KhatN,k,mm,N,b) *

### 3.5 subroutine move1kv(mcount,khatN,kth,kph,NK,iseed,divd)

Not done yet

### 3.6 real function rannum(iseed)

Not done yet

### 3.7 subroutine kvectors4(N,khatN,kth,kph,NK)

Not done yet

### 3.8 subroutine kvectors3(khatN,kth,kph,NK)

Not done yet

### 3.9 complex function CPSI(R,KhatN,k,mm,N,b)

Not done yet

### 3.10 complex Function Wcalc(X,d,k,EPS)

this function calculates the W-factor from the dissertation equation 39 and 141.

$$W(\nu) = \frac{\chi}{1 + \frac{4\pi}{3}\chi}$$

where $\chi$ is the complex susceptibility.

## 3.11 subroutine GG and GAM

We already did this math. It is repeated here. But what you need to know now is that the Green's function happens in Gcalc.

Now the math again.

Equation (4) can be written as a sum where the infinitesimal volume $d^3r'$ is now a finite volume $d_\nu^3$:

$$\begin{aligned} F_i(\mathbf{r}_\mu) &= E_i^{in}(\mathbf{r}_\mu) + \sum_{\nu\neq\mu} d_\nu^3 G_{ij}(\mathbf{r}_{\mu\nu}) W F_j(\mathbf{r}'_\nu) + I_i(\mathbf{r}_\mu) \qquad (51) \\ &= E_{\mu,i}^{in} + W \sum_{\nu\neq\mu} d_\nu^3 G_{\mu i\nu j} F_{\nu j} + I_{\mu i} \end{aligned}$$

Here $I_{\mu i}$ is the self term, and Greek indices run and sum over dipole cell number and the Latin indices run and sum over vector components. Also, $\mathbf{r}_\mu$ is at the center of cell number $\mu$. Equation (51) only approximates equation (4) well if $d_\mu^3$ is small. Experience shows that $|m|\,kd < 1$ should be used, where $m$ is the complex refractive index of the homogenous particle.

The Green's function is discretized as

$$G_{\mu i\nu j} = \exp(ikR_{\mu\nu}) \left[ \frac{k^2}{R_{\mu\nu}} \left( \delta_{ij} - \widehat{R_{\mu\nu i}} \widehat{R_{\mu\nu j}} \right) + \left( \frac{ik}{R_{\mu\nu}^2} - \frac{1}{R_{\mu\nu}^3} \right) \left( \delta_{ij} - 3\widehat{R_{\mu\nu i}} \widehat{R_{\mu\nu j}} \right) \right] \qquad (52)$$

where $\mathbf{R}_{\mu\nu} = \mathbf{r}_\mu - \mathbf{r}_\nu$. This equation is very similar to that used by Goedecke and O'Brien.[**?**]

### 3.11.1 Self-Term

A problem occurs when the summation index $\nu$ is equal to $\mu$ : Terms with $R_{\mu\mu}$ in the denominator would diverge. Thus in equation (51) these terms have been separated. Because such a term results from the action of the cell field on itself, it is known as the self-term. The self-term results from letting $\mathbf{r} = \mathbf{r}_\mu$, and $\mathbf{r}' = \mathbf{r}_\mu + \boldsymbol{\xi}$, in equation (4) , and taking $\chi(\mathbf{r}')\mathbf{F}_j(\mathbf{r}') = \chi(\mathbf{r}_\mu)\mathbf{F}_j(\mathbf{r}_\mu)$ . One gets

$$I_{\mu,i} = \left[ \int_\square d^3\xi G_{ij}(\boldsymbol{\xi}) \right] W F_j(\mathbf{r}_\mu) \qquad (53)$$

The $\square$ indicates that the integral is over the small dipole cell, of volume $d_\mu^3$. The integral in brackets in equation (53) can be written as

$$\int_\square d^3\xi G_{ij}(\boldsymbol{\xi}) = \int_\square d^3\xi e^{ik\xi} \left[ \frac{k^2}{\xi} \left( \delta_{ij} - \hat{\xi}_i \hat{\xi}_j \right) + \left( \frac{ik}{\xi^2} - \frac{1}{\xi^3} \right) \left( \delta_{ij} - 3\hat{\xi}_i \hat{\xi}_j \right) \right] \qquad (54)$$

