

Current status of carlomat_3.0, an automatic tool for low energetic electron-positron annihilation into hadrons

Karol Kołodziej

Institute of Physics
University of Silesia
Katowice

17th Radio Montecarlo Working Group meeting

Frascati, 20 – 21 April, 2015

Outline

The most promising theoretical frameworks for the description of $e^+e^- \rightarrow \text{hadrons}$ in the energy range below the J/ψ threshold are the Resonance Chiral Theory or Hidden Local Symmetry model which, among others, involve

- the photon–vector meson mixing
- a number of vertices of rather complicated Lorentz tensor structure that is not present in the Standard Model or scalar QED.

Already at low energies, the hadronic final states may consist of several particles, such as pions, kaons, or nucleons which can be accompanied by one or more photons, or light fermion pairs such as e^+e^- , or $\mu^+\mu^-$.

Outline

The most promising theoretical frameworks for the description of $e^+e^- \rightarrow \text{hadrons}$ in the energy range below the J/ψ threshold are the Resonance Chiral Theory or Hidden Local Symmetry model which, among others, involve

- the photon–vector meson mixing
- a number of vertices of rather complicated Lorentz tensor structure that is not present in the Standard Model or scalar QED.

Already at low energies, the hadronic final states may consist of several particles, such as pions, kaons, or nucleons which can be accompanied by one or more photons, or light fermion pairs such as e^+e^- , or $\mu^+\mu^-$.

The number of Feynman diagrams of such multiparticle reactions grows substantially with increasing numbers of interaction vertices and mixing terms of the effective models.

⇒ It is highly desirable to automatize the calculations.

Outline

The most promising theoretical frameworks for the description of $e^+e^- \rightarrow \text{hadrons}$ in the energy range below the J/ψ threshold are the Resonance Chiral Theory or Hidden Local Symmetry model which, among others, involve

- the photon–vector meson mixing
- a number of vertices of rather complicated Lorentz tensor structure that is not present in the Standard Model or scalar QED.

Already at low energies, the hadronic final states may consist of several particles, such as pions, kaons, or nucleons which can be accompanied by one or more photons, or light fermion pairs such as e^+e^- , or $\mu^+\mu^-$.

The number of Feynman diagrams of such multiparticle reactions grows substantially with increasing numbers of interaction vertices and mixing terms of the effective models.

⇒ It is highly desirable to automatize the calculations.

Program `carlomat` has been substantially modified in order to

- incorporate a photon–vector meson mixing,
- include the Feynman interaction vertices of new tensor structures,
- to compute the helicity amplitudes involving the mixing terms and new interaction vertices, like those predicted by the $R\chi T$ or HLS model and the effective Lagrangian of the EM interaction of nucleons,
- introduce new options to enable a better control over the effective models implemented.

A new version of `carlomat`, labelled 3.0, allows to generate automatically the Monte Carlo programs dedicated to the description of processes $e^+e^- \rightarrow$ hadrons at low centre of mass energies.

Program `carlomat` has been substantially modified in order to

- incorporate a photon–vector meson mixing,
- include the Feynman interaction vertices of new tensor structures,
- to compute the helicity amplitudes involving the mixing terms and new interaction vertices, like those predicted by the $R\chi T$ or HLS model and the effective Lagrangian of the EM interaction of nucleons,
- introduce new options to enable a better control over the effective models implemented.

A new version of `carlomat`, labelled 3.0, allows to generate automatically the Monte Carlo programs dedicated to the description of processes $e^+e^- \rightarrow$ hadrons at low centre of mass energies.

q^2 -dependent mixing terms

In spite of being conceptually quite simple, the implementation of particle mixing required substantial changes in the code generation part of the program.

$$A^\mu(q) \quad V^\nu(q) \quad \equiv \quad -ef_{AV}(q^2) g^{\mu\nu},$$

with $V = \rho^0, \omega, \phi, \rho_1 = \rho(1450), \rho_2 = \rho(1700)$.

The mixing term introduces an extra power of $e \Rightarrow$ it should be included only once in a Feynman diagram.

q^2 -dependent mixing terms

In spite of being conceptually quite simple, the implementation of particle mixing required substantial changes in the code generation part of the program.

$$\begin{array}{c} A^\mu(q) \\ \text{wavy line} \end{array} \quad \begin{array}{c} V^\nu(q) \\ \text{double line} \end{array} \quad \equiv \quad -ef_{AV}(q^2) g^{\mu\nu},$$

with $V = \rho^0, \omega, \phi, \rho_1 = \rho(1450), \rho_2 = \rho(1700)$.

