ResQED:

computer program for symbolic operator computations in Quantum Electrodynamics

Eugene V. Stefanovich

Mountain View, California

2002

CONTENTS

1. GENERAL

Purpose : ResQED program performs symbolic calculations with operators in Quantum Electrodynamics

Language : FORTRAN-90; source code is provided.

Platforms : the code has been compiled and tested only on Sun Solaris 5.7 (Unix). However, I belive, it should work on other platforms as well.

Disclaimer : As my recent tests show, this program is not bug-free. For example, the `MAT` keyword is not working as expected. If you are going to do some serious work with this code you should be prepared to learn how it is written, fix bugs, and probably use this code as a part of your own programming efforts. I am not planning to do anything with this code in the near future, but I would be happy to help you if you have problems/questions.

Contact : Please send your questions/comments to
Eugene Stefanovich, 2255 Showers Dr. Apt. 153, Mountain View, CA 94040, USA.
e-mail: eugenev@synopsys.com

2. INTRODUCTION

Calculations performed in QED and other quantum field theories are often time consuming. If we are interested in matrix elements of the S-operator, then there is available a powerful method of Feynman (covariant) diagrams which allows easily to write down integrals needed for calculation of amplitudes of different processes in a given perturbation order. However, sometimes we need non-covariant expressions, e.g., when we calculate the "clothed particle" Hamiltonian [1] using "old-fashioned" perturbation theory. A diagram technique can be developed for such cases as well, see, e.g., [2], however the calculations are still quite involved due to the absence of an analogue of the useful Wick theorem.

3. THEORY

This program is designed for QED in which there are three kinds of particles and corresponding particle operators: electrons (creation operators $a^\dagger$, annihilation operators a) and positrons ($b^\dagger$ and b) with mass m, and massless photons ($c^\dagger$ and c). The non-zero (anti)commutators of these operators are

$$\begin{aligned} \{a(\mathbf{p},\sigma), a^\dagger(\mathbf{p}',\sigma')\} &= \{b(\mathbf{p},\sigma), b^\dagger(\mathbf{p}',\sigma')\} \\ = \{a^\dagger(\mathbf{p},\sigma), a(\mathbf{p}',\sigma')\} &= \{b^\dagger(\mathbf{p},\sigma), b(\mathbf{p}',\sigma')\} = \delta(\mathbf{p}-\mathbf{p}')\delta_{\sigma\sigma'}, \end{aligned}$$

$$[c(\mathbf{p},\tau), c^\dagger(\mathbf{p}',\tau')] = -[c^\dagger(\mathbf{p},\tau), c(\mathbf{p}',\tau')] = \delta(\mathbf{p}-\mathbf{p}')\delta_{\tau\tau'},$$

All other commutators/anticommutators are zero. To simplify notation we assume summation over repeating indices and introduce operators

$$\begin{aligned} A_\alpha(\mathbf{p}) &= \sum_\sigma u_\alpha(\mathbf{p},\sigma) a(\mathbf{p},\sigma), \\ A^\dagger_\alpha(\mathbf{p}) &= \sum_\sigma \overline{u}_\alpha(\mathbf{p},\sigma) a^\dagger(\mathbf{p},\sigma), \\ B_\alpha(\mathbf{p}) &= \sum_\sigma \overline{v}_\alpha(\mathbf{p},\sigma) b(\mathbf{p},\sigma), \\ B^\dagger_\alpha(\mathbf{p}) &= \sum_\sigma v_\alpha(\mathbf{p},\sigma) b^\dagger(\mathbf{p},\sigma), \\ C_{\alpha\beta}(\mathbf{k}) &= \sum_\tau e_\mu(\mathbf{k},\tau)\gamma^\mu_{\alpha\beta} c(\mathbf{k},\tau), \\ C^\dagger_{\alpha\beta}(\mathbf{k}) &= \sum_\tau e^*_\mu(\mathbf{k},\tau)\gamma^\mu_{\alpha\beta} c^\dagger(\mathbf{k},\tau) \end{aligned}$$

depending on spinor indices α, β. Here u, v are 4-component (spinor) solutions of the free Dirac equation, and e is 4-component photon polarization function. The non-zero (anti)commutators of these operators are (see [3])

