technische universität
dortmund

Massive Recursive Tensor Integration with MaRTIn

Kai Sieja

July, 3 2025

TU Dortmund University

What does MaRTIn do?

besides complicating discussion in my projects:

What does MaRTIn do?

besides complicating discussion in my projects:

- diagram generation -> extracting amplitudes -> solving Dirac, color, and integral algebra -> algebraic manipulation

What does MaRTIn do?

besides complicating discussion in my projects:

- diagram generation -> extracting amplitudes -> solving Dirac, color, and integral algebra -> algebraic manipulation
- versatile, flexible, extensible, open source

What does MaRTIn do?

besides complicating discussion in my projects:

- diagram generation -> extracting amplitudes -> solving Dirac, color, and integral algebra -> algebraic manipulation
- versatile, flexible, extensible, open source
- very fast even for big, higher order calculations

MaRTIn in a nutshell

- the model files
- the prc directory
- the problems

MaRTIn in a nutshell

- the model files
- the prc directory
- the problems

The example theory

As a simplified example we will work with a Yukawa theory described by

$$\mathcal{L}_{\text{Yuk}} = \bar{\psi}_j \left(i\partial_\mu \gamma^\mu - m_{\psi_j} \right) \psi_j + \frac{1}{2}\partial^\mu \phi \, \partial_\mu \phi - \frac{m_\phi^2}{2}\phi^2 - g_{ij}\bar{\psi}_i \psi_j \phi \,,$$

with $i, j = 1, 2$.
We describe two massive fermions + their antiparticles and (so far) one massive scalar particle.
Fermions and scalar talk via a Yukawa interaction $\bar{\psi}\psi\phi$.

The model files: form

```
*--#[ DEF :
*--#] DEF :

*--#[ POLARIZATION :
*--#] POLARIZATION :

*--#[ GROUPTHEORY :
*--#] GROUPTHEORY :

*--#[ INSERTPROPAGATORS :
*--#] INSERTPROPAGATORS :

*--#[ INSERTVERTICES :
*--#] INSERTVERTICES :

*--#[ INSERTFERMIONVERTICES :
*--#] INSERTFERMIONVERTICES :

*--#[ FF :
*--#] FF :

*--#[ VERTICES :
*--#] VERTICES :
```

⋮

```
*--#[ INSERTCOUPLINGS :
*--#] INSERTCOUPLINGS :
```

declaration of:

symbolic expressions (fields, masses, couplings, ...)

propagators, polarization functions

group structure

The model files: form

```
*--#[ DEF :
*

#define DIMENSION "4"

* Field symbols

s fpsi1 fPsi1 fpsi2 fPsi2 phi;

set allfields: fpsi1 fPsi1 fpsi2 fPsi2 phi;

set fermions: fpsi1 fpsi2;

set antifermions: fPsi1 fPsi2;

*** constants
s g11 g12 g21 g22;

* Masses
s Mpsi1 Mpsi2 Mphi;

* propagators
cf propferm propphi;
```

```
*--#[ INSERTCOUPLINGS :

id vphipsi1psi1 = -i_ * g11;
id vphipsi1psi2 = -i_ * g12;
id vphipsi2psi1 = -i_ * g21;
id vphipsi2psi2 = -i_ * g22;
```

declaration of:

symbolic expressions (fields, masses, couplings, ...)

propagators, polarization functions

group structure

The model files: qgraf

```
*********************************************
** propagators [fermion,Antifermion;...]      **
*********************************************

*  fermions
  [fpsi1,fPsi1,-;pfunct='propferm',m='Mpsi1']
  [fpsi2,fPsi2,-;pfunct='propferm',m='Mpsi2']

*  scalar
   [phi,phi,+;pfunct='propphi',m='Mphi']
```

```
*********************************************
*  vertices with fermions
*  [Antifermion,fermion,other,...]
*********************************************

*  with scalar
  [fPsi1,fpsi1,phi;vfunct='Fphipsi1psi1']
  [fPsi1,fpsi2,phi;vfunct='Fphipsi1psi2']
  [fPsi2,fpsi1,phi;vfunct='Fphipsi2psi1']
  [fPsi2,fpsi2,phi;vfunct='Fphipsi2psi2']
```

declaration of:

what are the (anti-)fields symbols

what are the allowed vertices

which symbol represents which vertex/propagator

The model files: richard_draw

```
{
    "fields": {
        "fPsi1":      ["$\\bar \\psi_1$",         "anti fermion"],
        "fPsi2":      ["$\\bar \\psi_2$",         "anti fermion"],
        "fpsi1":      ["$\\psi_1$",               "fermion"],
        "fpsi2":      ["$\\psi_2$",               "fermion"],
        "phi" :       ["$\\phi$",                "scalar"]
    }
}
```

declaration of:

how to draw each field in the diagrams

how to draw vertices, ...