The mixing term introduces an extra power of $e \Rightarrow$ it should be included only once in a Feynman diagram.

To give a better control over the mixing contributions to a given process, subroutines `bbkk` and `bbmd` are equipped with the option:

`iwgt=0/1,2,...` if the additional complex factor c_1, c_2, \dots
is not/is to be included in $f_{AV}(q^2)$

q^2 -dependent mixing terms

The actual names for that option in `carlocom` are:

`imrho`, `imome`, `imphi`, `imrh1`, `imrh2` for $\rho^0, \omega, \phi, \rho_1, \rho_2$ meson, respectively.

The complex factor c_j , $j = 1, 2, \dots$ is given by

$$c_j = w_j e^{i\phi_j} f_j(q^2),$$

where w_j is a positive weight, ϕ_j is an angle in degrees, which should be both specified for each possible particle mixing term in the main program for the MC computation `carlocom`, and $f_j(q^2)$ is a possible four momentum transfer dependence that is defined in subroutine `weightfactor`.

q^2 -dependent mixing terms

The actual names for that option in `carlocom` are:

`imrho`, `imome`, `imphi`, `imrh1`, `imrh2` for $\rho^0, \omega, \phi, \rho_1, \rho_2$ meson, respectively.

The complex factor c_j , $j = 1, 2, \dots$ is given by

$$c_j = w_j e^{i\phi_j} f_j(q^2),$$

where w_j is a positive weight, ϕ_j is an angle in degrees, which should be both specified for each possible particle mixing term in the main program for the MC computation `carlocom`, and $f_j(q^2)$ is a possible four momentum transfer dependence that is defined in subroutine `weightfactor`.

Only three simple dependencies corresponding to `iwgt=1,2,3` are currently defined in `weightfactor`, but the user can easily add more options by implementing new

`else if (iwgt == ...) then` conditions.

q^2 -dependent mixing terms

The actual names for that option in `carlocom` are:

`imrho`, `imome`, `imphi`, `imrh1`, `imrh2` for $\rho^0, \omega, \phi, \rho_1, \rho_2$ meson, respectively.

The complex factor c_j , $j = 1, 2, \dots$ is given by

$$c_j = w_j e^{i\phi_j} f_j(q^2),$$

where w_j is a positive weight, ϕ_j is an angle in degrees, which should be both specified for each possible particle mixing term in the main program for the MC computation `carlocom`, and $f_j(q^2)$ is a possible four momentum transfer dependence that is defined in subroutine `weightfactor`.

Only three simple dependencies corresponding to `iwgt=1,2,3` are currently defined in `weightfactor`, but the user can easily add more options by implementing new

`else if (iwgt == ...) then` conditions.

q^2 -dependent interaction vertices

Triple vertices of the form similar to those of sQED:

$$\begin{aligned} & \begin{array}{c} P(p_1) \\ \diagup \\ A^\mu(q) \text{ (red wavy)} \\ \diagdown \\ \bar{P}(p_2) \end{array} \equiv ief_{APP}(q^2)(p_1 - p_2) \\ & \begin{array}{c} P(p_1) \\ \diagup \\ V^\mu(q) \text{ (red double)} \\ \diagdown \\ \bar{P}(p_2) \end{array} \equiv if_{VPP}(q^2)(p_1 - p_2)^\mu \end{aligned}$$

where $V = \rho^0, \omega, \phi$ and $P = \pi^+, K^+, K^0$.

q^2 -dependent interaction vertices

Triple vertices of a more complicated tensor form:

$$\begin{array}{c}
 A^\mu(p_1) \\
 \diagup \\
 \pi^0(q) \text{ --- } \bullet \\
 \diagdown \\
 A^\nu(p_2)
 \end{array}
 \equiv e^2 f_{\pi AA}(q^2) \varepsilon^{\mu\nu\alpha\beta} p_{1\alpha} p_{2\beta}$$

$$\begin{array}{c}
 A^\mu(p_1) \\
 \diagup \\
 \pi^0(q) \text{ --- } \bullet \\
 \diagdown \\
 V^\nu(p_2)
 \end{array}
 \equiv i e f_{\pi AV}(q^2) \varepsilon^{\mu\nu\alpha\beta} p_{1\alpha} p_{2\beta}$$

