PROGRAM NOTE

Antisymmetric Distorted Wave Impulse Approximation Calculations for Composite Particle Scattering

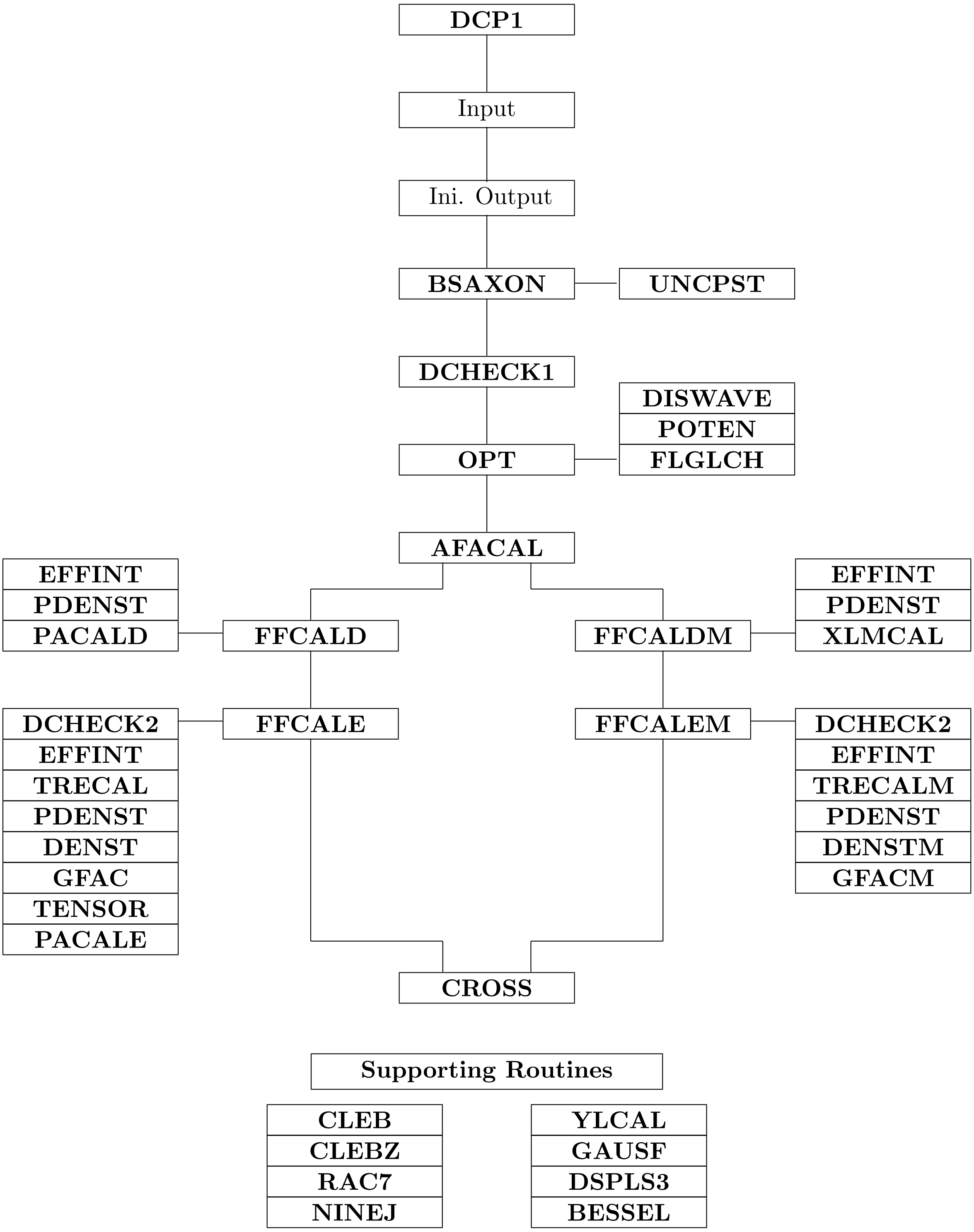
by

T. Udagawa, A. Schulte and F. Osterfeld

Nuclear Physics, **A474** (1987) 131-154

Contents

I. Structure of the Program



35 SUBROUTINES, 8300 LINES

II. Input Data

Line 1 : READ(7,10) (KTRLD(N),N=1,9)
FORMAT(24I3)
***Stored in COMMON /CNTRL/

IF KTRLD(1) = 0; Normal composite particle case with exact finite range form factor
1; Recoil effects in the plane wave approximation
2; No-recoil calculation is made
3; Nucleon-nucleon case (Not necessarily input)
IF KTRLD(2) = 1; PWIA calculation is made
IF KTRLD(3) = 1; Double folding potential is used
2; Single folding potential is used
IF KTRLD(4) = 1; Tensor force is neglected
Tensor force is set equal to zero in EFFINT
IF KTRLD(5) = 1; Exchange effect is neglected
IF KTRLD(6) = 1; Spin transfer coefficients are calculated.
IF KTRLD(9) = 1; Relativistic kinematics is used

Line 2 : READ(7,10) (KTLOUT(N),N=1,24)
FORMAT(24I3)
***Stored in COMMON /CNTRL/

KTLOUT(N) = 1; Output of bound state wave functions
2; Output of DW
3; Output of form factors
4; Output in GFAC
5; Output of elastic scattering in OPT
7; Output of Coulomb wave in FLGLCH
8; Output of ph form factors
13; Output of detailed form factors. 10 is suggested.

Line 3 : READ(7,10) JATW,ISATW
FORMAT(24I3)
***Stored in COMMON /FFCC/

JATW = Twice of the total angular momentum of the target A, I_A.
ISATW = Twice of the spin of the projectile a, s_a.

Line 4 : READ(7,12) WREXP
FORMAT(10F7.3)

WREXP = Dummy

Line 5 : READ(7,12) TMI,Z1I,PMI,Z2I,ELI
FORMAT(10F7.3)
***Stored in COMMON /DWCC/ by changing the names as
TMASA,TZA,PMASA,PZA

TMI =	Mass number of initial target, m_A
Z1I =	Charge number of target, Z_A
PMI =	Mass number of projectile, m_a (Presently up to 3. Otherwise, increase parameter "NXA")
Z2I =	Charge number of projectile, Z_a
ELI =	Incident beam LAB energy in MeV

Line 6 : READ(7,12) VA,WA,WAS,ARA,AIA,AISA,RZRA,RZIA,RZISA,RZCA
FORMAT(10F7.3)
***Stored in COMMON /POTCC/

	Optical Model Parameters (OMP) in the incident channel
VA =	Depth parameter of real potential in MeV
WA =	Depth parameter of imaginary potential
WAS =	Depth parameter of imaginary surface potential
ARA =	Diffuseness parameter of real potential in fm
AIA =	Diffuseness parameter of imaginary potential
AISA=	Diffuseness parameter of imaginary surface potential
RZRA =	Reduced radius parameter of real potential in fm
RZIA =	Reduced radius parameter of imaginary potential
RZIAS =	Reduced radius parameter of imaginary surface potential
RZCA =	Reduced radius parameter of Coulomb potential

Line 7 : READ(7,12) VX,ARX,RZRX ! TEMP
FORMAT(10F7.3)
***Stored in COMMON /POTCC/

	Optical Model Parameters (OMP) in the incident channel
VX =	Depth parameter of extra real potential
ARX =	Diffuseness parameter of extra real potential
RZRX =	Reduced radius parameter of extra real potential

Line 8 : READ(7,12) VB,WB,WBS,ARB,AIB,AISB,RZRB,RZIB,RZISB,RZCB
FORMAT(10F7.3)
***Stored in COMMON /POTCC/

Optical Model Parameters (OMP) in the exit channel
VB = Depth parameter of real potential
WB = Depth parameter of imaginary potential
WBS = Depth parameter of imaginary surface potential
ARB = Diffuseness parameter of real potential
AIB = Diffuseness parameter of imaginary potential
AISB= Diffuseness parameter of imaginary surface potential
RZRB = Reduced radius parameter of real potential
RZIB = Reduced radius parameter of imaginary potential
RZIBS = Reduced radius parameter of imaginary surface potential
RZCB = Reduced radius parameter of Coulomb potential

Line 9 : READ(7,12) VY,ARY,RZRY ! TEMP
FORMAT(10F7.3)
***Stored in COMMON /POTCC/

Optical Model Parameters (OMP) in the exit channel
VY = Depth parameter of extra real potential
ARY = Diffuseness parameter of extra real potential
RZRY = Reduced radius parameter of extra real potential

Line 10 : READ(7,12) TMF,Z1F,PMF,Z2F
FORMAT(10F7.3)
***Stored in COMMON /DWCC/ by changing the names as
TMASB,TZB,PMASB,PZB

TMF = Mass number of residual nucleus, m_B
Z1F = Charge number of residual nucleus, Z_B
PMF = Mass number of ejectile, m_b
Z2F = Charge number of ejectile, Z_b

Line 11 : READ(7,10) (LDWMIR(I),LDWMXR(I),LDWSTR(I),I=1,2)
FORMAT(24I3)
***Stored in COMMON /DWCC/

Number of partial waves, ℓ_a, ℓ_b
LDWMIR(1) = Starting partial wave in the incident channel, ℓ_a
LDWMXR(1) = Ending partial wave
LDWSTR(1) = Step of partial wave
LDWMIR(2) = Starting partial wave in the exit channel, ℓ_b
LDWMXR(2) = Ending partial wave
LDWSTR(2) = Step of partial wave

Line 12 : READ(7,11) (NXMIR(I),NXMXR(I),I=1,4),NONE,NHDMX,NHEMX,KTRLHE
FORMAT(14I5)
***Stored in COMMON /DWCC/ for NXMIR(I),NXMXR(I)
***Stored in COMMON /FFCC/ for NHDMX,NHEMX

NXMIR(1) = Starting mesh point in the incident channel, r_a
NXMXR(1) = Ending mesh point
NXMIR(2) = Starting mesh point in the exit channel, r_b
NXMXR(2) = Ending mesh point
NXMIR(3) = Starting mesh point in the projectile system, r_2
NXMXR(3) = Ending mesh point
NXMIR(4) = Starting mesh point in the target system, r_1
NXMXR(4) = Ending mesh point
NONE = Dummy
NHDMX = Interaction range in the direct form factor calculation, r^D
NHEMX = Interaction range in the exchange form factor calculation, r^E
KTRLHE = 0; Equal spaced mesh points in the exchange form factor
1; Cubical power spaced mesh points (See DCP1, Not used)