Employing the identity

$$\int_\square d\Omega_\xi \hat{\xi}_i \hat{\xi}_j = \delta_{ij} \frac{1}{3} \int_\square d\Omega_\xi \qquad (55)$$

the last term vanishes due to symmetry of cubical cells. Also, the remaining terms can be combined to yield

$$\int_{\square} d^3\xi G_{ij}\left(\boldsymbol{\xi}\right) = \delta_{ij}\frac{2k^2}{3}\int_{\square} d^3\xi \frac{k^2 e^{ik\xi}}{\xi} \tag{56}$$

A series expansion of the integrand on the right hand side yields

$$\frac{\exp(ikx)}{x} = \left[\frac{1}{x} + ik - \frac{1}{2}k^2x - \frac{1}{6}ik^3x^2 + \frac{1}{24}k^4x^3 + Ox^4\right] \tag{57}$$

which can be substituted into the previous expression to yield

$$\begin{aligned} \int_{\square} d^3\xi G_{ij}\left(\boldsymbol{\xi}\right) &= \delta_{ij}\frac{2k^2}{3}\int_{\square} d^3\xi\left(\frac{1}{\xi} + ik - \frac{1}{2}k^2\xi - \frac{1}{6}ik^3\xi^2 + \frac{1}{24}k^4\xi^3 + O\left(\xi^4\right)\right) \\ &\equiv \delta_{ij}\frac{4\pi}{3}\Gamma \end{aligned} \tag{58}$$

where this equation defines $\Gamma$ as

$$\Gamma = \left(\left(ka\right)^2 + \frac{2}{3}i\left(ka\right)^3 - \frac{1}{4}\left(ka\right)^4 - \frac{1}{15}i\left(ka\right)^5 + O\left(ka\right)^6\right) \tag{59}$$

and the integral has been done by using an equivalent volume sphere with radius $a_\mu$

$$a_\mu = d_\mu\left(\frac{3}{4\pi}\right)^{\frac{1}{3}} \tag{60}$$

thus

$$\begin{aligned} \Gamma_\mu &= \left(\frac{3}{4\pi}\right)^{\frac{2}{3}}\left(kd_\mu\right)^2 + \left(\frac{i}{2\pi}\right)\left(kd_\mu\right)^3 - \frac{1}{4}\left(\frac{3}{4\pi}\right)^{\frac{4}{3}}\left(kd_\mu\right)^4 \\ &\quad -\frac{1}{15}i\left(\frac{3}{4\pi}\right)^{\frac{5}{3}}\left(kd_\mu\right)^5 + O\left(\left(kd_\mu\right)^6\right) \end{aligned} \tag{61}$$

The expression for $I_{\mu i}$ can then be written as

$$I_{\mu i} = \delta_{ij}\frac{4\pi}{3}W\Gamma_\mu F_{\mu j} = \frac{4\pi}{3}W\Gamma_\mu F_{\mu i} \tag{62}$$

The first two terms in $\Gamma_\mu$ were used by Goedecke and O'Brien in their formulation of the DGF scattering code [**?**]. Draine and Goodman obtained a somewhat different expression for $\Gamma_\mu$ for placement of the dipole cells on a cubic lattice[**?**], i.e. the coefficients of the various powers of $kd_\mu$ are different than in equation(61). However, for ease in comparison with the DGF method, the form given by Goedecke and O'Brien is used here. The corrections of order $\left(kd_\mu\right)^4$and higher are insignificant; and even the term in $\left(kd_\mu\right)^2$ is not important if $kd_\mu \ll 1$. But, as shown by Goedecke and O'Brien, the lowest order imaginary term proportional to $\left(kd_\mu\right)^3$ is essential for agreement with the optical theorem, and this term is the same in the Draine and Goodman expression.