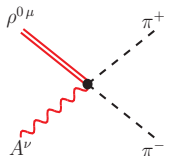
$$\begin{array}{c}
 A^\mu(p_1) \\
 \diagup \\
 \pi^\mp(q) \text{ --- } \bullet \\
 \diagdown \\
 \rho^{\pm\nu}(p_2)
 \end{array}
 \equiv i e f_{\pi^\mp A \rho^\pm}(q^2) \varepsilon^{\mu\nu\alpha\beta} p_{1\alpha} p_{2\beta}$$

$$\begin{array}{c}
 \omega^\mu(p_1) \\
 \diagup \\
 P(q) \text{ --- } \bullet \\
 \diagdown \\
 V^\nu(p_2)
 \end{array}
 \equiv f_{P\omega V}(q^2) \varepsilon^{\mu\nu\alpha\beta} p_{1\alpha} p_{2\beta}$$

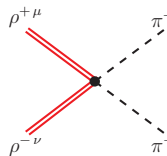
where, in the top right corner, $V = \rho^0, \omega$, and in the bottom right corner $P = \pi^0$ and $V = \rho^0$ or $P = \pi^\mp$ and $V = \rho^\pm$.

q^2 -dependent interaction vertices

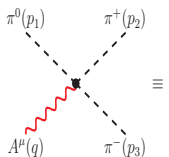
Quartic vertices of the HLS model:



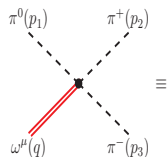
$$\equiv 2ief_{A\rho\pi\pi}(q^2)g^{\mu\nu}$$



$$\equiv 2if_{\rho\rho\pi\pi}(q^2)g^{\mu\nu}$$



$$\equiv -ef_{A\pi\pi\pi}(q^2)\varepsilon^{\mu\nu\alpha\beta}p_{1\nu}p_{2\alpha}p_{3\beta}$$



$$\equiv -ef_{\omega\pi\pi\pi}(q^2)\varepsilon^{\mu\nu\alpha\beta}p_{1\nu}p_{2\alpha}p_{3\beta}$$

The vertices in the first row have the same tensor form as the quartic vertex of the sQED or the quartic vertices of the Nambu-Goldstone boson – gauge boson interaction of the SM, which were implemented already in the first version of carlomat.

New program options

All subroutines that are used to compute the building blocks or the complete helicity amplitudes of the Feynman diagrams containing vertices of the HLS model have been supplied with **the running coupling option**.

The options are to be specified in subroutine `couplsm`, where they are defined below the assignment for each particular coupling.

```
icoupl_name=0/1,2,... if the fixed/running coupling
                        is to be used in the computation,
where choices 1,2,... corresponding to different running
couplings  $f_{...}(q^2)$  should be added by the user as extra
else if (ig == ...) then blocks in subroutine runcoupl.
```


New program options

All subroutines that are used to compute the building blocks or the complete helicity amplitudes of the Feynman diagrams containing vertices of the HLS model have been supplied with [the running coupling option](#).

The options are to be specified in subroutine `couplsm`, where they are defined below the assignment for each particular coupling.

`icoupl_name=0/1,2,...` if the [fixed/running coupling](#) is to be used in the computation, where choices `1,2,...` corresponding to different [running couplings](#) $f_{...}(q^2)$ should be added by the user as extra `else if (ig == ...) then` blocks in subroutine `runcoupl`.

The four momentum transfer q is determined automatically from the four momentum conservation in the corresponding interaction vertex at the stage of code generation.

New program options

All subroutines that are used to compute the building blocks or the complete helicity amplitudes of the Feynman diagrams containing vertices of the HLS model have been supplied with **the running coupling option**.

The options are to be specified in subroutine **couplsm**, where they are defined below the assignment for each particular coupling.

icoupl_name=0/1,2,... if the **fixed/running coupling** is to be used in the computation, where choices **1,2,...** corresponding to different **running couplings $f_{...}(q^2)$** should be added by the user as extra **else if (ig == ...) then** blocks in subroutine **runcoupl**.

The four momentum transfer q is determined automatically from the four momentum conservation in the corresponding interaction vertex at the stage of code generation.

New program options

Many couplings of the $R\chi T$ or HLS model are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data. If there are no hints as to the form of the running couplings $f_{\dots}(q^2)$ then it is recommended to set the corresponding running coupling option to 0, which means that the fixed coupling is to be used in the computation.