Line 13 : READ(7,12) (XMESR(I),I=1,4),XNONE,XMESHD,XMESHE
FORMAT(10F7.3)
***Stored in COMMON /DWCC/ for XMESR(I)
***Stored in COMMON /FFCC/ for XMESHD,XMESHE

XMESR(1) = Mesh size in the incident channel, Δr_a
XMESR(2) = Mesh size in the exit channel, Δr_b
XMESR(3) = Mesh size in the projectile system, Δr_2
XMESR(4) = Mesh size in the target system, Δr_1
XNONE = Dummy
XMESHD = mesh size in the direct form factor calculation, Δr^D
XMESHE = mesh size in the exchange form factor calculation, Δr^E

Line 14 : READ(7,11) NTHEB,MXMAX,MAMAX,
LAMMXD(1),LAMMXD(2),LAMMXD(3)
FORMAT(14I5)
***Stored in COMMON /ANGLC/ for NTHEB
***Stored in COMMON /FFCC/ for MXMAX,MAMAX
***Stored in COMMON /CDENS/ for LAMMXD(I)

NTHEB = Number of angles in the differential cross section calculations
MXMAX = Maximum value of m_{ℓ_t}
MAMAX = Dummy
LAMMXD(1) = Maximum value of λ_1 in exchange form factor
LAMMXD(2) = Maximum value of λ_2 in exchange form factor
LAMMXD(3) = Maximum value of λ in exchange form factor

Line 15 : READ(7,11) NBCMI,NBCMX,NBSTPD,NBSTPE,NGAUSR,
NONAR,NASTEP,N1STEP
FORMAT(14I5)
***Stored in COMMON /FFCC/

NBCMI =	Minimum mesh point in the form factor
NBCMX =	Maximum mesh point
NBSTPD =	Mesh step in the direct f. f.
NBSTPE =	Mesh step in the exchange f. f.
NGAUSR =	Number of gaussian integration
NONAR =	Number of mesh point in r_a integration for exchange ff. (Integration is made only around $r_a = r_b$ with a range NONAR.)
NASTEP =	Mesh step in the incident channel radius, r_a
N1STEP =	Mesh step in the target system radius, r_1

***For (p,n) or (p,p'), better put all steps unity.

Line 16 : READ(7,12) THEB,DTHEB,THMIN,THMAX,THIND
FORMAT(10F7.3)
***Stored in COMMON /ANGLC/ for THEB,DTHEB
***Stored in COMMON /OPTL/ for THMIN,THMAX,THIND

THEB =	Starting angle in the differential cross section calculations
DTHEB =	Steps in angle
THMIN =	Starting angle in elastic cross section calculations
THMAX =	Ending angle
THIND =	steps in angle

Line 17 : READ(7,10) JT,KPARIT,IST,NLSMAX,NOLTR,
N1,(LTR(N),N=1,NOLTR),
N2,(ITR(I),ISR(I),L1R(I),I=1,NLSMAX)
FORMAT(24I3)
***Stored in COMMON /SPSTAT/

JT =	Transferred total angular momentum, j_t
KPARIT =	0; No parity change in the reaction 1; Parity change
IST =	Transferred spin, $s_t (= s_1 = s_2)$
NLSMAX =	Maximum number of changed $(\ell_1 s_1 t_1)$ sets in the target system
NOLTR =	Number of transferred orbital angular momenta
N1 =	Dummy
LTR(NOLTR)	Transferred orbital angular momenta, ℓ_t
N2 =	Dummy
ITR(NLSMAX)	Isospin, $t_1 = t_2$
ISR(NLSMAX)	Spin, $s_1 (= s_2 = s_t)$
L1R(NLSMAX)	orbital angular momentum, ℓ_1

Line 18 : READ(7,14) LRP1MX,FACNR
FORMAT(I7,6F7.2)
***Stored in COMMON /CNORE/

LRP1MX = Maximum angular momentum of $\ell_r + 1$ in the PW approximation.
(By inputting LRP1MX=1, NR approximation can be obtained.)
FACNR = Recoil factor α. 1.0 is suggested. (See Formulation III.3.)

Line 19 : READ(7,10) (KCETN(N),N=1,2)
FORMAT(24I3)
***Stored in COMMON /FFCC/

KCETN(1) = 0; Central exchange form factor is calculated
1; not considered
KCETN(2) = 0; Tensor exchange form factor is calculated
1; not considered

Line 20 : READ(7,10) NEFMAX,MJMAX
FORMAT(24I3)
***Stored in COMMON /CTFAC/ for MJMAX

NEFMAX = Number of energies transferred. Used in DCP1 only.
MJMAX = Maximum value of m_{j_t} in Eq.(12)

Line 21 : READ(7,12) (OMEGR(NE),NE=1,NEFMAX)
FORMAT(10F7.3)
***Stored in COMMON /RELKIN/

OMEGR(NE) = Transferred energy

Line 22 : READ(7,11) NSTATE,KREAD
FORMAT(14I5)

NSTATE = Dummy
KREAD = Dummy

Line 23 : READ(7,14) KTRL1,(VRANG(N),N=1,6)
FORMAT(I7,6F7.2)
***Stored in COMMON /FFCC/ for VRANG(N)

KTRL1 = 1; Yukawa type interaction (Set equal to 1. Actually dummy)
2; Gaussian type interaction (Not used in the present program)
VRANG(N) = 6 different range parameters in the effective NN interaction

Line 24-35 : READ(7,13) (VSTR(K,N),K=1,12, N=1,6)
FORMAT(7F10.4)
***Stored in COMMON /FFCC/

VSTR(K,N) = Strength parameters of Love-Franey interaction
[W. G. Love and M. A. Franey, Phys. Rev. **C15** (1977) 1396]

Line 36 : READ(7,12) TMAS,PMAS,ZZT,ZZP
FORMAT(10F7.3)
***Stored in COMMON /UNCPSA/

TMAS =	Mass of target system (Target mass -1)
PMAS =	Mass of valence nucleon
ZZT =	Charge of target system
ZZP =	Charge of valence nucleon

Line 37 : READ(7,12) VSX,VSOR,DFNR,DFNSO,RZR,RZSO,RZC
FORMAT(10F7.3)
***Stored in COMMON /BSX/

	Bound state potential parameters
VSX =	Depth parameter
VSOR =	Spin-orbit depth parameter
DFNR =	Diffuseness parameter
DFNSO =	Spin-orbit diffuseness parameter
RZR =	Radius parameter
RXSO =	Spin-orbit radius parameter
RZC =	Coulomb radius parameter

Line 38 : READ(7,10) NOSP,NOSH
FORMAT(24I3)
***Stored in COMMON /SPSTAT/

NOSP =	Number of particle states
NOSH =	Number of hole states

Line 38 : READ(7,15) (ESP(N),NSP(N),LSP(N),JTWP(N),ITP(N),ITZP(N), N=1,NOSP)
$+\alpha$
α=NOSP FORMAT(F10.5,5I5)
***Stored in COMMON /SPSTAT/

Single particle state information

ESP(N) =	energy
NSP(N) =	Number of nodes
LSP(N) =	orbital angular momentum
JTWP(N) =	Twice of total angular momentum
ITP(N) =	Isospin T (Not used)
ITZP(N) =	1; neutron, -1; proton

Line 38 : READ(7,15) (ESH(N),NSH(N),LSH(N),JTWH(N),ITH(N),ITZH(N), N=1,NOSH)
$+\alpha+\beta$
β = NOSH FORMAT(F10.5,5I5)
***Stored in COMMON /SPSTAT/

Single hole state information

ESH(N) =	energy
NSH(N) =	Number of nodes
LSH(N) =	orbital angular momentum
JTWH(N) =	Twice of total angular momentum
ITH(N) =	Twice isospin T (Not used)
ITZH(N) =	1; neutron, -1; proton

III. Subroutines

1. **DCP1** - Main Program

Sec. 1. Constants

1. This section defines

FACLOG(N)	Log of factorials
PI	π
HBAR	$\hbar c$ in MeV $\cdot$ fm
AMAS	atomic mass unit in MeV
WNUNIT	wave number unit in 1/fm
FINE	fine structure constant

2. It stores in COMMON /CONST/

Sec. 2. Read input data

See Chapter II. Input data

Sec. 3. Write initial output

This section prints (Tape 8) initial input data.

Sec. 4. Calculate the single particle and hole states.

1. This section calls BSAXON (See III.2 SUBROUTINE BSAXON.) which calculates the single particle wave functions, "NOSP" times,
2. It saves as "USAVP(NXMAX(4)*NOSP)"
3. It stores in COMMON/CBST/
4. It calls BSAXON which calculates the single hole wave functions "NOSH" times,
5. It saves as "USAVH(NXMAX(4)*NOSH)"
6. It stores in COMMON/CBST/
7. It prints information of ph states in the output.

Sec. 5. Find Particle-hole pairs

1. This section finds out particle-hole pairs which satisfies

Parity	$\ell_p + \ell_h + \pi = \text{even}$
Triangle relations of	(j_p, j_h, j_t)
Triangle relations of	(ℓ_p, ℓ_h, ℓ_1) with $\vec{\ell}_1 = \vec{s}_1 + \vec{j}_t$
Neutron hole only	

2. It saves "NPST(I), NHST(I), SPECTA(I), I= NPAIRD"
3. It stores in COMMON /SPSTAT/
4. It prints in the output.
5. It writes these single ph information on tape 12.