The code that does this is the subroutine GG

```
C***********************************************************************
C*---------------------------------------------------------------------*
complex function GG(R,k,d,EPS,a,i,b,j)
C*---------------------------------------------------------------------*
C***********************************************************************
C*    Function to calculate the dyadic Green’s function for a dipole   *
C*      in the IBM write-up (equation ???)                             *
C*    single value checked 1 Aug 96  Formula Checked 6 Aug 96          *
C***********************************************************************
C *** Set the value of NMAX via an included file                     ***
implicit none
include ’nmax.inc’
C****  Variables                                                    ****
complex PHZ,t1,t2,temp,CI,GAM,EPS
real RMAG,Rhat(3),R(NMAX,3),Rab(3),k,K2,d,PI
real delta
integer a,b,i,j
Parameter(PI=3.141592654,CI=(0.0,1.0))
C
K2=k*k
C       d3=d**3
if(b.ne.a) then
C         calculate separation distance Rmn=Rn-Rm and RMAG=|Rmn|
 Rab(1)=R(a,1)-R(b,1)
 Rab(2)=R(a,2)-R(b,2)
 Rab(3)=R(a,3)-R(b,3)
 RMAG=Rab(1)**2+Rab(2)**2+Rab(3)**2
 RMAG=RMAG**0.5
C         Make a unit vector in the Rmn direction                    ***
 Rhat(1)=Rab(1)/RMAG
 Rhat(2)=Rab(2)/RMAG
 Rhat(3)=Rab(3)/RMAG
C
 temp=CI*k*RMAG
 PHZ=cexp(temp)
C
 t1=(K2/RMAG)*(delta(i,j)-Rhat(i)*Rhat(j))
 t2=(ci*k/RMAG**2-1.0/RMAG**3)*(delta(i,j)-3.*Rhat(i)*Rhat(j))
 GG=PHZ*(t1+t2)
else
 GG=4.*PI*GAM(d,k,EPS)/(3.0*d**3)
end if
return
end
```

which calls the subroutine GAM based on equation (61)

```
C***********************************************************************
C*---------------------------------------------------------------------*
complex Function GAM(d,k,EPS)
C*---------------------------------------------------------------------*
C***********************************************************************
C*    Function to calculate the self term contribution termed GAMMA    *
C*      in the IBM write-up. The form for GAM  is taken from the work  *
C*      of B. T. Draine and J. Goodman, Astrophysical Journal, 405:    *
C*      685-697, 1993 March 10. Two other self term calculations are   *
C*      listed here for reference.  In my experience, the Draine and   *
C*      Goodman formulation is the better of the three.                *
C* Goedecke and O'Brian: Note the sign change due the time dependance  *
C*      in IBM being exp(-iwt). This differes from Goedecke and        *
C*      O'Brien's choice.                                              *
C*       GAM=(3./(4.*PI))**(2./3.)*(kd)**2 + CI*kd**3/(2.*PI)          *
C* All The terms in the Goedecke and O'Brien series.  Goedecke and     *
C*      O'Brien expand the exponential in the self term integral and   *
C*      throw away most of the trems.  This is the result if you keep  *
C*      all the terms.                                                 *
C*       real a                                                        *
C*       complex temp                                                  *
C*       a=d*(3./(4.*PI))**(1./3.)                                     *
C*       temp=CI*k*a                                                   *
C*       GAM=2.*((1.-CI*k*a)*cexp(temp)-1.)                            *
C*  Single value checed 1 Aug 96   formula checked 6 Aug 96            *
C***********************************************************************
C**** Variables                                                     ****
implicit none
complex EPS,CI
real k,d,kd
real PI
parameter (PI=3.141592654,CI=(0.0,1.0))
c       real b1,b2,b3,S
real b1
kd=k*d
C *** Drain and Goodman                                              ***
c       b1=-1.8915316
c       b2=0.1648469
c       b3=-1.7700004
c       S=1./5.
c       GAM=(3./(4.*PI))*((b1+EPS*(b2+b3*S))*kd**2+(2.*CI*(kd**3)/3.))
c      b1=0.0
b1=(3./(4.*PI))**(2./3.)
GAM=b1*(kd)**2 + CI*kd**3/(2.*PI)
return
end
```

### 3.12 real function delta(alpha,beta)

not done yet

### 3.13 real function dd(alpha,i,beta, j)

Not done yet

### 3.14 subroutine ROT(RRR, alpha, beta, gamma)

Not done yet

### 3.15 subroutine MV(M,V,U) subroutine to multiply a 3x3 matrix by a vector with three components

Not done yet

## 4 In Order in the Code

This next part is probably not useful. Maybe Ignore it.