New program options

Many couplings of the $R\chi T$ or HLS model are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data. If there are no hints as to the form of the running couplings $f_{\dots}(q^2)$ then it is recommended to set the corresponding running coupling option to 0, which means that the fixed coupling is to be used in the computation.

The user can also modify any of the fixed couplings by changing the corresponding assignments in `coup1sm`, where the couplings are defined in terms of the physical parameters of module `inprms`.

New program options

Many couplings of the $R\chi T$ or HLS model are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data. If there are no hints as to the form of the running couplings $f_{\dots}(q^2)$ then it is recommended to set the corresponding running coupling option to 0, which means that the fixed coupling is to be used in the computation.

The user can also modify any of the fixed couplings by changing the corresponding assignments in `coup1sm`, where the couplings are defined in terms of the physical parameters of module `inprms`.

The main part of the MC computation program `carlocom` contains also a few flags: `iarho`, `iaome`, `iaphi`, `iarho1` and `iarho2` that allow to switch off and on the photon mixing with ρ, ω, ϕ , ρ_1 and ρ_2 vector mesons without a need of running the code generation program anew, provided that the corresponding mixing terms were included in a file `vertices.dat` when the MC code was generated.

New program options

Many couplings of the $R\chi T$ or HLS model are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data. If there are no hints as to the form of the running couplings $f_{\dots}(q^2)$ then it is recommended to set the corresponding running coupling option to 0, which means that the fixed coupling is to be used in the computation.

The user can also modify any of the fixed couplings by changing the corresponding assignments in `couplsm`, where the couplings are defined in terms of the physical parameters of module `inprms`. The main part of the MC computation program `carlocom` contains also a few flags: `iarho`, `iaome`, `iaphi`, `iarho1` and `iarho2` that allow to switch off and on the photon mixing with ρ, ω, ϕ , ρ_1 and ρ_2 vector mesons without a need of running the code generation program anew, provided that the corresponding mixing terms were included in a file `vertices.dat` when the MC code was generated.

New program options

The subroutines for computation of the four vectors representing vector mesons have been in addition supplied with the **running width option** `iwidth_name`, i.e. `igmrh`, `igmom`, `igmph`, `igmr1`, `igmr2` for the running width of $\rho^0, \omega, \phi, \rho_1, \rho_2$, respectively.

`iwidth_name=0/1,2,3` if the fixed/running width of the vector particle should be used,

where choices `1,2,3` refer to different running width options in subroutine `runwidth` which again can easily be extended by the user. The options are controlled from `carlocom`, the main part of the MC computation program.

New program options

An important new option in the program that allows to test the EM gauge invariance for processes with one or more external photons is `igauge` in `carlocom.f`:

`igauge=1,2,.../else` if the gauge invariance `is/is not` to be tested,

where `1,2,...` is the number of a photon, counting from left to right, whose polarization four vector is replaced with its four momentum.

To illustrate how this option can be used in practice, consider the following radiative processes:

$$\begin{aligned}e^+e^- &\rightarrow \pi^+\pi^-\mu^+\mu^-\gamma, \\e^+e^- &\rightarrow \pi^+\pi^-\pi^+\pi^-\gamma.\end{aligned}$$

Taking into account the Feynman rules of SM, without the Higgs couplings to electrons and muons, sQED, the $\gamma-\rho^0$ mixing and the vertices: $\gamma\pi^+\pi^-$, $\rho^0\pi^+\pi^-$, $\pi^0\gamma\gamma$, $\pi^0\gamma\rho^0$, $\gamma\rho^0\pi^+\pi^-$ and $\gamma\pi^0\pi^+\pi^-$
 \Rightarrow 209 and 774 Feynman diagrams.

If, in addition, the vertices $\pi^\mp\gamma\rho^\pm$ are included \Rightarrow 231 and 968 Feynman diagrams.

To illustrate how this option can be used in practice, consider the following radiative processes:

$$e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma,$$

$$e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma.$$

Taking into account the Feynman rules of SM, without the Higgs couplings to electrons and muons, sQED, the $\gamma-\rho^0$ mixing and the vertices: $\gamma\pi^+\pi^-$, $\rho^0\pi^+\pi^-$, $\pi^0\gamma\gamma$, $\pi^0\gamma\rho^0$, $\gamma\rho^0\pi^+\pi^-$ and $\gamma\pi^0\pi^+\pi^-$

\Rightarrow 209 and 774 Feynman diagrams.