Sec. 6. Check field lengths of variables

1. This section calls "DCHECK1". (See III.3 SUBROUTINE DCHECK1.)

Sec. 7. Calculate the projectile wave functions and density

1. Deuteron

a. Hulthen wavefunction

$$\begin{aligned}
\phi_d &= \frac{1}{\sqrt{4\pi}}\frac{u(r)}{r} \\
u(r) &= \sqrt{\frac{2\alpha\beta(\alpha+\beta)}{(\beta-\alpha)^2}}(e^{-\alpha r} - e^{-\beta r})
\end{aligned}$$

where $\alpha^{-1} = 4.3/2.0$ fm, and $\beta = 7\alpha$.
1) It calculates W1=$u(r)/r$
2) It saves as "USAVEX(NX)=W1", "FOLD(NX)=W1*W1*2"
3) It stores in COMMON/CDENS/ for USAVEX(NX) and COMMON/OPTO/ for FOLD(NX)

b. Scattering wave

$$\begin{aligned}
\phi_d &= \frac{1}{kr}\sin\delta e^{i\delta}(\cot\delta\sin kr + \cos kr - e^{-\eta r}) \\
&= \frac{1}{kr}e^{i\delta}[\sin(kr+\delta) - \sin\delta e^{-\eta r}]
\end{aligned}$$

where $k = \sqrt{\frac{2\mu E}{\hbar^2}} = k_{unit}\sqrt{\mu E} = k_{unit}\sqrt{0.5E}, \quad E = 2.224.$
1) It calculates $W2 = \sqrt{4\pi}\phi_d$, with $\delta = 0.0$, and $\eta = 0.0$
2) It saves as "USAVEX(NX+N2MAX)=W2", "DENSTY(NX)=W1*W2"
3) It stores in COMMON/CDENS/
4) It prints DENSTY(NX)

2. ^{3}He

a. Gaussian density

$$\rho_0(r) = \frac{z}{8\pi^{3/2}}[\frac{1}{a^3}\exp(\frac{-r^2}{4a^2}) - \frac{c^2(6b^2-r^2)}{4b^7}\exp(\frac{-r^2}{4b^2})]$$

1) It calculates "FOLD(NX)= $4\pi\rho_0(r)$" with $a = 0.675,\ b = 0.836,\ c = 0.366\ z = 3.$ fm.
2) It calculates "DENSTY(NX)= $4\pi\rho_0(r)$" with $a = 0.654,\ b = 0.821,\ c = 0.456\ z = 1.$ fm.
Note that FOLD is normalized to 3, and DENSTY to 1.
3) It calculates "USAVEX(NX)= SQRT(DENSTY(NX))"
4) It stores in COMMON/CDENS/ for USAVEX(NX), DENSTY(NX) and COMMON/OPTO/ for FOLD(NX)
5) It prints DENSTY(NX)

3. Nuclei greater than mass 4

1) It reads bound state information (3 lines)
2) It calls BSAXON NOSTX(INPUT)-times
3) It saves as "USAVEX(NX+NOSTX*N2MAX)= URSAVE(NX)"
4) It calculates "DENSTY(NX)"
5) It stores in COMMON/CDENS/ for USAVEX(NX),DENSTY(NX)
6) It prints DENSTY(NX)

Sec. 8. Calculate distorted waves in the incident and exit channels.

1. This section calculates the kinematical variables, such as E_{cm}, k, η, in the incident and exit channels.
2. It saves as "ECM=E_{cm}, WN=k, CE=η
   Note that KTRLD(9) calculates relativistically.
3. It stores in COMMON/DWCC/
4. It calls OPT (See III.4 SUBROUTINE OPT.) twice to calculate distorted waves in the incident and exit channels.
5. It prints OMP parameters and kinematical variables in the incident and exit channels.

Sec. 9. Calculate α coefficients of Eq.(13).

1. This section calls AFACAL. (See III.5 SUBROUTINE AFACAL.)

Sec. 10. Calculate the distortion factors.

1. This section calls XLMCAL to calculate the distortion factors. (See III.11 SUBROUTINE XLMCAL.)

Sec. 11. Calculate direct form factors.

1. This section calls FFCALD or FFCALDM. (See III.9 SUBROUTINE FFCALD and III.10 SUBROUTINE FFCALDM.)

Sec. 12. Calculate exchange form factors.

1. This section calls FFCALE or FFCALEM. (See III. 12 SUBROUTINE FFCALE and III.19 FFCALEM.) If KTRLD(1)=1, calls FFCALEM, otherwise FFCALE.

Sec. 13. Calculate differential cross sections.

1. This calls CROSS (See III.20 SUBROUTINE CROSS.)

2. SUBROUTINE BSAXON and UNCPST

1. SUBROUTINE BSAXON solves the Schrödinger equation to obtain the single particle (hole) radial wave functions

$$[-\frac{\hbar^2}{2\mu}\frac{d^2}{dr^2} + \frac{\hbar^2\ell(\ell+1)}{2\mu r^2} + V(r)]u_{nj\ell s}(r) \quad = \quad -|E_0|u_{nj\ell s}(r)$$

where $u_{nj\ell s}(r)$ is defined as

$$\varphi(r,\theta,\phi) = R_{nj\ell s}(r)Y_{\ell m}(\theta,\phi) = \frac{u_{nj\ell s}(r)}{r}Y_{\ell m}(\theta,\phi)$$

and $V(r)$ includes the spin-orbit force.

2. It searches the depth of potential well (Woods-Saxon type), $V(r)$. (KTRL2=1 is set.) SUBROUTINE UNCPST generates outward solutions from the origin by using the 4-point Stormer method and inward solutions from the far outside the well by a single step Stormer method and matches two solutions at a boundary point.

3. It calls SUBROUTINE UNCPST, adjusts the potential depth until the matching conditions are met, and finds $u_{nj\ell s}(r)$ for a given $(nj\ell s)$ and a binding energy E_0.

Output

1. It saves $R_{nj\ell s}(r) = \frac{u_{nj\ell s}(r)}{r}$ as "URSAVE(NX)" with the maximum of NX = NXRAWF = $(r_0 \times m_A^{1/3} + 10 \times a_0)$ in steps of 0.1 fm.
2. It stores "URSAVE(NX)" in COMMON/BSX/
3. It prints the bound state information by inputting KTLOUT(1)=1, and further print the bound state potential by setting KOPOT=10 (Presently, KOPOT=0).

3. SUBROUTINE DCHECK1

1. Check field lengths of variables with a given input data set.

2. The meaning of parameters given in the program is following,

NHS	NOSH	
NPS	NOSP	
LXA	LDWMXR(1)	
LXB	LDWMXR(2)	
LPH	NPHMX	
LTL	L1R(NLSMAX)+LAMMXD(1)+1	
LM1	MP1MAX = MIN(LTR(NOLTR),MXMAX)+1	
LMM	MAX(LAMMXD(1,2,3))	
NXA	NXMXR(1)*PMASA	
NXB	NXMXR(2)	
NIN	MAX(NHDMX,NHEMX)	
N1X	NXMXR(4)/N1STEP+1	
KFM	2*NOLTR*NLSMAX	
KCC	GREATER THEN KFM	
LMJ	(MJMAX+1)*3	
KAMAX	(LAMMX+1)*(LAMMX+2)/2	GGRI(KA,NH,NLS)
KBMAX	NLSMAX*LAM1MX*(LAM1MX+1)/2	
KCMAX	KBMAX*LAM2MX*(LAM2MX+1)/2	QC(KC,NH)
KFMAX	NLSMAX*7	

4. SUBROUTINE OPT, POTEN, GAUSF, and DISWAVE

1. SUBROUTINE OPT(IDCHNL,NEXPT) calculates distorted waves for the incident(IDCHNL=1) and exit(IDCHNL=2) channels. NEXPT is set to be 1, but dummy.[1]

2. It calls SUBROUTINE FLGLCH first to calculate the partial Coulomb waves.[2]

3. It calls SUBROUTINE POTEN(IDCHNL,IDC) secondly to calculate the Woods-Saxon type potential for the target-nucleon system. IDC is set to be 1 presently, and will be used for the plane wave approximation.

4. In order to prepare the folding potential, POTEN calls SUBROUTINE GAUSF(NGAUS,RTS,WGT) where the abscissas (RTS) and weighting factors (WGT) for gaussian integration with the number of gaussian points, n=NGAUS are generated. Presently NGAUS=24 is set.

5. It calculates a single folding potential (KTRLD(3)=2, a double folding potential KTRLD(3)=1, not included presently.) The single folding potential is defined as (See Formulation note A-6)

$$\begin{aligned} V(r) &= \int d\vec{r}_1 \rho(r_1) v(|\vec{r}-\vec{r}_1|) \\ v(|\vec{r}-\vec{r}_1|) &= \sum_\ell v_\ell(r,r_1)(2\ell+1)P_\ell(\cos\theta) \\ v_\ell(r,r_1) &= \frac{1}{2}\int_{-1}^{1} v(|\vec{r}-\vec{r}_1|)P_\ell(\cos\theta)d\mu \end{aligned}$$

with $|\vec{r}-\vec{r}_1|^2 = r^2 + r_1^2 - 2rr_1\mu$. The diagonal optical potential is the monopole part ($\ell = 0$) of the above equation. Thus the single folding optical (complex) potential becomes

$$\begin{aligned} V_0(r) &= 4\pi \int dr_1 r_1^2 \rho(r_1) v_0(r,r_1), \quad 4\pi = ???? \\ v_0(r,r_1) &= \frac{1}{2}\int_{-1}^{1} v(|\vec{r}-\vec{r}_1|)d\mu \end{aligned}$$

The last integration is done by the gaussian integration method. The Coulomb potential is added to V_0. The final optical potential is named as "VRIT(NX)", printed in the output, and saved in COMMON/POTCC/.

6. It finally calls SUBROUTINE DISWAVE where the internal partial wave functions are generated from the origin and matched with the Coulomb wave functions. The final distorted partial wave functions and the partial c-matrix are saved as "DISWT(NX)" and "CL(L)", respectively.

Output

1. OPT Saves "DISWT(NX)*R*R" as "DISWA(NX*LA)" for the incident channel (IDCHAN=1) and "DISWB(NX*LB)" for the exit channel (IDCHAN=2)
2. It stores them in COMMON//
3. It prints each distorted partial wave by inputting KTLOUT(2)=NSTEP, where NSTEP is the step of NX in the output.
4. With KTLOUT(5) not equal to 0, it calculates and prints the elastic scattering information, i.e., c-matrix, S-matrix, $d\sigma(\theta)/d\Omega$, ratio $d\sigma/d\sigma_{Ruth}$ and so on.

[1] T. Tamura and W. R. Coker, "Computers and the optical model for nuclear scattering", in the book edited by S. Fernbach and A. H. Taub, "Computers and their role in the physical sciences" (Gordon and Breach, 1967?) for details.

[2] A. R. Barnett, D. H. Feng, J. W. Steed and L. J. B. Goldfarb, Comp. Phys. Comm. **8**, (1974) 377 for details.