$$\begin{aligned} \{A_\alpha(\mathbf{p}), A^\dagger_\beta(\mathbf{p}')\} &= \delta(\mathbf{p}-\mathbf{p}')\sum_\sigma u_\alpha(\mathbf{p},\sigma)\overline{u}_\beta(\mathbf{p},\sigma) \\ &= \frac{1}{2\omega_{\mathbf{p}}}(-ip_\mu\gamma^\mu + m)\beta\delta(\mathbf{p}-\mathbf{p}') \\ &\equiv U_{\alpha\beta}(\mathbf{p})\delta(\mathbf{p}-\mathbf{p}') \\ \{B_\alpha(\mathbf{p}), B^\dagger_\beta(\mathbf{p}')\} &= \delta(\mathbf{p}-\mathbf{p}')\sum_\sigma v_\alpha(\mathbf{p},\sigma)\overline{v}_\beta(\mathbf{p},\sigma) \\ &= \frac{1}{2\omega_{\mathbf{p}}}(-ip_\mu\gamma^\mu - m)\beta\delta(\mathbf{p}-\mathbf{p}') \\ &\equiv V_{\alpha\beta}(\mathbf{p})\delta(\mathbf{p}-\mathbf{p}') \end{aligned}$$

$$
\begin{aligned}
[C_{\alpha\beta}(\mathbf{p}), C^\dagger_{\gamma\delta}(\mathbf{p}')] &= \delta(\mathbf{p}-\mathbf{p}')\sum_\tau e_\mu(\mathbf{p},\tau)\gamma^\mu_{\alpha\beta}e^*_\nu(\mathbf{p},\tau)\gamma^\nu_{\gamma\delta} \\
&= (\delta_{ij} - \frac{p_i p_j}{p^2})\gamma^i_{\alpha\beta}\gamma^j_{\gamma\delta}\delta(\mathbf{p}-\mathbf{p}') \qquad (1)\\
&\equiv E_{\alpha\beta\gamma\delta}(\mathbf{p})\delta(\mathbf{p}-\mathbf{p}')
\end{aligned}
$$

Here functions $U_{\alpha\beta}(\mathbf{p})$, $V_{\alpha\beta}(\mathbf{p})$, and $E_{\alpha\beta\gamma\delta}(\mathbf{p})$ are referred to as electron, positron, and photon *propagators*, respectively.

In the above notation the quantum fields for photons $a_\mu(\mathbf{x},t)$ and electrons $\psi_\beta(\mathbf{x},t)$ are

$$
\begin{aligned}
a_\mu(\mathbf{x},t) &= (2\pi)^{-3/2}\int_{k\neq 0}\frac{d\mathbf{k}}{\sqrt{2|\mathbf{k}|}}\sum_\tau[e^{i\mathbf{k}\mathbf{x}-i|\mathbf{k}|t}e_\mu(\mathbf{k},\tau)c(\mathbf{k},\tau) + e^{-i\mathbf{k}\mathbf{x}+i|\mathbf{k}|t}e^*_\mu(\mathbf{k},\tau)c^\dagger(\mathbf{k},\tau)], \\
\gamma^\mu_{\alpha\beta}a_\mu(\mathbf{x},t) &= (2\pi)^{-3/2}\int_{k\neq 0}\frac{d\mathbf{k}}{\sqrt{2|\mathbf{k}|}}[e^{i\mathbf{k}\mathbf{x}-i|\mathbf{k}|t}C_{\alpha\beta}(\mathbf{k}) + e^{-i\mathbf{k}\mathbf{x}+i|\mathbf{k}|t}C^\dagger_{\alpha\beta}(\mathbf{k})] \\
\psi_\beta(\mathbf{x},t) &= (2\pi)^{-3/2}\int d\mathbf{p}[e^{i\mathbf{p}\mathbf{x}-i\omega_p t}A_\beta(\mathbf{p}) + e^{-i\mathbf{p}\mathbf{x}+i\omega_p t}B^\dagger_\beta(\mathbf{p})] \\
\overline{\psi}_\alpha(\mathbf{x},t) &= (2\pi)^{-3/2}\int d\mathbf{p}[e^{-i\mathbf{p}\mathbf{x}+i\omega_p t}A^\dagger_\alpha(\mathbf{p}) + e^{i\mathbf{p}\mathbf{x}-i\omega_p t}B_\alpha(\mathbf{p})]
\end{aligned}
$$

where $\omega_\mathbf{p} = \sqrt{m^2+\mathbf{p}^2}$. Using standard QFT approach in the Coulomb gauge [3] we obtain the Hamiltonian

$$
\begin{aligned}
H(t) &= H_0 + V(t) \\
&= e^{iH_0 t}(H_0+V)e^{-iH_0 t} \\
&= H_0 + V_1(t) + V_2^{Coul}(t) \qquad (2)
\end{aligned}
$$