If, in addition, the vertices $\pi^\mp\gamma\rho^\pm$ are included \Rightarrow 231 and 968 Feynman diagrams.

EM gauge invariance tests

The cross sections in pb at $\sqrt{s} = 1$ GeV **without** and **with** contribution from the $\pi^\mp \gamma \rho^\pm$ vertices with the following cuts:

$$\theta_{\gamma l} > 5^\circ, \quad \theta_{\gamma \pi} > 5^\circ, \quad E_\gamma > 10 \text{ MeV}.$$

igauge	$\sigma(e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma)$		$\sigma(e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma)$	
0	11.86(5)	11.83(5)	0.0590(2)	0.0586(2)
1	0.124(2)e-30	0.441(1)e-10	0.636(9)e-33	0.973(1)e-9

The numbers in parentheses show the MC uncertainty of the last decimal.

The inclusion of the $\pi^\mp \gamma \rho^\pm$ vertices spoils the EM gauge invariance. The effect is not practically relevant, but in some phase space regions it may become sizable.

EM gauge invariance tests

The cross sections in pb at $\sqrt{s} = 1$ GeV **without** and **with** contribution from the $\pi^\mp \gamma \rho^\pm$ vertices with the following cuts:

$$\theta_{\gamma l} > 5^\circ, \quad \theta_{\gamma \pi} > 5^\circ, \quad E_\gamma > 10 \text{ MeV}.$$

igauge	$\sigma(e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma)$		$\sigma(e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma)$	
0	11.86(5)	11.83(5)	0.0590(2)	0.0586(2)
1	0.124(2)e-30	0.441(1)e-10	0.636(9)e-33	0.973(1)e-9

The numbers in parentheses show the MC uncertainty of the last decimal.

The inclusion of the $\pi^\mp \gamma \rho^\pm$ vertices spoils the EM gauge invariance. The effect is not practically relevant, but in some phase space regions it may become sizable.

Running time:

EM gauge invariance tests

The cross sections in pb at $\sqrt{s} = 1$ GeV **without** and **with** contribution from the $\pi^\mp \gamma \rho^\pm$ vertices with the following cuts:

$$\theta_{\gamma l} > 5^\circ, \quad \theta_{\gamma\pi} > 5^\circ, \quad E_\gamma > 10 \text{ MeV}.$$

igauge	$\sigma(e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma)$		$\sigma(e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma)$	
0	11.86(5)	11.83(5)	0.0590(2)	0.0586(2)
1	0.124(2)e-30	0.441(1)e-10	0.636(9)e-33	0.973(1)e-9

The numbers in parentheses show the MC uncertainty of the last decimal.

The inclusion of the $\pi^\mp \gamma \rho^\pm$ vertices spoils the EM gauge invariance. The effect is not practically relevant, but in some phase space regions it may become sizable.

Running time: the code generation for both processes takes a fraction of a second time.

EM gauge invariance tests

The cross sections in pb at $\sqrt{s} = 1$ GeV **without** and **with** contribution from the $\pi^\mp \gamma \pi^\pm$ vertices with the following cuts:

$$\theta_{\gamma l} > 5^\circ, \quad \theta_{\gamma\pi} > 5^\circ, \quad E_\gamma > 10 \text{ MeV}.$$

igauge	$\sigma(e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma)$		$\sigma(e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma)$	
0	11.86(5)	11.83(5)	0.0590(2)	0.0586(2)
1	0.124(2)e-30	0.441(1)e-10	0.636(9)e-33	0.973(1)e-9

The numbers in parentheses show the MC uncertainty of the last decimal.

The inclusion of the $\pi^\mp \gamma \pi^\pm$ vertices spoils the EM gauge invariance. The effect is not practically relevant, but in some phase space regions it may become sizable.

Running time: the code generation for both processes takes a fraction of a second time. The MC computation with $10 \times 200\,000$ calls takes 142s and 43s time for $\pi^+\pi^-\pi^+\pi^-\gamma$ and $\pi^+\pi^-\mu^+\mu^-\gamma$ final states on processor Intel^R CoreTM i5-4200M CPU @ 2.50 GHz with a 64 bit Intel Fortran compiler.