5. SUBROUTINE AFACAL, CLEB, and RAC7

1. SUBROUTINE AFACAL calculates the expansion coefficient $\alpha^{j_t s_t \nu_1}_{t_1 s_1 \ell_1 k \ell_t}$, Eq.(13) in USO, defined as

$$\begin{aligned}
\alpha^{j_t s_t \nu_1}_{t_1 s_1 \ell_1 k \ell_t} &= W(s_t \ell_t s_t \ell_1; j_t k) \hat{s}_t^{-1} \hat{t}_1^{-1} < b||[c^\dagger c]_{s_1 t_1 \tilde{\nu}_1}||a > \\
&= W(s_t \ell_t s_t \ell_1; j_t k) \times M \times L \\
L &= (1 - (-)^{s_a + t_a}) \hat{s}_a \hat{s}_b W(s_a \frac{1}{2} s_1 \frac{1}{2}; \frac{1}{2} s_b) \hat{t}_a \hat{t}_b W(t_a \frac{1}{2} t_1 \frac{1}{2}; \frac{1}{2} t_b) \\
M &= (-)^{t_1 + \nu_1 + t_a - \nu_a} \hat{t}_1^{-1} < t_a \nu_a t_b, -\nu_b | t_1 \nu_1 >
\end{aligned}$$

See Formulation I-12 for details. The L-values for different reactions are

Reaction	s_a	s_b	s_1	t_a	t_b	t_1	L
(p,p')	1/2	1/2	0	1/2	1/2	0	1
(p,n)	1/2	1/2	0	1/2	1/2	1	1
(d,d')	1	1	s_1	0	0	0	$\sqrt{(3+s_1)(2-s_1)/2}\delta(t_1,0)$
$(d,2p)$	1	s_b	s_1	0	1	1	$\sqrt{6}\hat{s}_b W(1\frac{1}{2}s_1\frac{1}{2};\frac{1}{2}s_b)\delta(t_1,t_b)$
(h,t)	1/2	1/2	s_1	1/2	1/2	t_1	$4\delta(t_1,0)\delta(s_1,0) - (-)^{t_1+s_1}$
(α,α')	0	0	0	0	0	0	2

$$\sqrt{6}\hat{s}_b W(1\frac{1}{2}s_1\frac{1}{2};\frac{1}{2}s_b) = \sqrt{(3+s_1)(2-s_1)/2}, \quad s_b = 1$$

2. It calls SUBROUTINE CLEB and RAC7 whenever Clebsch-Gordan and Racah coefficients, respectively, are needed.

3. It defines C4=$W(s_t \ell_t s_t \ell_1; j_t k) \times M$.

4. The overall phase factor $(-)^{\ell_1 + k + \ell_t}$ is also multiplied.

Output

1. It saves α-coefficients as "ALPHA(NLSMAX*2,NOLTR)" where the second part of NLSMAX*2 is for the tensor part ($k = 2$).
2. It stores them in COMMON/SPECFC/
3. It prints ALPHA(NLSK,NLT) in the output.

6. SUBROUTINE FFCALD - Direct Form Factors

Sec. 1. Calculate the effective interaction for the direct form factors.

1. This section calls EFFINT(0). (See III.7 SUBROUTINE EFFINT.)

Sec. 2. Calculate the projectile density for the direct form factors.

1. This section calls PDENST(0). (See III.8 SUBROUTINE PDENST.)

Sec. 3. Calculate the target density.

1. This section calculates the target density $\rho^D_{T,\ell_1}(r_1)$ in Eq.(27) in USO, (See Formulation I.5 for details.)

$$\rho^D_{T,\ell_1}(r_1) \;=\; X(\ell_p\frac{1}{2}j_p,\ell_h\frac{1}{2}j_h;\ell_1 s_1 j_t) \;\; < I_B||[\hat{a}^\dagger_{j_p\nu_p}\hat{a}_{j_h\nu_h}]_{j_t}||I_A > \sqrt{4\pi}d_{\ell_p\ell_h\ell_1}R_{\ell_p}(r_1)R_{\ell_h}(r_1)$$

2. It calls SUBROUTINE NINEJ for X-factor and SUBROUTINE CLEBZ for d-factor.

3. The extra factors PHACO $=(-)^{\ell_p}$ and PHASE=ν??? is multiplied.

4. It saves as "TRHO(r_1,ℓ_1)=TRHO(N1,NLS)", where N1MAX=NXMAX(4)(input), NLSMAX=NLSMAX(input), stores them in COMMON// and prints TRHO(N1,NLS) with nonzero KTLOUT(3).

Sec. 4. Calculate a factor in the projectile density and sum over ℓ_t and ℓ_1.

1. This section calculates a factor in the projectile density $\rho^D_{P,\lambda_2\ell\lambda_1}$ in Eq.(26) in USO, (See Formulation II.1 for details.)

$$C_{kf,m} \;=\; \frac{2\pi}{\hat{k}^2}\hat{\ell}_t(\ell_t 0\ell_1 m|km)\alpha^{j_t s_t \nu_1}_{t_1 s_1 \ell_1 k \ell_t}, \qquad kf=(\ell_t,\ell_1,k)$$

2. It calls SUBROUTINE CLEB for CG coefficients.

3. It saves as "CLEBQ(kf,m)=CLEBQ(KF,M1)", where KFMAX=number of triangle relations of (ℓ_t,ℓ_1,k), M1MAX= MIN of $(k+1,\ell_1+1)$, stores them in COMMON/CGFAC1/ and prints KFMAX, LTMIN, LTMAX with nonzero KTLOUT(3).

Sec. 5. Prepare the Gaussian integrations.

1. This section calls SUBROUTINE GAUSF with NGAUS=NGAUSR(input). (See III.4 SUBROUTINE GAUSF.)

Sec. 6. Calculate the direct form factors.

1. The structure of this section is as follows.

1) For a given r_a (Big loop up to statement number 900. Calculates not every Δr_a, but NBSTEP$*\Delta r_a$ and interpolates later.) and r_1 (Loop up to 300), calculate the gaussian integration of Eq.(26) in USO.

$$\begin{aligned}\rho^D_{P,k\ell_t\ell_1}(r_a,r_1,r) &= \frac{2\pi}{\hat{k}^2}\sum_m \hat{\ell}_t(\ell_t 0\ell_1 m|km)\int \rho^D_{P,k}(r_a,r_1,\mu,r)Y_{km}(\theta'_2,0)Y^*_{\ell_1 m}(\theta,0)d\mu \\ (r'_2)^2 &= r_1^2+r_a^2-2r_1r_a\mu \qquad \mu\equiv\cos\theta=\hat{r}_a\cdot\hat{r}_1, \\ &\qquad\qquad\qquad\qquad\quad \mu'\equiv\cos\theta'=\hat{r}_a\cdot\hat{r}'_2=\frac{r_1\mu-r_a}{r'_2}\end{aligned}$$

2) Integration over r_1 is then performed up to the statement number 300 in the present program, and the result is named ”QA(kf, r)”

$$\text{QA} = \int dr_1 \rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r)\rho^D_{T,\ell_1}(r_1)$$

3) For the NN scattering (statement number 200 and 300),

$$\rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r) \;\; = \;\; \frac{2\pi}{\hat{k}^2}\sum_m \hat{\ell}_t(\ell_t 0 \ell_1 m|km)\hat{k}(-)^k(\frac{-1}{r_1 r_a r})Y_{km}(\theta_2', 0)Y^*_{\ell_1 m}(\theta, 0)$$

and integration over r_1 is performed to have ”QA(kf, r)”.

4) Integration over r immediately follows and it is named the form factor ”GGRI(kf, r_a).

$$f^D_{t_1 s_1 \ell_1 k \ell_t, m_{\ell_t}}(r_a) \;\; = \;\; (-)^{\ell_1}\hat{\ell}_t^{-1}\int r^2 dr V^D_{t_1 s_1 k}(r)\int dr_1 \rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r)\rho^D_{T,\ell_1}(r_1)$$

2. It saves as ”GGRI(kf, r_a)=GGRI(KF,NA)”, where KFMAX=number of triangle relations of (ℓ_t, ℓ_1, k), NAMAX=NXMAX(2), stores them in COMMON/CGGR/ and prints GGRI with nonzero KTLOUT(3).

Sec. 7. Calculate the direct transition amplitudes.

1. This section calls SUBROUTINE PACALD(1). (See III.9 SUBROUTINE PACALD(IDEXCH).)

7. SUBROUTINE EFFINT, NINEJ

1. SUBROUTINE EFFINT(KEXCH) calculates the interaction potential integrand of $V^i_{t_1 s_1 k}(r)$ of Eq.(16) in USO, i.e.,

$$V^i_{t_1 s_1 k}(r) \;\; = \;\; \sqrt{4\pi} f_k \hat{s}_1^2 \hat{t}_1^2 \sum_{st} \hat{s}^3 \hat{t}^3 \begin{Bmatrix} \frac{1}{2} & \frac{1}{2} & s \\ \frac{1}{2} & \frac{1}{2} & s \\ s_1 & s_1 & k \end{Bmatrix} \begin{Bmatrix} \frac{1}{2} & \frac{1}{2} & t \\ \frac{1}{2} & \frac{1}{2} & t \\ t_1 & t_1 & 0 \end{Bmatrix} P_i v^i_{tsk}(r)$$

Where $P_D = 1$ (KEXCH=0) and $P_E = (-)^{s+t+1}$ (KEXCH=1), while $f_0 = 1$ and $f_2 = \sqrt{8}$.

2. The set of (kt_1s_1) is (000)(001)(201)(010)(011)(211) and makes 6 elements, and real and imaginary parts make 12. The integer variable ”M” indicates the order of the set, e.g., M=10 means the imaginary part of (010).

3. The radial parts of interaction are calculated with Love-Franey interaction[3] of which strength parameters and range parameters are input as VSTR(M,N) and VRANG(N), respectively. Remember that the radial shape of tensor term is taken to be r^2 times a sum of Yukawa terms. For the direct term, r^2 is thus multiplied, while for the exchange term, this factor smears into the formulation.

4. EFFINT calls SUBROUTINE NINEJ twice for 9-j symbols of the spin and isospin parts.

Output

1. It Saves effective interactions as ”VV(NH,M)” where
NHMAX=INTRAN (NHDMX,NHEMX=input), and MMAX=12.
2. It stores them in COMMON/FFCC/
3. It prints VV(NH,M) with nonzero KTLOUT(3).

[3] W. G. Love and M. A. Franey, Phys. Rev. **C15** 1396 (1977), **C24** 1073 (1981), **C27** 438(E) (1983), and **C31** 488 (1985).