[here and in what follows we denote the power of the coupling constant e (the perturbation order of an operator) by a subscript]. The free Hamiltonian is

$$
\begin{aligned}
H_0 &= \int d\mathbf{p}\omega_\mathbf{p}\sum_{\sigma=\pm 1/2}[a^\dagger(\mathbf{p},\sigma)a(\mathbf{p},\sigma) + b^\dagger(\mathbf{p},\sigma)b(\mathbf{p},\sigma)] \\
&+ \int_{k\neq 0} d\mathbf{k}|\mathbf{k}|\sum_{\tau=\pm 1}c^\dagger(\mathbf{k},\tau)c(\mathbf{k},\tau),
\end{aligned}
$$

The interaction V_1 is first-order

$$
V_1(t) = -\int d\mathbf{x}j^\mu(\mathbf{x},t)a_\mu(\mathbf{x},t) = ie(2\pi)^{-3/2}\int_{k\neq 0}\frac{d\mathbf{p}d\mathbf{k}}{\sqrt{2|\mathbf{k}|}}
$$

$$
\begin{aligned}
& (A^\dagger_\alpha(\mathbf{p}+\mathbf{k})A_\beta(\mathbf{p})C_{\alpha\beta}(\mathbf{k})e^{i(\omega_{\mathbf{p}+\mathbf{k}}-\omega_{\mathbf{p}}-|k|)t} \\
+ \; & A^\dagger_\alpha(\mathbf{p}-\mathbf{k})C^\dagger_{\alpha\beta}(\mathbf{k})A_\beta(\mathbf{p})e^{i(\omega_{\mathbf{p}-\mathbf{k}}-\omega_{\mathbf{p}}+|k|)t} \\
- \; & B^\dagger_\alpha(\mathbf{p}-\mathbf{k})C^\dagger_{\alpha\beta}(\mathbf{k})B_\beta(\mathbf{p})e^{i(\omega_{\mathbf{p}-\mathbf{k}}-\omega_{\mathbf{p}}+|k|)t} \\
- \; & B^\dagger_\alpha(\mathbf{p}+\mathbf{k})B_\beta(\mathbf{p})C_{\alpha\beta}(\mathbf{k})e^{i(\omega_{\mathbf{p}+\mathbf{k}}-\omega_{\mathbf{p}}-|k|)t} \\
+ \; & A^\dagger_\alpha(\mathbf{p}-\mathbf{k})B^\dagger_\beta(\mathbf{p})C^\dagger_{\alpha\beta}(\mathbf{k})e^{i(\omega_{\mathbf{p}-\mathbf{k}}+\omega_{\mathbf{p}}+|k|)t} \\
+ \; & A^\dagger_\alpha(\mathbf{p}-\mathbf{k})B^\dagger_\beta(\mathbf{p})C_{\alpha\beta}(\mathbf{k})e^{i(\omega_{\mathbf{p}-\mathbf{k}}+\omega_{\mathbf{p}}-|k|)t} \\
- \; & A_\alpha(\mathbf{p}-\mathbf{k})B_\beta(\mathbf{p})C_{\alpha\beta}(\mathbf{k})e^{i(-\omega_{\mathbf{p}-\mathbf{k}}-\omega_{\mathbf{p}}-|k|)t} \\
- \; & C^\dagger_{\alpha\beta}(\mathbf{k})A_\alpha(\mathbf{p}-\mathbf{k})B_\beta(\mathbf{p})e^{i(-\omega_{\mathbf{p}-\mathbf{k}}-\omega_{\mathbf{p}}+|k|)t})
\end{aligned} \tag{3}
$$

The total electron current is

$$j^\mu = -ie\overline{\psi_\alpha}\gamma^\mu_{\alpha\beta}\psi_\beta$$

The third term on the right hand side of (2) is the direct fermion-fermion interaction

$$V_2^{Coul}(t) = \int d\mathbf{x}d\mathbf{y}\frac{j_0(\mathbf{x},t)j_0(\mathbf{y},t)}{8\pi|\mathbf{x}-\mathbf{y}|}$$

which is needed to compensate the non-covariance of the photon propagator [3]. We may decide not to use the direct fermion-fermion interaction. Then we can use covariant photon propagator $g_{\mu\nu}$ instead of $(\delta_{ij} - \frac{p_ip_j}{p^2})$ in eq. (1) as explained in [3].

4. S-OPERATOR

4.1. “Old-fashioned” perturbation theory.