EM gauge invariance tests

The cross sections in pb at $\sqrt{s} = 1$ GeV **without** and **with** contribution from the $\pi^\mp \gamma \rho^\pm$ vertices with the following cuts:

$$\theta_{\gamma l} > 5^\circ, \quad \theta_{\gamma \pi} > 5^\circ, \quad E_\gamma > 10 \text{ MeV}.$$

igauge	$\sigma(e^+e^- \rightarrow \pi^+\pi^-\pi^+\pi^-\gamma)$		$\sigma(e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma)$	
0	11.86(5)	11.83(5)	0.0590(2)	0.0586(2)
1	0.124(2)e-30	0.441(1)e-10	0.636(9)e-33	0.973(1)e-9

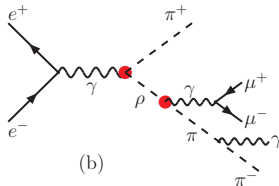
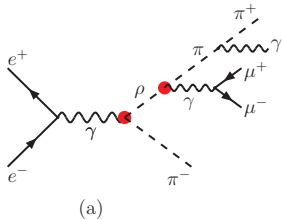
The numbers in parentheses show the MC uncertainty of the last decimal.

The inclusion of the $\pi^\mp \gamma \rho^\pm$ vertices spoils the EM gauge invariance. The effect is not practically relevant, but in some phase space regions it may become sizable.

Running time: the code generation for both processes takes a fraction of a second time. The MC computation with $10 \times 200\,000$ calls takes 142s and 43s time for $\pi^+\pi^-\pi^+\pi^-\gamma$ and $\pi^+\pi^-\mu^+\mu^-\gamma$ final states on processor Intel^R CoreTM i5-4200M CPU @ 2.50 GHz with a 64 bit Intel Fortran compiler.

EM gauge invariance tests

The Feynman diagrams of $e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma$ that spoil the EM gauge invariance, where **blobs** indicate the vertices $\pi^\mp \gamma \rho^\pm$:



Amplitudes:

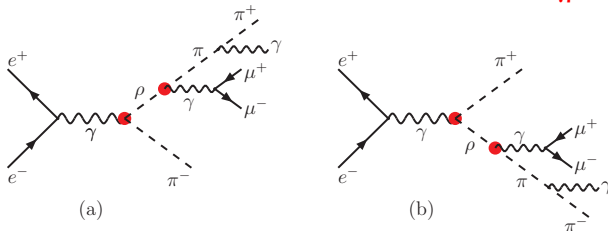
$$M_a = g^2 \varepsilon_{12\nu} \varepsilon^{\nu\mu\alpha\beta} p_{12\alpha} (-q_\beta) \frac{-g_{\mu\rho} + \frac{q_\mu q_\rho}{M^2}}{q^2 - M^2} \varepsilon_{56\sigma} \varepsilon^{\sigma\rho\gamma\delta} (-p_{56\gamma}) q_\delta s_{37},$$

$$M_b = g^2 \varepsilon_{12\nu} \varepsilon^{\nu\mu\alpha\beta} p_{12\alpha} (-r_\beta) \frac{-g_{\mu\rho} + \frac{r_\mu r_\rho}{M^2}}{r^2 - M^2} \varepsilon_{56\sigma} \varepsilon^{\sigma\rho\gamma\delta} (-p_{56\gamma}) r_\delta s_{47}$$

where $g = ef_{\pi^\mp A\rho^\pm}(q^2) = ef_{\pi^\mp A\rho^\pm}(r^2) = \text{const.}$, $M^2 = m_\rho^2 - im_\rho\Gamma_\rho$.

EM gauge invariance tests

The Feynman diagrams of $e^+e^- \rightarrow \pi^+\pi^-\mu^+\mu^-\gamma$ that spoil the EM gauge invariance, where **blobs** indicate the vertices $\pi^\mp \gamma \rho^\pm$:



Amplitudes:

$$M_a = g^2 \epsilon_{12\nu} \epsilon^{\nu\mu\alpha\beta} p_{12\alpha} (-q_\beta) \frac{-g_{\mu\rho} + \frac{q_\mu q_\rho}{M^2}}{q^2 - M^2} \epsilon_{56\sigma} \epsilon^{\sigma\rho\gamma\delta} (-p_{56\gamma}) q_\delta s_{37},$$

$$M_b = g^2 \epsilon_{12\nu} \epsilon^{\nu\mu\alpha\beta} p_{12\alpha} (-r_\beta) \frac{-g_{\mu\rho} + \frac{r_\mu r_\rho}{M^2}}{r^2 - M^2} \epsilon_{56\sigma} \epsilon^{\sigma\rho\gamma\delta} (-p_{56\gamma}) r_\delta s_{47}$$

where $g = ef_{\pi^\mp A\rho^\pm}(q^2) = ef_{\pi^\mp A\rho^\pm}(r^2) = \text{const.}$, $M^2 = m_\rho^2 - im_\rho \Gamma_\rho$.