8. SUBROUTINE PDENST, YLCAL, GAUSF

1. SUBROUTINE PDENST(KEXCH=0) calculates the expansion coefficients $\rho^D_{P,\lambda_2}(r'_2, r)$ of Eq.(25) in USO, i.e.,

$$\begin{aligned}\rho^D_{P,\lambda_2}(r'_2, r) &= \sqrt{\pi}\int_{-1}^{1} \rho^D_{P,\lambda_2}(r'_2, r, \mu) Y_{\lambda_2 0}(\theta, 0) d\mu, \\ r_2^2 &= (r'_2)^2 + r^2 + 2r'_2 r\mu \quad \mu \equiv \cos\theta = \hat{r}\cdot\hat{r}'_2\end{aligned}$$

2. SUBROUTINE PDENST(KEXCH=1) calculates the expansion coefficients $\rho^E_{P,\lambda_2}(r'_2, r)$ of Eq.(31a) in USO, i.e.,

$$\begin{aligned}\rho^E_{P,\lambda_2}(r'_2, r) &= \sum_{\ell_2\eta_2} \hat{\ell}_2\hat{\eta}_2\omega_{\ell_2}(r'_2)(\eta_2 0\lambda_2 0|\ell_2 0) \\ &\quad\times \frac{2\pi}{2\ell_2+1}\sum_m \hat{\eta}_2(\eta_2 0\lambda_2 m|\ell_2 m)\int \omega_{\ell_2}(r_2) Y_{\ell_2 m}(\theta, 0) Y^*_{\lambda_2 m}(\theta', 0) d\mu \\ &\quad (\mu = \cos\theta = \vec{r}_2\cdot\vec{r}'_2, \quad \text{and} \quad \cos\theta' = \vec{r}'_2\cdot\vec{r})\end{aligned}$$

3. It calculates the direct part first in a straightforward way.

4. It returns for the nucleon scattering. (KTRLD(1)=3)

5. It then calculates the non-local projectile density for the exchange part.

6. It calls SUBROUTINE GAUSF to prepare gaussian integration. NGAUS=16 is set.

7. It calls SUBROUTINE YLCAL to obtain the spherical harmonics.

Output

1. It Saves expansion coefficients as "RHOD(r'_2, r, λ_2)=RHOD(N2,NH,LAMP1)",
where N2MAX=NXMAX(3)(input), NHMAX=INTRAN(NHDMX,NHEMX=input),
LAMP1MAX=LAMMXD(2)(input)+1, for both direct and exchange form factor.
2. It stores them in COMMON/CDENS/
3. It prints RHOD(N2,NH,LAMP1) with nonzero KTLOUT(3).

9. SUBROUTINE PACALD(IDEXCH)

1. SUBROUTINE PACALD(IDEXCH) calculates the overlap integrals of the direct form factor (IDEXCH=1) and the exchange form factor for no-recoil approximation (IDEXCH=2).

$$O^D_{t_1 s_1 \ell_1 k \ell_t, \ell_a \ell_b} = d_{\ell_a \ell_t \ell_b}\int dr_a \chi_{\ell_b}(r_a) f^D_{t_1 s_1 \ell_1 k \ell_t}(r_a)\chi_{\ell_a}(r_a)$$

2. It then calculates direct transition amplitudes.

$$T^i_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}} = \frac{(4\pi)^{3/2}}{k_a k_b}\sum_{\ell_a \ell_b} i^{\ell_a - \ell_b + \pi}\hat{\ell}_a(\ell_a 0 \ell_t m_{\ell_t}|\ell_b m_{\ell_t}) O^i_{t_1 s_1 \ell_1 k \ell_t, \ell_a \ell_b} Y_{\ell_b m_{\ell_t}}(\hat{k}_b)$$

3. The structure of this subroutine is as follows. It first calculates the vector coupling constants and other factors in the above equations, and saves them as "CLEBD(JLS,M1)", where JLS(MAX) (usually large number) will be printed.

4. It calls SUBROUTINE DSPLS3 to obtain GGRI(kf, r_b, m_{ℓ_t}) every Δr_b. (See Sec. 6.1.1.) DSPLS3(X,Y,N,XI,FI,M,Q,AU,IGO) builds up spline and interpolates M points with IGO=1. X, Y, and N are the initial x, y and the number of points, respectively, and XI, FI, M are final ones.

5. It then calculates final overlap integrals over r_b and saves as "AMPD(LBLT,M1)" for the direct part, and as "AMPE(LBLT,M1)" for the no-recoil exchange part.

6. It finally calls YLCAL to obtain $Y_{\ell_b m_{\ell_t}}(\hat{k}_b)$ and obtain the transition amplitudes.

7. It saves as "OVDD(ℓ_t, m_{ℓ_t})=OVDD(NLT,M1)", where NLT(MAX)=NOLTR=input, M1(MAX)= MP1MAX=MIN0(LTMAX+1,MXMAX+1) (MXMAX=input) for the direct part and "OVDE(NLT,M1)" for the no-recoil exchange part, and stores them in COMMON/OVDE/ and prints OVDD and OVDE, respectively.

10. SUBROUTINE FFCALDM - Modified Direct Form Factors (KKSU-paper)[4]

Sec. 1 to **Sec. 5** Exactly the same as SUBROUTINE FFCALD in Section III.6.

Sec. 6. Calculate CG coefficients in transition amplitudes in Eq.(12).

1. This section calculates $C = (s_t m_t j_t m_{j_t} | \ell_t m_{\ell_t})$ for a given $(s_t = 1, j_t)$
   (These are needed for source functions in FFCALD and FFCALEM.)
2. It saves as "TFAC(m_t, m_{j_t}, ℓ_t)", = C/sqrt(T1), T1=1 for $m_t = 0$, otherwise 2
3. It stores in COMMON/CTFAC/
4. It prints TFAC

Sec. 7. Calculate the direct form factors.

1. The structure of this section is as follows.

1) For given r_1 (Big loop up to statement number 400. Calculates not every Δr_1, but N1STEP$*\Delta r_1$ and interpolates later.) and r_a (Loop up to 300), calculate the gaussian integration of Eq.(26) in USO.

$$\begin{aligned}
\rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r) &= \frac{2\pi}{\hat{k}^2} \sum_m \hat{\ell}_t (\ell_t 0 \ell_1 m | km) \int \rho^D_{P,k}(r_a, r_1, \mu, r) Y_{km}(\theta'_2, 0) Y^*_{\ell_1 m}(\theta, 0) d\mu \\
(r'_2)^2 &= r_1^2 + r_a^2 - 2 r_1 r_a \mu \qquad \mu \equiv \cos\theta = \hat{r}_a \cdot \hat{r}_1, \\
&\qquad\qquad\qquad\qquad \mu' \equiv \cos\theta' = \hat{r}_a \cdot \hat{r}'_2 = \frac{r_1 \mu - r_a}{r'_2}
\end{aligned}$$

2) Integration over r_a (instead of r_1 in USO) is then performed up to the statement number 300 in the present program, and the result is named "QA(kf, r, m_{ℓ_t})"

$$\text{QA} = \int dr_a \rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r) I_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}}(r_a)$$

where I is defined in subroutine XLMCAL (See III.6.) instead of $\rho^D_{T,\ell_1}(r_1)$ in Eq.(27) in USO.
3) For the NN scattering (statement number 200 and 300),

$$\rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r) = \frac{2\pi}{\hat{k}^2} \sum_m \hat{\ell}_t (\ell_t 0 \ell_1 m | km) \hat{k} (-)^k (\frac{-1}{r_1 r_a r}) Y_{km}(\theta'_2, 0) Y^*_{\ell_1 m}(\theta, 0)$$

and integration over r_a is performed to have "QA(kf, r, m_{ℓ_t})".

4) Integration over r immediately follows and it is named the form factor "GGRI($kf, r_1, , m_{\ell_t}$).

$$f^D_{t_1 s_1 \ell_1 k \ell_t, m_{\ell_t}}(r_1) = (-)^{\ell_1} \hat{\ell}_t^{-1} \int r^2 dr V^D_{t_1 s_1 k}(r) \int dr_a \rho^D_{P,k\ell_t\ell_1}(r_a, r_1, r) I_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}}(r_a)$$

[4] B. T. Kim, D. P. Knobles, S. A. Stotts, and T. Udagawa, Phys. Rev. C61, 044611 (2000).

2. It saves as "GGRI(kf, r_1, m_{ℓ_t})=GGRI(KF,N1,M1)", where KFMAX=number of triangle relations of (ℓ_t, ℓ_1, k), N1MAX=NXMAX(4), M1MAX= MP1MAX=MIN0(LTMAX+1,MXMAX+1) (MXMAX=input), stores them in COMMON/CGGR/ and prints GGRI with nonzero KTLOUT(3).

Sec. 8. Calculate the overlap integrals.

1. It calls SUBROUTINE DSPLS3 to obtain GGRI($kf, r_1, , m_{\ell_t}$) every Δr_1. (See Sec. 6.1.1.) DSPLS3(X,Y,N,XI,FI,M,Q,AU,IGO) builds up spline and interpolates M points with IGO=1. X, Y, and N are the initial x, y and the number of points, respectively, and XI, FI, M are final ones.

2. It then calculates final overlap integrals over r_1,

$$O^D_{\ell_t, m_{\ell_t}} = \sum_{t_1 s_1 \ell_1 k} \int r_1^2 dr_1 f^D_{t_1 s_1 \ell_1 k \ell_t, m_{\ell_t}}(r_1) \rho^D_{T,\ell_1}(r_1)$$

3. It saves as "OVDD(ℓ_t, m_{ℓ_t})=OVDD(NLT,M1)", where NLT(MAX)=NOLTR=input, M1(MAX)= MP1MAX=MIN0(LTMAX+1,MXMAX+1) (MXMAX=input), and stores them in COMMON/OVDE/ and prints OVDD.

11. SUBROUTINE XLMCAL, YLCAL, CLEBZ, and CLEB

1. SUBROUTINE XLMCAL calculates the integrand of $T^i_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}}$ of Eqs.(18) and (19) in USO, so-called the distortion factor, i.e.,

$$\begin{aligned} I_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}} &\equiv \frac{(4\pi)^{3/2}}{k_a k_b} i^{\ell_a - \ell_b + \pi} \hat{\ell}_a (\ell_a 0 \ell_t m_{\ell_t} | \ell_b m_{\ell_t}) d_{\ell_a \ell_t \ell_b} \chi_{\ell_b}(r_a) \chi_{\ell_a}(r_a) Y_{\ell_b m_{\ell_t}}(\hat{k}_b) \\ d_{\ell_a \ell_t \ell_b} &\equiv \frac{1}{\sqrt{4\pi}} \hat{\ell}_a \hat{\ell}_t \hat{\ell}_b^{-1} (\ell_a 0 \ell_t 0 | \ell_b 0) \end{aligned}$$

2. It calls first SUBROUTINE YLCAL to calculate the spherical harmonics $Y_{\ell_b m_{\ell_t}}(\hat{k}_b)$ at the given angle $(\theta, 0)$, and YLCAL saves them as "P(LBP1,M1)" and transfers to XLMCAL.