Results of experiments are normally described by matrix elements of the scattering operator (S-operator). The “old-fashioned” perturbation expansion of the S-operator is given by

$$S \;=\; 1 - i\int_{-\infty}^{+\infty} V(t)\,dt - \int_{-\infty}^{+\infty} V(t)\,dt \int_{-\infty}^{t} V(t')\,dt' + \ldots,$$

Using convenient symbols for time integrals $\underline{Y(t)} = \int_{-\infty}^{t} Y(t')dt'$ and $\underbrace{Y} = \int_{-\infty}^{+\infty} Y(t')dt'$ and often omit the time arguments of operators, the “old-fashioned” perturbation expansion of the S-operator is given by $S = 1 + \underbrace{\Sigma}$, where

$$\begin{aligned} \Sigma &= -iV_1 - V_1\underline{V_1} + iV_1\underline{V_1\underline{V_1}} + V_1\underline{V_1\underline{V_1\underline{V_1}}} + \ldots, \qquad (4)\\ V_1(t) &= e^{iH_0t}\hat{V}_1 e^{-iH_0t}e^{-\epsilon|t|}, \;\; \epsilon \to 0, \end{aligned}$$

and factor $e^{\epsilon t}$ describes the adiabatic switching of the interaction.

4.2. Magnus perturbation theory.

There is another equivalent representation suggested by W. Magnus [4] in which the S-operator is manifestly unitary: $S = e^{-iG}$, where

$$G = \underbrace{F}.$$

and Hermitian operator F is represented as a series of multiple commutators with time integrals

$$\begin{aligned} F \;=\;& V + \frac{i}{2}[\underline{V},V] - \frac{1}{6}[\underline{\underline{V},[V},V]] - \frac{1}{6}[\underline{[\underline{V},V]},V] - \frac{i}{12}[\underline{\underline{\underline{V},[[V},V]},V]] \\ -\;& \frac{i}{12}[\underline{[\underline{\underline{V},[V},V]]},V] - \frac{i}{12}[\underline{[\underline{\underline{V},V]},[V},V]] + \ldots \qquad (5) \end{aligned}$$

4.3. Feynman-Dyson perturbation theory.

The most familiar form is the covariant Feynman–Dyson perturbation expansion. It is equivalent to the two theories discussed above and will not be considered here.

5. ResQED PROGRAM

ResQED program is designed to assist in rather tedious calculations of terms in (4) and (5).

For example, just trilinear part V_1 of the interaction Hamiltonian in QED has 8 terms (3). When calculating commutator $[\underline{V}_1, V_1]$ which is present in the 2nd order S-operator (5) and in the "clothed particle" Hamiltonian [1, 2], one needs to calculate $8 \times 8 = 64$ commutators and convert to the normal form each of the resulting terms. In higher perturbation orders the number of terms and complexity of expressions grow very rapidly. So, it is desirable to have an automatic computerized way of such calculations.

ResQED program provides such a tool. The idea is similar to REDUCE or "Mathematica": Symbolic operator expressions are represented in ASCII format (see section xx) and written in *expression files*. The program uses formal rules to operate with these expression files: take normal products, commutators, etc. (full list of operations see in Table 5). The sequence of desired operations is written in "instructions file".

ResQED also attempts to simplify resulting expressions, e.g., given two operators with identical operator parts, the program will try to convert them to just one operator with combined numerical (momentum-dependent) factor.

Limitations : Calculations beyond 1-loop 4th perturbation order were not attempted. I suspect that the code should be modified to accomodate those.

6. EXPRESSION FILE

Each operator used in calculations is a sum of terms and is stored in a separate *expression file.*

