# PHACT : Helicity amplitudes for present and future colliders

Alessandro Ballestrero
INFN and Dip. Fisica Teorica
via Giuria 1 - 10125 Torino
Italy

**Abstract**

Helicity amplitudes calculations with the program PHACT are explained. Some examples of their application in WPHACT and SIXPHACT MC's are given.

## 1 Introduction

The possibility of exploring higher and higher energies at present and future accelerators, entails the necessity of predicting and computing with high precision more and more complicated processes, with more external particles and more Feynman diagrams to be evaluated. This has lead to progressively abandon the old trace methods for perturbative computations and to use instead the helicity amplitudes ones. [1]

In general these can be divided in three categories, depending on how the various fermionic lines appearing in a diagram are computed: by Trace evaluation, Spinor techniques or Matrix multiplication. We will not review here explicitly the implementations of the three methods, but we refer to [3], [4], [5] respectively as typical examples of them.

The program PHACT [6] implements an helicity amplitude method[2] which belongs to both the second and third type: one makes use of a generalization of BDKS spinors[4], which induce naturally the evaluation by multiplication of so called $\tau$ matrices. One of the advantages of the method is that in such a way the matrices for spinor propagators become trivial.

We will review some of its characteristics, give a short account of PHACT and describe some of its applications.

## 2 Method

The method of ref. [2] turns out to be easy to program, particularly fast and very convenient for computations involving massive fermions.

It is based on the following steps:

- Generalize spinors so that $u(p,\lambda)$ and $\bar{u}(p,\lambda)$ are defined for any timelike, lightlike and spacelike vector $p$ and in all cases the completeness relation holds:

$$\sum_{\lambda} \frac{u(p,\lambda)\bar{u}(p,\lambda) - v(p,\lambda)\bar{v}(p,\lambda)}{2m} = 1 \tag{1}$$

- Make use of the above completeness relation to diagonalize $\not{p}$ in fermion propagators

- Use explicitly generalized BDKS spinors for which the following relations hold for all $p^2$ $(m = \sqrt{p^2})$.

$$u(p,\lambda) = \frac{\not{p}+m}{\sqrt{2\,p\cdot k_0}}\,w(k_0,-\lambda) \qquad v(p,\lambda) = \frac{\not{p}-m}{\sqrt{2\,p\cdot k_0}}\,w(k_0,-\lambda) \tag{2}$$

$$\bar{u}(p,\lambda) = \bar{w}(k_0,-\lambda)\,\frac{\not{p}+m}{\sqrt{2\,p\cdot k_0}} \qquad \bar{v}(p,\lambda) = \bar{w}(k_0,-\lambda)\,\frac{\not{p}-m}{\sqrt{2\,p\cdot k_0}} \tag{3}$$

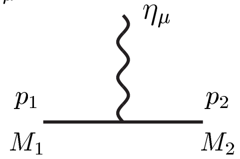
with

$$w(k_0,\lambda)\bar{w}(k_0,\lambda) = \frac{1+\lambda\gamma_5}{2}\not{k}_0 \qquad w(k_0,\lambda) = \lambda\not{k}_1 w(k_0,-\lambda) \tag{4}$$

$$k_0^2 = 0 \qquad k_1^2 = -1 \qquad k_1\cdot k_0 = 0$$

As a consequence, we have that

- Every insertion of a vector (or scalar) $\eta_\mu$ in a fermion line

$M_1 = \sqrt{p_1^2}$  $M_2 = \sqrt{p_2^2}$


$$\overline{U}(p_1)\not\eta \left[ c_r \left( \frac{1+\gamma_5}{2} \right) + c_l \left( \frac{1-\gamma_5}{2} \right) \right] U(p_2) \tag{5}$$

turns out to be of the form (also for scalar coupling):