3. It calls SUBROUTINE CLEB and CLEBZ whenever Clebsch-Gordan coefficients are needed. CLEBZ is for CG coefficients with zero m's.

Output

1. It Saves integrands as "XLMRI(LLP1,M1,NA)" where LLP1MAX=$\ell_a + \ell_t + 1$,
and M1MAX=MP1MAX=MIN0(LTMAX+1,MXMAX+1) (MXMAX=input).
2. It stores them in COMMON/CXLM/
3. It prints XLMRI(LLP1,M1,NA) with nonzero KTLOUT(3).

12. SUBROUTINE FFCALE - Exchange Form Factors

Sec. 1. Check the field lengths of variables for the exchange form factors.

1. Calls DCHECK2 (See III.11 SUBROUTINE DCHECK2.)

Sec. 2. Calculate the effective interaction for the exchange form factors.

1. Calls EFFINT(1). (See III.7 SUBROUTINE EFFINT.)

Sec. 3. Calculate the projectile and target density.

1. Calls TRECAL (See III.12 SUBROUTINE TRECAL.)

Sec. 4. Define the basic variables and constants.

1. This section rewinds tapes 13, 14 and 15, and defines various basic variables and constants for the calculations.

Sec. 5. Calculate the central and tensor exchange form factors.

1. This section has a big DO loop up to the statement number 1000, for the central exchange form factors (KCT=1) and the tensor exchange form factors (KCT=2). By inputting 1 for KCETN(KCT) (Input data Line #19), you may exclude the corresponding form factors.

2. It calls SUBROUTINE GAUSF with NGAUS=NGAUSR(input) for the Gaussian integrations. (See III.4 SUBROUTINE GAUSF.)

3. It then calls SUBROUTINE GFAC(KCT) to calculate the various coupling factors. (See III.13 SUBROUTINE GFAC.)

4. It starts next big DO loop up to the state number 900. In the DO loop, Section 6, 7, and 8 for a given r_b are performed.

Sec. 6. Calculate the integration in G-factors.

1. This section calculates the projectile density, $\rho^E_{P,\lambda_2\ell_a\ell}(r_b, r_1, r)$,

$$\rho^E_{P,\lambda_2\ell_a\ell}(r_b, r_1, r) = \frac{2\pi}{\hat{\lambda}_2^2}\sum_m \hat{\ell}_b(\ell_b 0 \ell m|\lambda_2 m)\int \rho^E_{P,\lambda_2}(r_b, r_1, \mu, r) Y_{\lambda_2 m}(\theta_2', 0) Y^*_{\ell m}(\theta, 0) d\mu$$

2. The non-local target density (See Section III.12 SUBROUTINE TRECAL.) is multiplied to the projectile density and integration over r_1 is performed.

$$\int r_1^2 dr_1 \rho^E_{P,\lambda_2\ell_b\ell_c}(r_b, r_1, r)\rho^E_{T,\ell_1\lambda_1\ell_c}(r_1, r)$$

3. The results are are saved as "QA(KC,NH)", where KCMAX = MAX (KB, λ_2, ℓ_b, (See SUBROUTINE GFAC.) and NH= r for a given (r_b, r_1). They are printed with KTLOUT(3) ≥ 2.

3. For a nucleon scattering, "QA" is $\rho^E_{T,\ell_1\ell_b\lambda}(r_b, r) \times \hat{\lambda}_2(-)^{\lambda_2}$, (See Formulation III.3.2.)

Sec. 7. Calculate $G-$ factors.

1. This section calculates G-factors,

$$G^k_{t_1 s_1 \ell_1 \ell_b \lambda}(r_b, r) = \frac{1}{\sqrt{4\pi}} r^{-k} V^E_{t_1 s_1 k}(r) \sum_{\lambda_1\lambda_2\ell_c} (-)^{\ell_b} \hat{\lambda}_1 \hat{\lambda}_2 (\lambda_1 0 \lambda_2 0|\lambda 0) W(\lambda_1 \lambda_2 \ell_1 \ell_b; \lambda \ell_c)$$

$$\times \int r_1^2 dr_1 \rho^E_{P,\lambda_2\ell_b\ell_c}(r_1,r)\rho^E_{T,\ell_1\lambda_1\ell_c}(r_1,r)$$

2. The results are named "GGRI(KA,NH,NLS)=GGRI($(\ell_b,\lambda),r,\ell_1)$)" and are printed with non-zero KTLOUT(3).

Sec. 8. Calculate c-factors.

1. This section first decomposes r into r_a and r_b in the G-factors,

$$G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r_a) = \frac{2\pi}{\hat{\ell}_t^2}\sum_{\ell_c\lambda}\hat{\ell}_c(\ell_c 0\lambda m_{\ell_t}|\ell_t m_{\ell_t})\int d\mu G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r)Y_{\lambda m_{\ell_t}}(\theta',\pi)Y^*_{\ell_\alpha m_{\ell_t}}(\theta,0)$$

2. It stores the results as "FFAB(K,M1,NAM)=FFAB($(\ell_\beta\ell_\alpha),m_{\ell_t},r_a$)"

3. It finally calculates the c-factor,

$$c_{t_1s_1\ell_1k,\ell_\alpha\ell_\beta}(r_b,r_a) = \frac{2\pi}{\hat{\ell}_1^2}\sum_{m_{\ell_1}}\hat{\ell}_\beta(\ell_\alpha m_{\ell_1}\ell_\beta 0|\ell_1 m_{\ell_1})\sum_{\ell_c\lambda}\hat{\ell}_c(\ell_c 0\lambda m_{\ell_1}|\ell_1 m_{\ell_1})$$
$$\times \int d\mu G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r)Y_{\lambda m_{\ell_1}}(\theta',\pi)Y^*_{\ell_\alpha m_{\ell_1}}(\theta,0)$$

4. It stores the results as "FFRIT(JLS), JLS=(ℓ_α, ℓ_β,NLS)". Remember that this is the exchange form factors for central interactions. (See Formulation II.2.)

5. It writes on tapes as

Transitions	Exchange Interactions	Tapes
Direct	Central Only	15
Direct	Tensor	13
Exchange	Both	14

6. It then calculates the radial exchange form factor in the no-recoil approximation. (See Formulation III.2.)

$$f^{NR}_{t_1s_1\ell_1k\ell_t}(r_b) = \sqrt{4\pi}(-)^k\hat{\ell}_1\hat{\ell}_t^{-1}\int dr\ r^{k+2}G^k_{t_1s_1\ell_1\ell_t\lambda}(r_b,r)$$

It saves as "GGRIE(NLT,NBM)=GGRIE($t_1s_1\ell_1k\ell_t,r_b$)" and stores in COMMON/CGGR/.

7. It concludes the big DO loop for a given r_b.

Sec. 9. Calculate the exchange form factors.

1. It calls SUBROUTINE TENSOR (See Section 16.) to calculate final form factors for the tensor interaction.

$$f^E_{t_1s_1\ell_1k\ell_t,\ell_b\ell_a}(r_b,r_a) = J\ 4\pi\ m_a^k\sum_{\lambda_a\lambda_b\ell_\alpha\ell_\beta}[\frac{(2k+1)!}{(2\lambda_a+1)!(2\lambda_b+1)!}]^{1/2}\delta_{\lambda_a+\lambda_b,k}\ (-r_a)^{\lambda_a}(r_b)^{\lambda_b}$$
$$\times X(\ell_\alpha\lambda_a\ell_a,\ell_\beta\lambda_b\ell_b;\ell_1k\ell_t)d_{\ell_\alpha\lambda_a\ell_a}d_{\ell_\beta\lambda_b\ell_b}\ c_{t_1s_1\ell_1k,\ell_\alpha\ell_\beta}(r_b,r_a)$$

13. SUBROUTINE DCHECK2

1. Check field lengths of variables in the SUBROUTINE FFCALE.
2. The meaning of parameters given in the program is following,

NWTMAX	(LAM1MX+1)*(LAM1MX+2)/2	
KBMAX	NLSMAX*LAM1MX*(LAM1MX+2)/2	
KB	(NLS,LAM1,LPP)	
KCMAX	KBMAX*NOLTR*LAM2MX*(LAM2MX+1)/2	QC(KC,NH)
KC	(LT,KB,LAM2,LL)	
KAMAX	LAMMX*(LAMMX+1)/2	GGRI(KA,NH)
KA	(LL,LAM)	
KGMAX	KCMAX*LAMP1X	
KG	(KC,LAM)	
NLS	(LT1,L1,S)	
	PRESENT PROGRAM ASSUMES	
	NLSMAX.LE.5	
	LAM1MX.LE.5	
	KAMAX.LE.40	
	KBMAX.LE.30	
	KCMAX.LE.80	
	KGMAX.LE.125	
	NBLOCK.LE.360	
	RHOE(N,NH,LAMP1)	FFCALE, TRECAL, PDENST
	TRHO(N1 OR NB,NH,KB)	FFCALE, TRECAL
	GWT (N1 OR NB,NH,NWT)	FFCALE, TRECAL
	GGRI(KA,NH,NLS)	FFCALE ONLY
	QA(KC,NH)	FFCALE ONLY

14. SUBROUTINE TRECAL, TRECALM

1. SUBROUTINE TRECAL first calls SUBROUTINE PDENST(1) and calculates the non-local projectile density, $\rho^E_{P,\lambda_2}(r'_2, r)$ of Eq.(31a) in USO,

$$
\begin{aligned}
\rho^E_{P,\lambda_2}(r'_2, r) &= \sum_{\ell_2\eta_2} \hat{\ell}_2 \hat{\eta}_2 \omega_{\ell_2}(r'_2)(\eta_2 0 \lambda_2 0|\ell_2 0) \\
& \quad \times \frac{2\pi}{2\ell_2+1} \sum_m \hat{\eta}_2(\eta_2 0 \lambda_2 m|\ell_2 m) \int \omega_{\ell_2}(r_2) Y_{\ell_2 m}(\theta, 0) Y^*_{\lambda_2 m}(\theta', 0) d\mu \\
& \quad (\mu = \cos\theta = \vec{r}_2 \cdot \vec{r}'_2, \quad \text{and} \quad \cos\theta' = \vec{r}'_2 \cdot \vec{r})
\end{aligned}
$$

2. It then calculates the non-local target density, Eq. (31b) in USO,

$$
\begin{aligned}
\rho^E_{T,\ell_1\lambda_1\ell_c}(r_1, r) &= \sum_{ph,\eta_1} i^{\ell_p+\ell_h-\pi} X(\ell_p \frac{1}{2} j_p, \ell_h \frac{1}{2} j_h; \ell_1 s_1 j_t) \; < I_B||[\hat{a}^\dagger_{j_p\nu_p} \hat{a}_{j_h\nu_h}]_{j_t}||I_A > R_{\ell_p}(r_1) \\
& \quad \times (-)^{\eta_1} \hat{\ell}_h \hat{\ell}_c \hat{\eta}_1 (\ell_c 0 \eta_1 0|\ell_p 0) \; W(\ell_c \eta_1 \ell_1 \ell_h; \ell_p \lambda_1) \\
& \quad \times \frac{2\pi}{2\ell_h+1} \sum_{m_1} \hat{\eta}_1(\eta_1 0 \lambda_1 m_1|\ell_h m_1) \int R_{\ell_h}(r'_1) Y_{\ell_h m_1}(\theta, 0) Y^*_{\lambda_1 m_1}(\theta', 0) d\mu'
\end{aligned}
$$

3. It calls SUBROUTINE DENST(N,L) (See III.13.) to calculate the last line of the above equation.

4. It calls SUBROUTINE NINEJ, RAC7, CLEBZ for the vector coupling coefficients.

5. SUBROUTINE TRECALM is basically the same as TRECAL except that r_1 is interpolated and the target densities without the particle states are added.