Example of the expression file

```
                                                           line #  Comments

1 1 3 4 -6 1 4 0                                          |  1 |=> numerical factor
U(+h*)U(+g*)d*d*f*f*                                      |  2 |-\ propagator part
5 6 7 8 8 5 6 3 2 4 1 7                                   |  3 |-/
+[+Om(+g*-f*-d*)-Om(+f*)-Os(+g*-d*)]                      |  4 |-
+[+Om(+g*-f*-d*)-Om(+f*)-Os(+g*-d*)]                      |  5 | |
+[+Om(+h*)-Om(+g*)+Os(+h*-g*)]                            |  6 | | Energy
+[+Os(+h*-g*)]                                            |  7 | | denominators
+[+Os(+g*-d*)]                                            |  8 | |
x                                                         |  9 |-
+Ei-[+Om(+g*-f*-d*)+Om(+g*)-Om(+f*)-Om(+d*)]t             | 10 |-\ Time exponent
x                                                         | 11 |-/
A^(+d*)                                                   | 12 |-
1 0                                                       | 13 | |
A^(+f*)                                                   | 14 | |
2 0                                                       | 15 | | Creation/destruction
A_(+g*)                                                   | 16 | | operators
3 0                                                       | 17 | |
A_(-g*+f*+d*)                                             | 18 | |
4 0                                                       | 19 | |
x                                                         | 20 |-
1 1 3 4 -6 1 4 0
U(+h*)U(+d*)d*d*f*f*
5 6 7 8 8 3 2 4 6 7 1 5
+[+Om(+h*)-Om(+d*)+Os(+h*-d*)]
+[+Om(+g*)-Om(+d*)-Os(+g*-d*)]
+[+Om(+g*)-Om(+d*)-Os(+g*-d*)]
+[+Os(+h*-d*)]
+[+Os(+g*-d*)]
x
+Ei-[+Om(+g*-f*-d*)+Om(+g*)-Om(+f*)-Om(+d*)]t
x
A^(+d*)
1 0
```

```
A^(+f*)
2 0
A_(+g*)
3 0
A_(-g*+f*+d*)
4 0
x
```

This file describes an operator which is a sum of two terms.

$$
\begin{aligned}
- \quad & ie^4\pi^{-6}\int d\mathbf{d}d\mathbf{f}d\mathbf{g}\frac{U_{56}(\mathbf{h})U_{78}(\mathbf{g})E_{8563}E_{2417}e^{-i(\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}}+\omega_{\mathbf{g}}-\omega_{\mathbf{f}}-\omega_{\mathbf{d}})t}}{(\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}}-\omega_{\mathbf{f}}-|\mathbf{g}-\mathbf{d}|)^2(\omega_{\mathbf{h}}-\omega_{\mathbf{g}}+|\mathbf{h}-\mathbf{g}|)|\mathbf{h}-\mathbf{g}||\mathbf{g}-\mathbf{d}|} \\
& A_1^\dagger(\mathbf{d})A_2^\dagger(\mathbf{f})A_3(\mathbf{g})A_4(-\mathbf{g}+\mathbf{f}+\mathbf{d}) \qquad (6) \\
- \quad & ie^4\pi^{-6}\int d\mathbf{d}d\mathbf{f}d\mathbf{g}\frac{U_{56}(\mathbf{h})U_{78}(\mathbf{d})E_{8324}E_{6715}e^{-i(\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}}+\omega_{\mathbf{g}}-\omega_{\mathbf{f}}-\omega_{\mathbf{d}})t}}{(\omega_{\mathbf{h}}-\omega_{\mathbf{d}}+|\mathbf{h}-\mathbf{d}|)(\omega_{\mathbf{g}}-\omega_{\mathbf{d}}-|\mathbf{g}-\mathbf{d}|)^2|\mathbf{h}-\mathbf{d}||\mathbf{g}-\mathbf{d}|} \\
& A_1^\dagger(\mathbf{d})A_2^\dagger(\mathbf{f})A_3(\mathbf{g})A_4(-\mathbf{g}+\mathbf{f}+\mathbf{d}) \qquad (7)
\end{aligned}
$$

The first term is described in first 20 lines. Then goes the second term. The stuff on the right hand side (under `line #` and `Comments`) is not a part of the expression file and is given here only to facilitate discussion.

Each term in the exression file normally consists of 5 *records*:

- numerical factor
  - rational factor
  - power of i
  - power of electron charge (e)
  - power of π
- propagator part
  - electron propagator $U_{\alpha\beta}(\mathbf{p})$;
  - positron propagator $V_{\alpha\beta}(\mathbf{p})$;
  - photon propagator $E_{\alpha\beta\gamma\delta}(\mathbf{p})$;
- energy denominators
- time exponent
- creation/destruction operators

- electron creation/destruction operators;
- positron creation/destruction operators;
- photon creation/destruction operators;

Before describing the syntax of these records (see section 8) let us focus on some basic notation rules.

7. BASIC NOTATION RULES

In order to have ASCII representation of operators like (6) and (7) we need to represent i) single particle momentum variable, ii) linear combination of momentum variables, iii) one-particle energy, and iv) linear combination of one-particle energies.

7.1. Momentum variable

Momentum variable is composed of three symbols

Table 1: Structure of momentum variable.