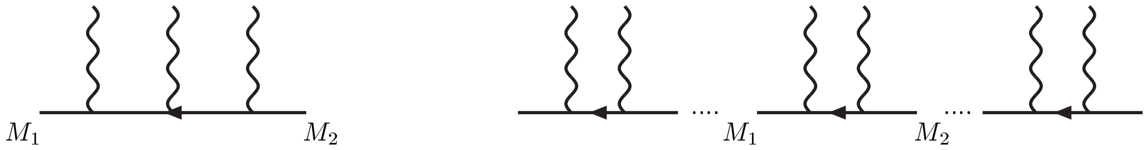
$$A + B\, M_1 + C\, M_2 + D\, M_1 M_2 \tag{6}$$

with $A$, $B$, $C$, $D$ 2x2 matrices (+ and - correspond to the values of the helicity $\lambda$ indices) :

$$\begin{pmatrix} A_{++} & A_{+-} \\ A_{-+} & A_{++} \end{pmatrix} \tag{7}$$

Here $\eta_\mu$ represents a polarization vector or a whole subdiagrams with appropriate indices.

- Every fermion line or piece of it



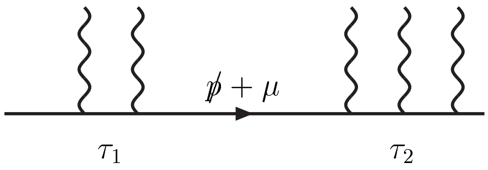

is of the same form:

$$A + B\, M_1 + C\, M_2 + D\, M_1 M_2 \tag{8}$$

It can be represented by a $\tau$ matrix:

$$\tau = \begin{pmatrix} A & C \\ B & D \end{pmatrix} \tag{9}$$

- Composition of two pieces of fermion line is trivial:

$\tau =$ 


$$\tau = \tau_1 \otimes \tau_2$$

$$\begin{pmatrix} A & C \\ B & D \end{pmatrix} = \begin{pmatrix} A_1 & C_1 \\ B_1 & D_1 \end{pmatrix} \begin{pmatrix} 1 & \mu \\ \mu & p^2 \end{pmatrix} \begin{pmatrix} A_2 & C_2 \\ B_2 & D_2 \end{pmatrix} \tag{10}$$

With the above results and the expressions of $A$, $B$, $C$, $D$[2] for a single vector or scalar insertion, any calculation can in principle be performed. This way of evaluating amplitudes has some advantages as it eliminates the combinatorics induced by expressing propagator momenta in terms of other 4-vectors and it allows a natural decomposition in subdiagrams. This in turn reduces the number of graphs to account for and easily avoids repeating computations of subgraphs common to different diagrams.
A typical example of modular computation with `PHACT` is described in [9]: for the process $e^+e^- \to \mu\nu u\bar{d}b\bar{b}$ which has 8 external fermions and 209 Feynman diagrams we have computed the amplitude by first evaluating the subdiagrams corresponding to the emission of 4 fermions from a virtual vector boson and to the emission of four fermion from a fermion line. With these subdiagrams we end up with only 18 diagrams.

## 3 PHACT

PHACT (**P**rogram for **H**elicity **A**mplitude **C**alculations with **T**au matrices)[6] is a program to write optimized fortran codes for helicity amplitude processes with the above described method. It is composed of many routines which write the code for the different parts: $\tau$ matrices for a single insertion, products $\tau \otimes \tau$ of matrices, vector boson polarizations, triple vertices, determinants, etc. In all cases one gives as arguments to the routines the names to be used for vectors, matrices, couplings, etc. and the code will be written with the appropriate variables.
The purpose is to generate a final code in the easiest way. Instead of generating all the fortran program with PHACT, one could also think of another strategy. At least in some simple cases one could in fact write a fortran program where one subroutine (say for instance the one to compute $\tau$) is called many times. However just in the case at hand the "polarization" vector $\eta_\mu$ can correspond to different subdiagrams, and as such it can carry different number of indices. This is not naturally and easily implemented in a repeated call to a fortran routine, while with PHACT one can produce the code for each different call with its own indices and the resulting program will be much less time consuming.
Moreover, writing a code with the help of PHACT is greatly simplified by some useful features:

- normal indented fortran lines can be alternated to specific calls to PHACT in the input program, and they will be recognized as such and left unchanged. So after the input has been processed by PHACT, the resulting final fortran code will be composed by the parts written directly in the input and by those generated automatically.
- when one wants to repeat a piece of input just changing some names or indices, one can use a `dotab` command. After the command, in the first line are given the symbols to change and in the following ones the values they will assume at the first, second, ... repetition. `endtab` terminates the list and `enddo` terminates the part of input code to be repeated. `dotab` substitutions can be applied also to normal fortran lines in input and can be nested.
- for many PHACT routines the use of indices followed by a question mark ? will result in fortran do loops on those indices in the code generated by the routine.

The subroutines which can be called using PHACT are at present the following:

```
pol,pk0,quqd,p.q,eps,trip,mline
t,tsts,tst,tts,tt
t0,tl0,tr0,t10,tlt0,ttr0,tltr0
tw,tstw,twts,twt,ttw,twtw
tw0,twl0,twr0,tw10,tlt0_w,ttr0_w,tltr0_w
th,tsc,tsct,ttsc,tscts,tstsc,tsctw,twtsc,tsctsc
tot,tto,tots,tsto,totw,twto,toto,totsc,tscto
tcr,tcrtcr
```

By calling the ones in the first line and specifying their arguments, one obtains the fortran code for computing scalar products, determinants, polarization vectors, triple vertices etc. In the second line the routine `t` writes the fortran code for the $\tau$ matrix of one single insertion, and the others combine different $\tau$'s together. Similar routines in the third line serve for the case of massless fermion lines. In the two following lines are the corresponding routines for insertions of W's (left coupling only), while the remaining ones are for scalars, pseudoscalars and right couplings.
As a simple example of the use of these routines let's consider how one gets the code for the single insertion:

```
t        !subroutine T [qu,qd,v,abcd,cr,cl,den,nquqd,nsum]
p1
p2
v34(i,j).e
res(i?,j?).
coupr
```

```
coupl
0
0
0
```

After calling the routine `t`, one gives as arguments the names of the 4-momenta of the fermion line, the name of the "polarization vector" (which may correspond to a subdiagram) of the insertion with the corresponding indexes and the name of the couplings. The three last entries correspond to the possibility of dividing by a denominator, of performing the scalar product of the two 4-vectors and of summing the result with a previous one.
The output by `PHACT` corresponding to the above input is:

```
      do i=1,2
      do j=1,3
* T -- qu=p1,qd=p2,v=v34(i,j).e,a=res(i,j).a,b=res(i,j).b,c=res(i,j).c,d=re
* s(i,j).d,cr=coupr,cl=coupl,nsum=0
      eps_0=-v34(i,j).ek0*(p1(2)*p2(3)-p2(2)*p1(3))+p1k0*(v34(i,
     & j).e(2)*p2(3)-p2(2)*v34(i,j).e(3))-p2k0*(v34(i,j).e(2)*p1
     & (3)-p1(2)*v34(i,j).e(3))
      ceps_0=eps_0*cim
      eps_1=-v34(i,j).e(3)*p1k0+p1(3)*v34(i,j).ek0
      ceps_1=eps_1*cim
      eps_2=-v34(i,j).e(3)*p2k0+p2(3)*v34(i,j).ek0
      ceps_2=eps_2*cim
      vqu=v34(i,j).e(0)*p1(0)-v34(i,j).e(1)*p1(1)-v34(i,j).e(2)*
     & p1(2)-v34(i,j).e(3)*p1(3)
      vqd=v34(i,j).e(0)*p2(0)-v34(i,j).e(1)*p2(1)-v34(i,j).e(2)*
     & p2(2)-v34(i,j).e(3)*p2(3)
      auxa=-v34(i,j).ek0*quqd+p1k0*vqd+p2k0*vqu
      auxb=-v34(i,j).ek0*p2(2)+p2k0*v34(i,j).e(2)
      auxc=+v34(i,j).ek0*p1(2)-p1k0*v34(i,j).e(2)
      res(i,j).a(1,1)=coupr*(auxa+ceps_0)
      res(i,j).a(2,2)=coupl*(auxa-ceps_0)
      res(i,j).b(1,2)=coupl*(auxb-ceps_2)
      res(i,j).b(2,1)=coupr*(-auxb-ceps_2)
      res(i,j).c(1,2)=coupr*(auxc+ceps_1)
      res(i,j).c(2,1)=coupl*(-auxc+ceps_1)
      res(i,j).d(1,1)=coupl*v34(i,j).ek0
      res(i,j).d(2,2)=coupr*v34(i,j).ek0
      end do
      end do
```