51-78 Declare variables

80 Set inputfile (from vgfin.f), the output file name, and kfile (from vgfkv.f)

82-83 Declare function types. We will use several functions in this code that are specific to the code. They are declared here.

87-88 Define some constants

89-98 Interact with the user to get input data

99-116 Open the input file (from vgfin.f) and read in NUSE, Wave, Alpha, Beta, Gamma, Psi, RAD, Er, EI, TD and the cell positions. Along the way calculate $\chi$, $\varepsilon$, and $m$.

117-135 Make the incoming plane wave and then rotate it into the right position. This starts with angles $\alpha$, $\beta$, and $\gamma$ and converts from degrees to radians. The the function Rot is used to calculate a rotation matrix. This function calculates the Euler rotation matrix where the three rotation angles are alpha, a rotation about the z-axis, beta, a rotation about the new y-axis, and gamma, a rotation about the new z-axis.

$$RRR = \begin{pmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & \cos\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\beta\cos\gamma \\ -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\beta\sin\gamma \\ \cos\alpha\sin\beta & \sin\alpha\sin\beta & \cos\beta \end{pmatrix}$$

we will start with our input direction being in the $z$-direction

$$V_k = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

so our new direction for our incoming wave in the particle frame is

$$Khat = RRR * V$$

where the function MV is used to do the multiply. But then we want the $E$-vector to be in the $x - y$ plane in the lab frame (can't remember why). So take

$$V_E = \begin{pmatrix} \cos\psi \\ \sin\psi \\ 0 \end{pmatrix}$$

then rotate into our particle frame

$$E_{Ohat} = RRR * V$$

and use MV again to do the multiply.

137 Calculate the $W$ factor using function Wcalc

$$W(\nu) = \frac{\chi}{1 + \frac{4\pi}{3}\chi}$$

138-148 Form the incoming field. This will be of the form

$$\mathbf{E}(\mathbf{r}, t) = E_o \exp(i\mathbf{r} \cdot \mathbf{k}) \exp(-i\omega t)$$

And we want to know the value of the field at every cell. The code makes the $\mathbf{r} \cdot \mathbf{k}$ in a loop from 140-142. It has to do this for each cell. So there is an outside loop from 138 to 148 that loops over cells and the $r$ components will be different for every cell as we calculated in vgfin.f. The $\mathbf{r} \cdot \mathbf{k}$ is then multiplied by $i$ in line 143 and then put into an exponent in line 144 and finally the efield components are assembled in a small loop in lines 145 to 147.

151-154 Now go and get the k-vectors that we built in vgfkv.f. These will be used to form the outgoing scattered field.

156-242 is the Monty Carlo loop that finds the outgoing scattered field. It calculates the $a_{Nj}$ so later we can form

$$\tilde{F}_{\beta j} = \sum_{N=1}^{N_k} a_{Nj} \Psi_{\beta N}$$

and then

$$\mathbf{E}(\mathbf{r}) = \frac{\mathbf{F}(\mathbf{r})}{(1 + \frac{4\pi}{3}\chi)}$$

which is what we want as our answer.

The loop starts with finding $\Psi_{\beta N}$

$$\Psi_{\beta N} = e^{imk\hat{\mathbf{k}}_N \cdot \mathbf{r}_\beta}$$

$$\psi = e^{(i*k*khatN.R(b))}$$

and this is done with a call to the function CPHI.

Then it starts to calculate

$$ERR = \left|\Psi_{\beta N}\right|^2$$

This will be our optimization test. In line 248 we check to see if $ERR > ERR0$ The $ERR0$ was something we asked the user to input at the start. If $ERR < ERR0$ then we are done with one iteration. The original inputs worked. So the code jumps down to the else clause in line 275 and it calls Ecalc to calculate the scattered efield