Output

1. PDENST saves projectile density as "RHOE(r'_2, r, λ_2)=RHOD(N2,NH,LAMP1)", where N2MAX=NXMAX(3)(input), NHMAX=INTRAN(NHDMX,NHEMX=input),
LAMP1MAX=LAMMXD(2)(input)+1, for exchange form factor.
2. It stores them in COMMON/CDENS/, and prints RHOD(N2,NH,LAMP1) with nonzero KTLOUT(3).
3. TRECAL saves ρ_T as TRHO(N1,NH,KB) (KB=$(t_1 s_1 \ell_1), \lambda_1, \ell_c$), and stores them in COMMON//.

15. SUBROUTINE DENST(N,L), DENSTM(N,L)

1. SUBROUTINE DENST(N,L) calculates the last line of the non-local target density, Eq. (31b) in USO,

$$
\begin{aligned}
\rho^E_{T,\ell_1\lambda_1\ell_c}(r_1, r) &= \sum_{ph,\eta_1} i^{\ell_p+\ell_h-\pi} X(\ell_p \frac{1}{2} j_p, \ell_h \frac{1}{2} j_h; \ell_1 s_1 j_t) \; < I_B||[\hat{a}^\dagger_{j_p\nu_p} \hat{a}_{j_h\nu_h}]_{j_t}||I_A > \\
& \quad \times (-)^{\eta_1} \hat{\ell}_h \hat{\ell}_c \hat{\eta}_1 (\ell_c 0 \eta_1 0|\ell_p 0) \; W(\ell_c \eta_1 \ell_1 \ell_h; \ell_p \lambda_1) \\
& \quad \times \frac{2\pi}{2\ell_h+1} \sum_{m_1} \hat{\eta}_1(\eta_1 0 \lambda_1 m_1|\ell_h m_1) \int R_{\ell_h}(r'_1) Y_{\ell_h m_1}(\theta, 0) Y^*_{\lambda_1 m_1}(\theta', 0) d\mu' \\
G^{E,n_h\ell_h}_{T,\ell_1\lambda_1\ell_c}(r_1, r) &= \frac{2\pi}{2\ell_h+1} \sum_{m_1} \hat{\eta}_1(\eta_1 0 \lambda_1 m_1|\ell_h m_1) \int R_{\ell_h}(r'_1) Y_{\ell_h m_1}(\theta, 0) Y^*_{\lambda_1 m_1}(\theta', 0) d\mu' \\
& \quad (\mu = \cos\theta = \vec{r}_1 \cdot \vec{r}'_1, \quad \text{and} \quad \cos\theta' = \vec{r}_1 \cdot \vec{r})
\end{aligned}
$$

for a given $n_h = N, \ell_h = L$.

2. It calls SUBROUTINE GAUSF to prepare the gaussian integration, SUBROUTINE YLCAL for the spherical coefficients and SUBROUTINE CLEB for the CG coefficients.

3. It defines G as GWT(N1,NH,NGW) (NGW=λ_1, η_1), and stores them in COMMON//.

5. SUBROUTINE DENSTM is basically the same as DENST except that r_1 is interpolated.

16. SUBROUTINE GFAC(K), GFACM(K)

1. SUBROUTINE GFAC(K) defines various vector coupling factors for the exchange form factors for the given K=1 for the central interaction and K=2 for the tensor interaction.

2. It first calculates a coupling factor in the projectile density of (Eq.(32) in USO)

$$\begin{aligned}
\rho^E_{P,\lambda_2\ell_a\ell}(r_b,r_1,r) &= \frac{2\pi}{2\lambda_2+1}\sum_m \hat{\ell}_b(\ell_b0\ell m|\lambda_2 m)\int \rho^E_{P,\lambda_2}(r_b,r_1,\mu,r)Y_{\lambda_2 m}(\theta'_2,0)Y^*_{\ell m}(\theta,0)d\mu \\
\text{CLEBQ(KC,M2)} &= \frac{2\pi}{2\lambda_2+1}\hat{\ell}_b(\ell_b0\ell m_2|\lambda_2 m_2)\frac{2}{1+\delta_{m_20}} \\
\text{KC} &= (\ell,\lambda_2,\ell_b)
\end{aligned}$$

3. It then calculates a coupling factor in the G^k of (Eq.(33b) in USO)

$$\begin{aligned}
G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r) &= \frac{1}{\sqrt{4\pi}}r^{-k}V^E_{t_1s_1k}(r)\sum_{\lambda_1\lambda_2\ell_c}(-)^{\ell_b}\hat{\lambda}_1\hat{\lambda}_2(\lambda_10\lambda_20|\lambda0)W(\lambda_1\lambda_2\ell_1\ell_b;\lambda\ell_c) \\
&\qquad \times \int r_1^2dr_1\rho^E_{P,\lambda_2\ell_b\ell_c}(r_b,r_1,r)\rho^E_{T,\ell_1\lambda_1\ell_c}(r_1,r) \\
\text{COEF(KG)} &= \frac{1}{\sqrt{4\pi}}(-)^{\ell_b}\hat{\lambda}_1\hat{\lambda}_2(\lambda_10\lambda_20|\lambda0)W(\lambda_1\lambda_2\ell_1\ell_b;\lambda\ell_c) \\
\text{KG} &= (\text{KC},\lambda)
\end{aligned}$$

4. It finally calculates two coupling factors in c of (Eq.(35) in USO)

$$\begin{aligned}
c_{t_1s_1\ell_1k,\ell_\alpha\ell_\beta}(r_b,r_a) &= \frac{2\pi}{\hat{\ell}_1^2}\sum_{m_{\ell_1}}\hat{\ell}_\beta(\ell_\alpha m_{\ell_1}\ell_\beta0|\ell_1m_{\ell_1})\sum_{\ell_c\lambda}\hat{\ell}_c(\ell_c0\lambda m_{\ell_1}|\ell_1m_{\ell_1}) \\
&\qquad \times \int d\mu G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r)Y_{\lambda m_{\ell_1}}(\theta',\pi)Y^*_{\ell_\alpha m_{\ell_1}}(\theta,0) \\
&\qquad (\mu=\cos\theta=\hat{r}_b\cdot\hat{r}'_a, \text{ and } \cos\theta'=\hat{r}_b\cdot\hat{r}) \\
\text{CLEBF(KA,M1,NLS)} &= \frac{2\pi}{2\ell_1+1}\hat{\ell}_c(\ell_c0\lambda m|\ell_1m)\frac{2}{1+\delta_{m0}}(-)^m \\
\text{KA} &= (\ell_b\lambda) \\
\text{NLS} &= (t_1s_1\ell_1) \\
\text{CCDD(JLS,M1)} &= \hat{\ell}_\beta(\ell_\alpha m\ell_\beta0|\ell_1m) \\
\text{JLS} &= (\ell_\alpha,\ell_\beta,\text{NLS})
\end{aligned}$$

5. SUBROUTINE GFACM(K) calculates a coupling factor in the form factors for the recoil effects in the PW approximation, instead of the previous section 4.

$$\begin{aligned}
f^{PW}_{t_1s_1\ell_1k\ell_t}(r_b) &= \sqrt{4\pi}\sum_{\ell_b\lambda\ell_r}i^{\pi-\ell_r-\ell_b}\hat{k}\hat{\lambda}(k0\lambda0|\ell_r0)\hat{\ell}_1\hat{\ell}_rW(\ell_b\lambda\ell_tk:\ell_1\ell_r) \\
&\qquad \times(-)^{k+\ell_1-\ell_t}(\ell_bm_{\ell_t}\ell_r0|\ell_tm_{\ell_t})\int dr\ r^{k+2}G^k_{t_1s_1\ell_1\ell_b\lambda}(r_b,r)j_{\ell_r}(\alpha k_ar/a) \\
\text{CCDD(JLS)} &= i^{\pi-\ell_r-\ell_b}\hat{k}\hat{\lambda}(k0\lambda0|\ell_r0)\hat{\ell}_1\hat{\ell}_rW(\ell_b\lambda\ell_tk:\ell_1\ell_r)
\end{aligned}$$

17. SUBROUTINE TENSOR

1. SUBROUTINE TENSOR calculates exchange form factor $f^E_{t_1s_1\ell_1k\ell_t,\ell_b\ell_a}(r_b,r_a)$ in Eq.(36) in USO for k=2, i.e. for tensor interaction,

$$\begin{aligned} f^E_{t_1s_1\ell_1k\ell_t,\ell_b\ell_a}(r_b,r_a) &= J\ 4\pi\ m_a^k \sum_{\lambda_a\lambda_b\ell_\alpha\ell_\beta} [\frac{(2k+1)!}{(2\lambda_a+1)!(2\lambda_b+1)!}]^{1/2}\delta_{\lambda_a+\lambda_b,k}\ (-r_a)^{\lambda_a}(r_b)^{\lambda_b} \\ &\times\ X(\ell_\alpha\lambda_a\ell_a,\ell_\beta\lambda_b\ell_b;\ell_1\lambda\ell_t)d_{\ell_\alpha\lambda_a\ell_a}d_{\ell_\beta\lambda_b\ell_b}\ c_{t_1s_1\ell_1k,\ell_\alpha\ell_\beta}(r_b,r_a) \end{aligned}$$

Remember that multiplication of all geometrical factors becomes unity for central interaction ($k=0$). (See Formulation Section II.2.)