Notice the use of ? which has produced the `do i` and `do j` commands.
If the same input lines for `PHACT` had been preceded by

```
dotab
1  2  34
1  2  34
3  4  56
5  6  78
endtab
```

and followed by

```
enddo
```

in the output the fortran lines above would have been followed by two repetitions, one in which 1 would have been changed to 3, 2 to 4 and 34 to 56, the second with 1 to 5, 2 to 6 and 34 to 78.

# 4 Applications

Since the first calculation performed for the process $e^+e^- \to b\bar{b}W^+W^-$[7], `PHACT` has been continuously used and tested in recent years in many high energies computations. In particular the two MonteCarlo's, `WPHACT` [8] and `SIXPHACT` [9][10] have been written with it.
These MC's compute respectively 4 and 6 fermion final states and they have been used for phenomenological applications and compared with results from several other groups in LEP2 and Linear Collider workshops.
The method has thus proven to be able to deal with thousands of Feynman diagrams producing high precision results in short CPU times.
As a full description of the characteristics of these programs is far beyond the scope of the present review, only some relevant and more recent features of them will be sketched in the rest of the section.

## 4.1 WPHACT

`WPHACT` computes all SM 4 fermion processes including Higgs and MSSM Higgs diagrams.
In the first version all fermions except the b's were considered massless. For most WW and ZZ studies this approximation is completely adequate and allows much faster computations.
For particular processes with at least one final state electrons at low angle or lost in the beam pipe, massless approximations are of course not viable, as the matrix elements diverge in this approximation because of the t channel photon propagator. Moreover exact gauge invariance controls the behaviour of such a divergence and even a small violation results in huge numerical differences.
For such a reason and also in order to regulate other possible divergent behaviour at small invariant masses, completely massive matrix elements for all final states have recently been added. Therefore, using `WPHACT` one can now choose between

- massless
- only b massive
- fully massive

matrix elements, depending on the problem at hand.
Correspondingly, phase spaces appropriate for forward region evaluations have been provided.
As an example of these recent improvements, we show in fig. 1 some comparisons at LC energies for the so called single W process $e^+e^- \to e\bar{\nu}u\bar{d}$ (CC20) with the final $e$ at small angle. We notice that even in this difficult region and at high energy the agreement between the massive codes `WPHACT` , `COMPHEP`[11] and `GRC4F`[12] is of the order of a few per mille.

## 4.2 SIXPHACT

Six fermion final states will be relevant at LC for all studies of top, intermediate Higgs, WWZ productions and anomalous couplings (quartic coupling).
As already proven by the studies at LEP2, also for these processes the production times decay approximation will not be sufficient, as it ignores finite width effects and spin correlations and it is not suited for irreducible background evaluation.
`SIXPHACT` can at present compute all Charged Currents six fermion final states, which are the states that can be produced by the decay of two W's and a neutral particle, but not from three neutral particles. These are the cleanest final states to study intermediate Higgs, WWZ, $t\bar{t}$ physics: with one lepton and a neutrino (missing energy) one automatically eliminates the most relevant part of QCD (6 jets, 4 jets + $l\,\bar{l}$, ..) and ZZZ background.
Even if the number of Feynman diagrams for 6 fermion processes is one order of magnitude greater than for 4 fermions, `SIXPHACT` can easily reach per mille accuracy in a few hours at present workstations.
From a technical point of view it is interesting to point out that both `WPHACT` and `SIXPHACT` can produce automatically any distribution the user defines. Even for `SIXPHACT` high statistical accuracy is easily achieved in these distributions. An example of this feature is given in fig. 2, where background and signal