EM gauge invariance tests

The scalars s_{37} and s_{47} representing the $\pi^+\pi^-\gamma$ and $\pi^-\pi^+\gamma$ vertex multiplied with the adjacent pion propagator, take the following form:

$$\begin{aligned}s_{37} &= e \frac{(p_3 + p_7 - (-p_3))^\mu \epsilon_\mu^*(p_7)}{(p_3 + p_7)^2 - m_\pi^2} \Big|_{\epsilon(p_7) \rightarrow p_7} = e \frac{2p_3 \cdot p_7}{2p_3 \cdot p_7} = e, \\s_{47} &= e \frac{(-p_4 - (p_4 + p_7))^\mu \epsilon_\mu^*(p_7)}{(p_4 + p_7)^2 - m_\pi^2} \Big|_{\epsilon(p_7) \rightarrow p_7} = -e \frac{2p_4 \cdot p_7}{2p_4 \cdot p_7} = -e.\end{aligned}$$

Hence

$$\begin{aligned}M_a &= \frac{eg^2}{q^2 - M^2} \epsilon^{\mu\nu\alpha\beta} \epsilon_{\mu\sigma\gamma\delta} \epsilon_{12\nu} p_{12\alpha} p_{4\beta} \epsilon_{56}^\sigma p_{56}^\gamma (p_3 + p_7)^\delta, \\M_b &= -\frac{eg^2}{r^2 - M^2} \epsilon^{\mu\nu\alpha\beta} \epsilon_{\mu\sigma\gamma\delta} \epsilon_{12\nu} p_{12\alpha} p_{3\beta} \epsilon_{56}^\sigma p_{56}^\gamma (p_4 + p_7)^\delta.\end{aligned}$$

\Rightarrow M_a and M_b neither vanish separately nor cancel each other.

Preparation for running and program usage

carlomat_3.0 is distributed as a single tar.gz archive
carlomat_3.0.tgz which can be downloaded from:
<http://kk.us.edu.pl/carlomat.html>.

When untared with a command

```
tar -xzvf carlomat_3.0.tgz
```

it will create directory carlomat_3.0 with sub directories:

code_generation, mc_computation, carlolib, test_output
and test_output0.

Preparation for running and program usage

carlomat_3.0 is distributed as a single tar.gz archive
carlomat_3.0.tgz which can be downloaded from:

<http://kk.us.edu.pl/carlomat.html>.

When untared with a command

```
tar -xzvf carlomat_3.0.tgz
```

it will create directory carlomat_3.0 with sub directories:

`code_generation`, `mc_computation`, `carllib`, `test_output`
and `test_output0`.

Preparation for running requires the following steps:

- Choose a Fortran 90 compiler in makefile's of `code_generation` and `mc_computation` and compile all the routines of `carllib` with the same compiler as that chosen in `mc_computation`;
- Specify the process and required options in `carlomat.f` and execute `make` code from the command line in `code_generation`;

Preparation for running and program usage

carlomat_3.0 is distributed as a single tar.gz archive
carlomat_3.0.tgz which can be downloaded from:

<http://kk.us.edu.pl/carlomat.html>.

When untared with a command

```
tar -xzvf carlomat_3.0.tgz
```

it will create directory carlomat_3.0 with sub directories:

`code_generation`, `mc_computation`, `carllib`, `test_output`
and `test_output0`.

Preparation for running requires the following steps:

- Choose a Fortran 90 compiler in makefile's of `code_generation` and `mc_computation` and compile all the routines of `carllib` with the same compiler as that chosen in `mc_computation`;
- Specify the process and required options in `carlomat.f` and execute `make` code from the command line in `code_generation`;

- Go to `mc_computation`, choose the centre of mass energy and required options in `carlocom.f` and execute `make mc` in the command line.