1st symbol	Sign (`+` or `-`)
2nd symbol	name of the variable which may take the following values: `d`, `f`, `g`, `h`, `i`, `k`, `l`, `m`, `n`, `o`, `p`, `r`, `s`, `t`, `w`, `x`, `y`, `z`.
3rd symbol	Dummy symbol `*`

Example: `-p*` means momentum $-\mathbf{p}$

7.2. Momentum expression

Each momentum expression is a sum of one or more momentum variables in parentheses.

Example: `(+g*-f*-d*)` means linear combination of momenta $\mathbf{g} - \mathbf{f} - \mathbf{d}$

7.3. Particle energy

The energy of electron or positron with momentum $\mathbf{p}$ is $\omega_p = \sqrt{\mathbf{p}^2 + m^2}$. The energy of a photon with momentum $\mathbf{p}$ is $|\mathbf{p}|$. In QED we also need a square root of the photon energy $\sqrt{|\mathbf{p}|}$ [1]. These quantities are represented in the following format:

Table 2: Structure of the particle energy.

1st symbol	sign (`+` or `-`)
2nd and 3rd symbols	Type of "energy" `Om` for $\omega_{\mathbf{p}}$ `Os` for $|\mathbf{p}|$ `Oz` for $\sqrt{|\mathbf{p}|}$
next symbols	momentum expression

Example:

[1] it seems that the factor of 2 in $\sqrt{2|\mathbf{p}|}$ (see eq. (3)) is missed in the code

`+Om(+g*-f*-d*)` corresponds to $\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}}$

7.4. Energy expression

Energy expression consists of i) total sign (+ or -); ii) a sum of one or more particle energies in square brackets

Example:

```
+[+Om(+g*-f*-d*)-Om(+f*)-Os(+g*-d*)]
```

corresponds to

$$\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}} - \omega_{\mathbf{f}} - |\mathbf{g} - \mathbf{d}|$$

8. RECORDS OF THE TERM

Unless otherwise noted, all lines start from the first position without blank spaces. Line numbers used below refer to the sample expression file on page 10.

8.1. Numerical factor (line 1)

These are elements of the integer array `IPowr()` entered in free format: normally, separated by blank spaces.

- `IPowr(1)` ... numerator of the factor
- `IPowr(2)` ... denominator of the factor
- `IPowr(3)` ... power of i
- `IPowr(4)` ... power of e
- `IPowr(5)` ... power of π (numerator)
- `IPowr(6)` ... power of π (denominator)
- `IPowr(7)` ... divergence index: twice the power of the integration variable in denominator (`IPowr(7)` is calculated by the program, you don't need to specify it)
- `IPowr(8)` ... not used

Example:

```
1 1 3 4 -6 1 4 0
```

This line corresponds to factor $\frac{1}{1}(i)^3 e^4 \pi^{-6/1} = -\frac{ie^4}{\pi^6}$

8.2. Propagator part (lines 2, 3)

Propagator part takes 2 lines. The first line contains symbols of propagators. The second line contains spinor indices of propagators.

Symbols of propagators Electron (positron) propagator is written as a letter `U`(`V`) followed by momentum expression. Photon propagator is denoted by four symbols `letter*letter*`, where `letter` is any character from the list given in table 1. (Note that momentum variable is omitted.)

Example: `d*d*`

Spinor indices Electron (positron) propagator has 2 spinor indices. Photon propagator has 4 spinor indices. Normally these indices are denoted by Greek letters. In the program I use just integers `1,2,3,...`. The indices are written in a free format on one line in the order corresponding to the order of propagator symbols in the line above.

Example:

```
U(+h*)U(+g*)d*d*f*f*
5 6 7 8 8 5 6 3 2 4 1 7
```

corresponds to

$$U(\mathbf{h})_{56}U(\mathbf{g})_{78}E_{8563}E_{2417}$$

8.3. Energy denominators (lines 4 - 8)

Energy denominators are listed as energy expressions (one per line), e.g., line 4

```
+[+Om(+g*-f*-d*)-Om(+f*)-Os(+g*-d*)]
```

describes factor

$$\frac{1}{\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}} - \omega_{\mathbf{f}} - |\mathbf{g} - \mathbf{d}|},$$

Symbol `x` in the last line of this record indicates that there are no more energy denominators

8.4. Time exponent (lines 10, 11)

This section is a remnant from the earlier version of the code. It is probably redundant as all operators have time exponents which can be unambiguously determined just looking at the arguments of annihilation/creation operators.