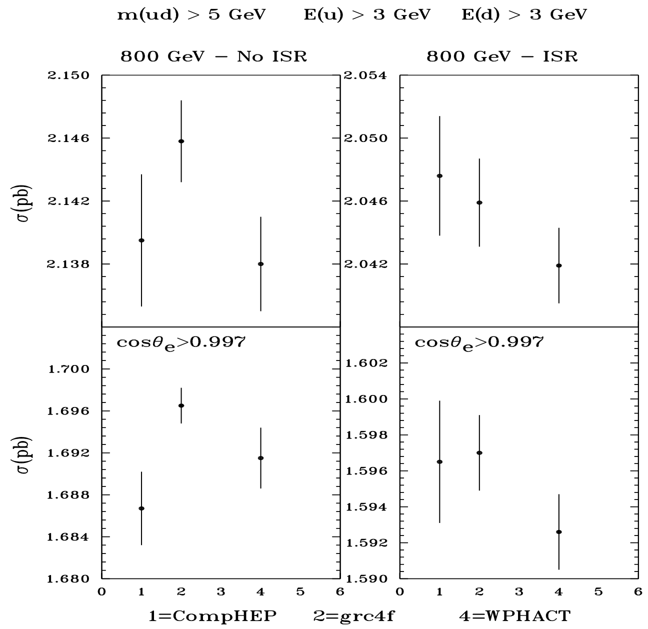



Figure 1: Comparison of the predictions from COMPHEP, GRC4F and WPHACT for the process $e^+e^- \to e\bar{\nu}u\bar{d}$ with the electron in the forward region.

distributions for an intermediate higgs is studied in events with one neutrino, one lepton and four jets. It has to be noticed that the statistical errors on the single bins are not visible in the figure as they are below the percent.

# 5 Conclusions

We have briefly described a method for high energy computations, the way it can be programmed and some of its realizations.
In recent years computer CPU performances have continuously improved. Normal workstations are now faster by about a factor of twenty than five years ago. One may therefore wonder whether it is still important to find faster and easier computational methods in presence of such technical achievements. The answer to this question becomes evident if one considers that at the same time the continuous improvements in statistics, luminosity and energy of the experiments and of the accelerators always demand for higher theoretical precision. The complexity of the problems and processes in the future is orders of magnitude greater than at present and therefore it will be necessary to

- extend automatic computations
- find new techniques for the higher complexity
- continue comparing results of the different approaches

PHACT is intended as a tool that can contribute to this progress. It has so far allowed the construction of fast and precise MC's for $e^+e^-$ colliders. In the future it will hopefully be improved and it will also be used for LHC physics.

**Acknowledgment** I am grateful to the organizers of QFTHEP'99 workshop, and in particular to Edward Boos and Mikhail Dubinin, for the pleasant and constructive atmosphere during the workshop and for the warm hospitality.