Whenever the Fortran compiler is changed, or a compiled program is transferred to another computer with a different processor then all the object and module files should be deleted by executing the commands:

```
rm *.o
```

```
rm *.mod
```

and the necessary steps of those listed above should be repeated.

- Go to `mc_computation`, choose the centre of mass energy and required options in `carlocom.f` and execute `make mc` in the command line.

Whenever the Fortran compiler is changed, or a compiled program is transferred to another computer with a different processor then all the object and module files should be deleted by executing the commands:

```
rm *.o
```

```
rm *.mod
```

and the necessary steps of those listed above should be repeated.

The basic output of the MC run is written to file `tot_name`, where `name` is created automatically if the assignment

`prcsnm='auto'`

in `carlomat.f` is not changed to arbitrary user's defined name.

When the run is finished all output files, except for `test`, that may contain information relevant in case of unexpected program stop, are moved to directory `test_output`.

Program output

The basic output of the MC run is written to file `tot_name`, where `name` is created automatically if the assignment `prcsnm='auto'` in `carlomat.f` is not changed to arbitrary user's defined name. When the run is finished all output files, except for `test`, that may contain information relevant in case of unexpected program stop, are moved to directory `test_output`.

The output files for processes

$$\begin{aligned} e^+e^- &\rightarrow \pi^+\pi^-\mu^+\mu^-\gamma, \\ e^+e^- &\rightarrow \pi^+\pi^-\pi^+\pi^-\gamma. \end{aligned}$$

with the preselected parameters and options should reproduce those delivered in directory `test_output0`.

Program output

The basic output of the MC run is written to file `tot_name`, where name is created automatically if the assignment `prcsnm='auto'` in `carlomat.f` is not changed to arbitrary user's defined name. When the run is finished all output files, except for `test`, that may contain information relevant in case of unexpected program stop, are moved to directory `test_output`. The output files for processes

$$\begin{aligned}e^+e^- &\rightarrow \pi^+\pi^-\mu^+\mu^-\gamma, \\e^+e^- &\rightarrow \pi^+\pi^-\pi^+\pi^-\gamma.\end{aligned}$$

with the preselected parameters and options should reproduce those delivered in directory `test_output0`.

If the differential cross sections/distributions are required then set `idis=1` in `carlocom.f`.

The number of distributions to be calculated must be specified in `distrib.f` and their parameters should be defined in `calcdis.f`. The output will be stored in data files `db#_name` and `dl#_name` which can be plotted with boxes and lines, respectively, with `gnuplot`.

If the differential cross sections/distributions are required then set `idis=1`

in `carlocom.f`.

The number of distributions to be calculated must be specified in `distribs.f` and their parameters should be defined in `calcdis.f`. The output will be stored in data files `db#_name` and `dl#_name` which can be plotted with boxes and lines, respectively, with `gnuplot`.

Restrictions

As in previous versions of the program the number of particles in the final state is **limited to 10** which exceeds typical numbers of particles observed in the exclusive low energy e^+e^- -annihilation processes.

However, in the presence of photon–vector meson mixing, the Feynman diagrams proliferate, for example, with currently implemented Feynman rules, there are 90672 diagrams of $e^+e^- \rightarrow 3(\pi^+\pi^-)$.

⇒ The compilation time of generated code may become very long already for processes with smaller number of the final state particles.

Restrictions

As in previous versions of the program the number of particles in the final state is **limited to 10** which exceeds typical numbers of particles observed in the exclusive low energy e^+e^- -annihilation processes.

However, **in the presence of photon–vector meson mixing, the Feynman diagrams proliferate, for example, with currently implemented Feynman rules, there are 90672 diagrams of $e^+e^- \rightarrow 3(\pi^+\pi^-)$.**

⇒ The compilation time of generated code may become very long already for processes with smaller number of the final state particles.

Many couplings of the effective models are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data.

Restrictions

As in previous versions of the program the number of particles in the final state is **limited to 10** which exceeds typical numbers of particles observed in the exclusive low energy e^+e^- -annihilation processes.

However, **in the presence of photon–vector meson mixing, the Feynman diagrams proliferate, for example, with currently implemented Feynman rules, there are 90672 diagrams of $e^+e^- \rightarrow 3(\pi^+\pi^-)$.**

⇒ The compilation time of generated code may become very long already for processes with smaller number of the final state particles.

Many couplings of the effective models are not known well enough and must be adjusted in consecutive runs of the program in order to obtain satisfactory description of the experimental data.

Thank you for your attention