Time exponent consists of 4 parts (see table 3)

Example:

```
+Ei-[+Om(+g*-f*-d*)+Om(+g*)-Om(+f*)-Om(+d*)]t
```

corresponds to time exponent $e^{-i(\omega_{\mathbf{g}-\mathbf{f}-\mathbf{d}}+\omega_{\mathbf{g}}-\omega_{\mathbf{f}}-\omega_{\mathbf{d}})t}$
If there is no time dependence, i.e., this is "renorm" term [1], then you should put symbol `x` in the first position instead of the time exponent.

Symbol `x` on line 11 indicates the end of the time exponent record.

Table 3: Syntax for time exponent

1st symbol	total sign (`+` or `-`)
2nd and 3rd symbols	`Ei` stands for e^i
next	energy expression
last symbol	`t` stands for time

8.5. Operator part (lines 12 - 20)

Each particle creation/destruction operator is written in two lines. First line has operator symbol (see table 4) followed by the momentum expression.

Second line has spinor indices (see section 8.2).

Example[2]:

```
A^(+d*)
1 0
```

is electron creation operator $A_1^\dagger(\mathbf{d})$

Example:

```
C_(+d*)
2 3
```

is photon destruction operator $C_{23}(\mathbf{d})$

[2]although electron (positron) operators need just one spinor index, the second (dummy) index should be entered as “0” to keep the reading routine happy.

9. EXPRESSION FILE DESCRIBING STATE

Expression files described above are designed to represent operators of the theory. They may be parts of the S-operator or Hamiltonian, etc. In order to calculate matrix elements of such operators we need to describe both initial and final states of the multiparticle system. These two states are specified in one file in the same format as operator part of an operator expression (section 8.5) with "small" particle operators (see Table xx). Destruction (creation) particle operators describe final (initial) state of the system. For example, expression file

```
a^(+d*)
1 0
a^(+f*)
2 0
c^(+g*)
3  4
a_(+h*)
5 0
a_(+k*)
6 0
```

desribes a final state with 2 electrons having momenta **h** and **k** and an initial state with 2 electrons (momenta **d** and **f**) and one photon with momentum **g**. Spinor indices do not play any role in this case. Spin of electrons/positrons and photon polarization in initial/final states are ignored.

Table 4: Creation/destruction operator symbols.

`A^`, `a^`	electron creation $A^\dagger$, $a^\dagger$
`A_`, `a_`	electron destruction A, a
`B^`, `b^`	positron creation $B^\dagger$, $b^\dagger$
`B_`, `b_`	positron destruction B, b
`C^`, `c^`	photon creation $C^\dagger$, $c^\dagger$
`C_`, `c_`	photon destruction C, c

10. INSTRUCTION FILE

The input format is free everywhere unless otherwise noted.

Example[3]:

```
      0                                                       |  1 |=> IPrint
 Grand calculation FD                                         |  2 |=> Title
INT                                                           |  3 |-\ First instruction
V                    Vint                                     |  4 |-/
MUL                                                           |  5 |-\
Vint                 Vinta                                    |  6 | | Second instruction
1  1  2                                                       |  7 |-/
PRO                                                           |  8 |-|
V                    Vinta                  VxVinta           |  9 | |
FIL                                                           | 10 | | Third instruction
A^A^A_A_                                                      | 11 | |
A^C^A_C_                                                      | 12 | |
END                                                           | 13 |-|
```

First two lines in the instruction file always have special meaning

line 1 `IPrint`

Parameter telling how to print:
Possible values are integers from 0 to 12
`IPrint`=0 ... bare minimum is printed
`IPrint`= 1-11 ... various grades of debugging output.
`IPrint`=12 ... a lot of debugging output with obscure meaning.

line 2 Title/Comments

Then go lines with actual instructions.

10.1. A single instruction

Each instruction takes at least 2 lines.

[3]symbols on the right hand side are not part of the instruction file and shown here only to facilitate discussion

first line of instruction `Keywrd`, `Order` (Format(A3,A2))

contains the keyword indicating the type of instruction. In the example above the Keyword `INT` tells that 'integration' is performed. The full list of keywords is given in Table xx.

`Order` is not used.

second line of instruction contains names of *input* (one or two) files and the *output* file. In the example above (on lines 3-4) operator from file `V` is 'integrated' and result is written in file `Vint`. Note that output file is written in the same "expression file" format as input file(s). Therefore, output of one instruction may be used as input of subsequent instructions allowing you to perform rather complicated calculations in one instruction file.

In addition to the output file the program generates three expression files named *Unphysical*, *Renorm*, and *Physical*. These files contain the same terms as the output expression file arranged according to their types [1]. Each new instruction overwrites previous files *Unphysical*, *Renorm*, and *Physical*.