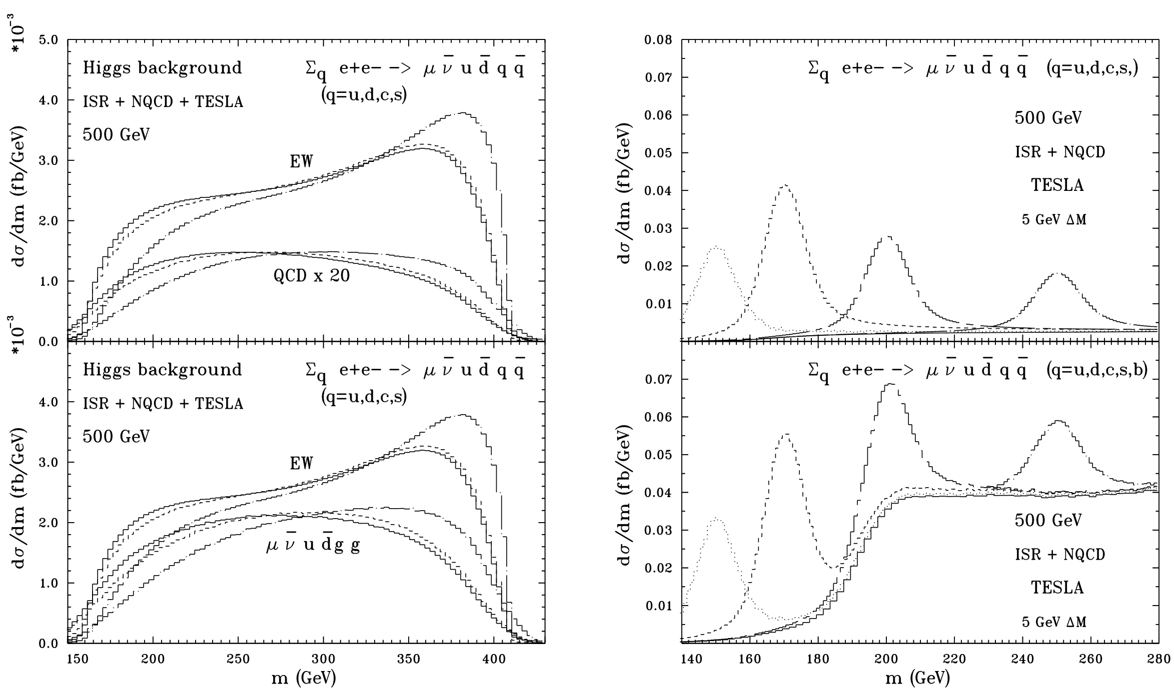



Figure 2: Invariant mass distributions for intermediate higgs background (left) and signal (right). For the background, continuous line corresponds to $(\mu\bar{\nu}u\bar{d})$ mass, dashed to *reconstructed* and chaindot to the *missing*. On the right reconstructed mass distributions with gaussian smearing are reported and the continuous line represents the total background. The others correspond to the total cross sections for $m_h$= 150, 170, 200, 250 GeV. From ref. [10]

# References


[1] For a rather complete set of references on helicity amplitude methods see [2]

[2] A. Ballestrero and E. Maina, Phys. Lett. B350 (1995) 225.

[3] M. Caffo and E. Remiddi, Helv. Phys. Acta 55 (1982) 339,
G. Passarino, Nucl. Phys. B237 (1984) 249.

[4] F.A. Berends, P.H. Daverveldt and R. Kleiss, Nucl. Phys. B253 (1985) 441,
R. Kleiss and W.J. Stirling, Nucl. Phys. B262 (1985) 235.

[5] K. Hagiwara and D. Zeppenfeld, Nucl. Phys. B274 (1986) 1.

[6] A. Ballestrero. Fortran program PHACT

[7] A. Ballestrero, E. Maina, S. Moretti, Phys. Lett. B333 (1994) 434.

[8] E. Accomando and A. Ballestrero, Comput.Phys.Commun. 99 (1997) 270.

[9] E. Accomando, A. Ballestrero and M. Pizzio Nucl.Phys. B512 (1998) 19,
A. Ballestrero, Acta Phys.Polon. B29 (1998) 2811.

[10] E. Accomando, A. Ballestrero and M. Pizzio Nucl.Phys. B547 (1999) 81.

[11] A. Pukhov et al., hep-ph/9908288.

[12] J. Fujimoto et al, Comput.Phys.Commun. 100 (1997) 128.