2. It first calculates the geometrical factor,

$$4\pi\ [\frac{(2k+1)!}{(2\lambda_a+1)!(2\lambda_b+1)!}]^{1/2}\delta_{\lambda_a+\lambda_b,k}X(\ell_\alpha\lambda_a\ell_a,\ell_\beta\lambda_b\ell_b;\ell_1\lambda\ell_t)d_{\ell_\alpha\lambda_a\ell_a}d_{\ell_\beta\lambda_b\ell_b}$$

and saves as "C(KJLS)" in a linearized form.

3. It reads c-factor (FFRIT) from Tape 14, multiplies all factors and sums over $\lambda_a\lambda_b\ell_\alpha\ell_\beta$, and then saves as "GGRIT(JLS)", where JLS=(ℓ_α, ℓ_β,NLS).

4. If the central interaction is considered (KCNTN(1).EQ.0), it reads c-factor for the central interaction from Tape 13, and adds to the tensor part. It then finally stores form factors as "GGRIT(JLS)" for every (r_b,r_a), and writes on Tape 15.

5. With nonzero KTLOUT(13), the tensor form factors are printed for a given ℓ_a.

18. SUBROUTINE PACALE

1. SUBROUTINE PACALE first calculates a geometrical factor in the transition amplitudes of Eq. (18) in USO,

$$\frac{(4\pi)^{3/2}}{k_ak_b} i^{\ell_a-\ell_b+\pi}\hat{\ell}_a(\ell_a0\ell_tm_{\ell_t}|\ell_bm_{\ell_t})O^i_{t_1s_1\ell_1k\ell_t,\ell_a\ell_b}$$

and stores as "CLEBD(JLS,M1)" where M1=m_{ℓ_t}.

2. It then calculates the overlap integrals of Eq.(19b) in USO

$$O^E_{t_1s_1\ell_1k\ell_t,\ell_a\ell_b} = J\int dr_b\int dr_a r_b r_a \chi_{\ell_b}(r_b)f^E_{t_1s_1\ell_1k\ell_t,\ell_a\ell_b}(r_b,r_a)\chi_{\ell_a}(r_a)$$

During the integration processes, it reads exchange form factors $f^E_{t_1s_1\ell_1k\ell_t,\ell_a\ell_b}(r_b,r_a)$ from Tape 15.

3. It finally obtains transition amplitudes except $Y_{\ell_bm_{\ell_t}}(\hat{k}_b)$ for a given (ℓ_a,ℓ_b),

$$T^i_{t_1s_1\ell_1k\ell_tm_{\ell_t}} = \frac{(4\pi)^{3/2}}{k_ak_b}\sum_{\ell_a\ell_b} i^{\ell_a-\ell_b+\pi}\hat{\ell}_a(\ell_a0\ell_tm_{\ell_t}|\ell_bm_{\ell_t})O^E_{t_1s_1\ell_1k\ell_t,\ell_a\ell_b}Y_{\ell_bm_{\ell_t}}(\hat{k}_b)$$

and sums over $(t_1s_1\ell_1,k=2,\ell_a)$.

4. The results are saved as "AMPE(LBLT,M1)" where LBLT=(ℓ_b, ℓ_t), and are stored in COMMON/AMPL/. It prints AMPE(LBLT,1) for $m_{\ell_t}=0$.

19. SUBROUTINE FFCALEM - Exchange Form Factors with recoil effects in PW approximation

Sec. 1. Check the field lengths of variables for the exchange form factors.

1. Calls DCHECK2 (See III.10 SUBROUTINE DCHECK2.)

Sec. 2. Calculate the effective interaction for the exchange form factors.

1. Calls EFFINT(1). (See III.7 SUBROUTINE EFFINT.)

Sec. 3. Calculate the projectile and target density.

1. Calls TRECALM (See III.12 SUBROUTINE TRECAL.)

Sec. 4. Calculate the Bessel functions for the recoil effect in the plane wave approximation.

1. This section calculates the Bessel functions $j_{\ell_r}(\alpha k_a r/a)$ for the recoil effect in the plane wave approximation. (See Formulation III.3.) The Bessel functions are used only for the tensor exchange form factors.

2. It saves as "$j_{\ell_r}(\alpha k_a r/a)$ = FJLD(NH,LRP1)", where LRP1 is ℓ_r, and prints FJLD with nonzero KTLOUT(3).

Sec. 5. Calculate the central and tensor exchange form factors.

1. This section has a big DO loop up to the statement number 1000, for the central exchange form factors (KCT=1) and the tensor exchange form factors (KCT=2). By inputting 1 for KCETN(KCT) (Input data Line #19), you may exclude the corresponding form factors.

2. It calls SUBROUTINE GAUSF with NGAUS=NGAUSR(input) for the Gaussian integrations. (See III.4 SUBROUTINE GAUSF.)

3. It then calls SUBROUTINE GFAC(KCT) to calculate the various coupling factors. (See III.13 SUBROUTINE GFAC.)

4. It starts next big DO loop up to the state number 900. In the DO loop, Section 6 and 7 for a given r_1 are performed.

Sec. 6. Calculate the projectile density and integration over r_b.

1. This section calculates the projectile density, $\rho^E_{P,\lambda_2\ell_a\ell}(r_b, r_1, r)$,

$$\rho^E_{P,\lambda_2\ell_a\ell}(r_b, r_1, r) \quad = \quad \frac{2\pi}{\hat{\lambda}_2^2}\sum_m \hat{\ell}_b(\ell_b 0 \ell m|\lambda_2 m)\int \rho^E_{P,\lambda_2}(r_b, r_1, \mu, r) Y_{\lambda_2 m}(\theta_2', 0) Y^*_{\ell m}(\theta, 0) d\mu$$

2. It saves as "QAT (KC,NH)", where KCMAX = MAX (KB, λ_2, ℓ_b, (See SUBROUTINE GFAC.) and NH= r for a given (r_b, r_1).

3. For a nucleon scattering, it calculates the projectile density as, (See Formulation III.2.)

$$\begin{aligned}\rho^E_{P,\lambda_2\ell_a\ell}(r_b, r_1, r) \quad &= \quad \frac{2\pi}{\hat{\lambda}_2^2}\sum_m \hat{\ell}_b(\ell_b 0 \ell m|\lambda_2 m)\hat{\lambda}_2(-)^{\lambda_2}(-\frac{1}{r_1 r_b r})Y_{\lambda_2 m}(-\mu_p)Y^*_{\ell m}(\mu) \\ \mu_p \quad &= \quad \frac{r_1^2 - r_b^2 - r^2}{2 r r_b}, \quad \mu \; = \; \frac{r_b^2 + r_1^2 - r^2}{2 r_1 r_b}\end{aligned}$$

4. The distortion factor, (See Section III.6 SUBROUTINE XLMCAL.) is multiplied to the

projectile density and integration is performed over r_b.

$$Z^E_{P,\lambda_2\ell_b\ell_c}(r_1,r) = \int dr_b \rho^E_{P,\lambda_2\ell_b\ell_c}(r_1,r) X_{\ell_t m_{\ell_t}}$$

5. It saves as "QA(KC,KH,NLTM1), where NLTM1=$(\ell_t m_{\ell_t})$ and thus NLTM1MAX=NOLTR * MP1MX.

Sec. 7. Calculate $G-$ factors and form factors.

1. This section basically calculates

$$\begin{aligned} G^k_{t_1 s_1 \ell_1 \ell_b \lambda}(r_1,r) &= \frac{1}{\sqrt{4\pi}} r^{-k} V^E_{t_1 s_1 k}(r) \sum_{\lambda_1\lambda_2\ell_c} (-)^{\ell_b} \hat{\lambda}_1 \hat{\lambda}_2 (\lambda_1 0 \lambda_2 0 | \lambda 0) W(\lambda_1 \lambda_2 \ell_1 \ell_b; \lambda \ell_c) \\ &\times \int r_1^2 dr_1 Z^E_{P,\lambda_2\ell_b\ell_c}(r_1,r) \rho^E_{T,\ell_1\lambda_1\ell_c}(r_1,r) \end{aligned}$$

2. For given r_1 (Big loop up to statement number 900. Calculates not every Δr_1, but N1STEP$*\Delta r_1$ and interpolates later.), it calculates the integrand of G-factor including the distortion factors and is named "GGRI(KA,NH,NLS,NLTM1)=GGRI($(\ell_b,\lambda), r, \ell_1, (\ell_t m_{\ell_t})$)

3. Integrating over r and summing over various angular momenta are performed, and the results are saved as "GGRIE(NLTM1,N1M)=GGRIE($(\ell_t m_{\ell_t}), r_1$)". The big DO loop over r_1 (Section 5.) is ended (Statement number 900).

4. Lastly, the central and tensor exchange form factors are added up and the large DO loop (Section 5.) is ended (Statement number 1000).

Sec. 8. Calculate the overlap integrals.

1. It calls SUBROUTINE DSPLS3 to obtain GGRIE(ℓ_t, m_{ℓ_t}) every Δr_1. (See Sec. 6.1.1.)

2. It then calculates final overlap integrals over r_1,

$$O^E_{\ell_t,m_{\ell_t}} = \int r_1^2 dr_1 f^D_{t_1 s_1 \ell_1 k \ell_t, m_{\ell_t}}(r_1)$$

3. It saves as "OVED(ℓ_t, m_{ℓ_t})=OVDD(NLT,M1)", where NLTMAX=input, M1MAX= MP1MAX=MIN0(LTMAX+1,MXMAX+1) (MXMAX=input), stores them in COMMON/OVDE/ and prints OVED.

20. SUBROUTINE CROSS

1. SUBROUTINE CROSS calculates the final differential cross sections,

$$\frac{d\sigma}{d\Omega} = \frac{\mu_a \mu_b}{(2\pi\hbar^2)^2} \frac{k_b}{k_a} \frac{1}{(2I_A + 1)(2s_a + 1)} |\sum_i \sum_{k\ell_1 t_1} \alpha^{j_t s_t \nu_1}_{t_1 s_1 \ell_1 k \ell_t} T^i_{t_1 s_1 \ell_1 k \ell_t m_{\ell_t}}|^2$$

2. This subroutine will be modified to include the angular distributions.