Depending on the type of instruction additional lines may be needed. For example, in instruction `MUL`, the third line (line 7 in the example above) gives the value of the multiplicative factor f stored in three integers a, b, and c: $f = \frac{a}{b} i^c$

Instructions `MUL`, `PRO` and `COM` permit and instruction `DEL` requires additional lines with *templates* (see section 10.2). For `MUL`, `PRO` and `COM` keywords the templates section can be empty, i.e., consist of only keyword `END`. The format of 2nd, 3rd, etc lines of instructions is given in the 3rd column of Table 5.

Table 5: The full list of allowed keywords (operations).

Keyword	Operation	Usage: 2nd, 3rd, etc lines	Comments
`INT`	Indefinite integral	I O	$O = \int_{-\infty}^{t} I(t')dt$
`PRO`	Normal product	I1 I2 O [Templates]	$O = I1 \cdot I2$ Terms agreeing with templates are left in O
`COM`	Commutator	I1 I2 O [Templates]	$O = [I1, I2]$ Terms agreeing with templates are left in O
`SUM`	Sum	I1 I2 O	$O = I1 + I2$
`SOL`	Solve equation	I O	$[O, H_0] = iI1$
`MUL`	Numerical factor	I O INum IDen IPow [Templates]	$O = \frac{INum}{IDen}(i)^{IPow} I$
`MAT`	Matrix element	I S O	$O = \langle S_{in} \vert I \vert S_{out} \rangle$
`ENR`	Definite integral	I O	$O = \int_{-\infty}^{\infty} I(t)dt$
`DEL`	Delete terms	I O Templates	Terms agreeing with templates are deleted
`CON`	Remove unconnected diagrams	I O	$O = I^{conn}$

The symbols in the 3rd and 5th columns have the following meaning:

- I, I1, I2 ... names of input expression files;
- O ... name of output expression file;
- S ... input expression file with state (for keyword `MAT` only); S_{in} is initial state constructed from creation operators; S_{out} is final state constructed from destruction operators;
- fields in brackets [] are optional;
- INum ... numerator (integer);

- IDen ... denominator (integer);
- IPow ... power of i (integer); e.g., IPow=2 denotes $i^2 = -1$.

10.2. Templates

The template section consists of one or more lines containing templates and ends with a line containing `END` which indicates the end of the templates section. Each template occupies one line. Each template is a sequence of operator symbols (see table 4) which starts from 1st position without blank spaces. It allows to select from the result of instruction only operators agreeing with the template.

The use of templates with keywords `PRO`, `COM`, and `MUL` is optional[4]. If you use templates there, the template section must be preceded by keyword `FIL`. If you don't use templates, you should put keyword `END` in place of the template section.

Example:

```
FIL
A^B^A_B_C_
A^A_
END
```

this template selects operators of the type $A^\dagger B^\dagger ABC$ and $A^\dagger A$

[4]templates are mandatory with keyword DEL

11. INSTALLING AND TESTING THE CODE

To install the code you need to do the following

1. Download file ResQED.tar.gz (for Unix) or ResQED.zip (for Windows) from my web-site. The rest of instructions are for Unix.

2. Unzip and untar the ResQED.tar.gz file:

   `gunzip ResQED.tar.gz`

   `tar xvf ResQED.tar`

3. Go to the newly created directory ResQED. It has two subdirectories:

   `fortran_files` ... source code

   `Example` ... sample input/output files

4. Go to `fortran_files` and type `make` (you should have FORTRAN-90 compiler f90 installed in your system). The code will be compiled and linked; executable `cmt.exe` will be created.

5. To test basic functionality of the code go to directory `Example` and run test calculation

   ```
   ../fortran_files/cmt.exe < inp > out &
   ```

   The content of files created during execution can be compared with files in directory

   `Example/output_files/`

References

[1] E. V. Stefanovich, *Quantum Field Theory without Infinities*, Ann. Phys. (NY) **292** (2001), 139-156.

[2] E. V. Stefanovich, *Quantum Field Theory without Infinities II. A Simple Model*, unpublished; see

`http://www.geocities.com/meopemuk`

[3] S. Weinberg, "The Quantum Theory of Fields," Vol. 1., University Press, Cambridge, 1995.

[4] W. Magnus, *On the Exponential Solution of Differential Equations for a Linear Operator*, Commun. Pure Appl. Math. **7** (1954), 649; P. Pechukas and J. C. Light, *On the Exponential Form of Time-Displacement Operators in Quantum Mechanics*, J. Chem. Phys. **44** (1966), 3897.