The problems and the loop.dat files

The Problem directory

- the problems folder may contain many subdirectories
  - → the results directory will mirror the structure

The loop.dat files

- ! they have to be named like: ” loop.{some name}.dat ”
- specifies the process via:
  - initial and final state
  - types of diagrams generated (e.g. exclude tadpoles, 1PI, ...)

```
*****************************************************************
*** INPUT *******************************************************
*****************************************************************

*--#[ QGRAF :
model = 'yukawa.prop.lag' ;
model = 'yukawa.vrtx.lag' ;

in =  fpsi1[q1];
out = fpsi1[q1];

loops = 1 ;
loop_momentum = p;

options = notadpoles;


*--#] QGRAF :

*--#[ MAIN :
*

#define FINALPRINT

* Mathematica output
#define MATHOUTPUT

* write info file
#define INFOFILE

* do not declare rat() as polyratfun
*#define NOPOLYRAT

*****************************************************************
* The physics ****************************************************
*****************************************************************

#define FINALEPLIM "-1"
```

MaRTIn has two modes:

EXPDENO

performs an expansion in external momenta (e.g. for matching)

IRA

performs infrared rearrangement (for extracting poles)

Hands-on: Exercise 1

We want to extend the interactions of out theory:

Add a quartic interaction $c\phi^4$ to our model. Draw all diagrams at 2-loop for the process $\phi\phi \to \phi\phi$, and calculate at least one of those diagrams.

Hands-on: Exercise 2

We want to extend the particle content and to further extend the interactions of out theory:

Add a second massive scalar ϕ_2 to our model. Let it talk to the fermions also via a Yukawa interaction. Also add an interaction between the two scalars.

Bonus

In the Bonus folder you can find model files for the Standard Model. Feel free to play around with it!

Take-home messages:

- → MaRTIn is a great tool for big, higher order calculations
- → MaRTIn is very fast compared to other symbolic algebra programs

Remember: We just saw the tip of the iceberg!

Backup: Installation

Installing form

Go to: https://github.com/form-dev/form and clone the repository to your system

you find step-by-step instructions here:
https://github.com/form-dev/form/blob/master/INSTALL

→ read and follow all of it carefully (**NOT just the overview!**)

Installing Prerequisites of MaRTIn

A detailed overview of all prerequisites is given in https://arxiv.org/abs/2401.04033
The big ones: **QGRAF**, form, richard_draw

Go to: http://cefema-gt.tecnico.ulisboa.pt/~paulo/d.html and download then extract the latest stable version

note: username: anonymous (also the password if required)

Install: use your favorite **fortran** compiler to compile the ".f" (sometimes ".f08")

Test: run the file "qgraf.dat" using your installed QGRAF version (everything is good if you get 34098 connected diagrams)

(macOS only)

MaRTIn (and richard_draw) require GNU utilities, to make this work on macOS:

1. Install Developer Tools

   run: sudo xcode-select --install

2. from https://gist.github.com/skyzyx/3438280b18e4f7c490db8a2a2ca0b9da
   install the GNU packages `coreutils`, `gawk`, `grep`, ... and make sure
   /usr/local/opt/coreutils/libexec/gnubin comes first in `$PATH`

(Windows only)

- Microsoft provides their very own subsystem for Linux

Go to: https://learn.microsoft.com/de-de/windows/wsl/install
and follow the instructions there, then follow the instructions for Linux

Note: Installing a Linux subsystem is strongly recommended for your future life as physicists

Installing Prerequisites of MaRTIn

A detailed overview of all prerequisites is given in https://arxiv.org/abs/2401.04033

The big ones: QGRAF, form, **richard_draw**

Go to: https://gitlab.com/manstam/richard_draw and clone the repository to your system

Test: follow the instructions from https://gitlab.com/manstam/richard_draw/-/blob/main/example_suite/README?ref_type=heads

Installing MaRTIn

Go to: https://gitlab.com/manstam/martin and clone the repository to your system

Install: you find step-by-step instructions here: https://arxiv.org/abs/2401.04033 (Section 2)

Test: see also the end of section 2

Questions?

If there are any problems and/or questions feel free to ask:

macOS : Max Uetrecht (max.uetrecht@tu-dortmund.de)

Linux : Kai Sieja (kai.sieja@tu-dortmund.de)