Computations and generation of elements on the Hopf algebra of Feynman graphs

Michael Borinsky[1]

Humboldt-University Berlin
Departments of Physics and Mathematics

International workshop on Advanced Computing and Analysis Techniques in physics research, 2014

[1] borinsky@physik.hu-berlin.de

Setting: The problem of renormalization

- Perturbative Quantum Field Theory demands for renormalization.

Setting: The problem of renormalization

- Perturbative Quantum Field Theory demands for renormalization.
- Choose BPHZ (also known as the MOM scheme) as renormalization scheme.

Setting: The problem of renormalization

- Perturbative Quantum Field Theory demands for renormalization.
- Choose BPHZ (also known as the MOM scheme) as renormalization scheme.
- Does not require a regulator to make a theory UV-finite.

Setting: The problem of renormalization

- Perturbative Quantum Field Theory demands for renormalization.
- Choose BPHZ (also known as the MOM scheme) as renormalization scheme.
- Does not require a regulator to make a theory UV-finite.
- It has good algebraic properties $\Rightarrow$ Hopf algebra.

The Hopf algebra of Feynman graphs

- The Hopf algebra of Feynman graphs gives renormalization a mathematically sound algebraic framework.

The Hopf algebra of Feynman graphs

- The Hopf algebra of Feynman graphs gives renormalization a mathematically sound algebraic framework.
- The most important object in the Hopf algebra: The coproduct Δ.

The Hopf algebra of Feynman graphs

- The Hopf algebra of Feynman graphs gives renormalization a mathematically sound algebraic framework.
- The most important object in the Hopf algebra: The coproduct Δ.
- It formalizes the BPHZ forest formula.

The Hopf algebra of Feynman graphs

- The Hopf algebra of Feynman graphs gives renormalization a mathematically sound algebraic framework.
- The most important object in the Hopf algebra: The coproduct Δ.
- It formalizes the BPHZ forest formula.
- Gives the prescription how counterterms need to be substracted to make the Feynman integral finite.

The Hopf algebra of Feynman graphs

- The Hopf algebra of Feynman graphs gives renormalization a mathematically sound algebraic framework.
- The most important object in the Hopf algebra: The coproduct Δ.
- It formalizes the BPHZ forest formula.
- Gives the prescription how counterterms need to be substracted to make the Feynman integral finite.
- Definition:

$$\Delta\Gamma := \sum_{\substack{\gamma\subseteq\Gamma \\ \gamma=\bigcup_i \gamma_i \\ \gamma_i \text{ 1PI and } \omega(\gamma_i)\leq 0}} \underbrace{\gamma}_{\text{Counterterms}} \otimes \underbrace{\Gamma/\gamma}_{\text{Cographs}}$$

Motivation for the development of **feyngen** and **feyncop**

- The study of new techniques[2] for systematic of Feynman integration demand for high loop order Feynman diagrams and their coproducts.

[2] Brown and Kreimer 2013; Panzer 2014.

[3] Borinsky 2014.

Motivation for the development of **feyngen** and **feyncop**

- The study of new techniques[2] for systematic of Feynman integration demand for high loop order Feynman diagrams and their coproducts.
- Two **python** programs were developed[3]. **feyngen** for Feynman graph generation:

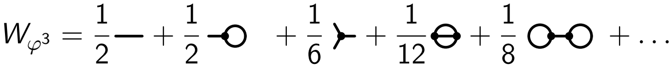
$W_{\varphi^3} = \frac{1}{2}$ $+ \frac{1}{2}$  $+ \frac{1}{6}$  $+ \frac{1}{12}$  $+ \frac{1}{8}$  $+ \ldots$

[2]Brown and Kreimer 2013; Panzer 2014.

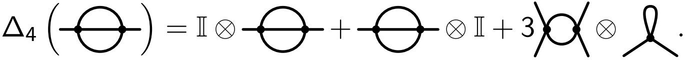
[3]Borinsky 2014.

Motivation for the development of **feyngen** and **feyncop**

- The study of new techniques[2] for systematic of Feynman integration demand for high loop order Feynman diagrams and their coproducts.
- Two **python** programs were developed[3]. **feyngen** for Feynman graph generation:



- and **feyncop** for coproduct computation:



[2] Brown and Kreimer 2013; Panzer 2014.
[3] Borinsky 2014.

Feynman graph generation with **feyngen**

- Generates φ^k for $k \geq 3$, QED (with Furry or without), Yang-Mills, $\varphi^3 + \varphi^4$ diagrams with symmetry factors.

[4] McKay 1981.

Feynman graph generation with **feyngen**

- Generates φ^k for $k \geq 3$, QED (with Furry or without), Yang-Mills, $\varphi^3 + \varphi^4$ diagrams with symmetry factors.
- Uses the established **nauty**[4] package for fast generation and isomorphism testing.

[4] McKay 1981.

Feynman graph generation with **feyngen**

- Generates φ^k for $k \geq 3$, QED (with Furry or without), Yang-Mills, $\varphi^3 + \varphi^4$ diagrams with symmetry factors.
- Uses the established **nauty**[4] package for fast generation and isomorphism testing.
- Filters for connectivity, 1PI-ness, vertex-2-connectedness and tadpole freeness are implemented.

[4] McKay 1981.

Feynman graph generation with **feyngen**

- Generates φ^k for $k \geq 3$, QED (with Furry or without), Yang-Mills, $\varphi^3 + \varphi^4$ diagrams with symmetry factors.
- Uses the established **nauty**[4] package for fast generation and isomorphism testing.
- Filters for connectivity, 1PI-ness, vertex-2-connectedness and tadpole freeness are implemented.
- High performance: 342430 1PI, QED, vertex residue type, 6-loop diagrams can be generated in three days.

[4] McKay 1981.

- **feyngen** assigns an auxillary labeling to the vertices of a graph.

Feynman graph generation with **feyngen**

- **feyngen** assigns an auxillary labeling to the vertices of a graph.
- Edges are represented as pairs of vertices.

Feynman graph generation with **feyngen**

- **feyngen** assigns an auxillary labeling to the vertices of a graph.
- Edges are represented as pairs of vertices.
- Graphs are represented as a list of edges.

Feynman graph generation with **feyngen**

- **feyngen** assigns an auxillary labeling to the vertices of a graph.
- Edges are represented as pairs of vertices.
- Graphs are represented as a list of edges.
- The auxillary labeling is unique for every isomorphism class.

φ^3, 1PI graph generation

- Suppose all two loop, propagator, 1PI φ^3 diagrams shall be generated.

φ^3, 1PI graph generation

- Suppose all two loop, propagator, 1PI φ^3 diagrams shall be generated.
- The call `./feyngen 2 -p -k3 -j2` will yield

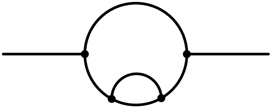
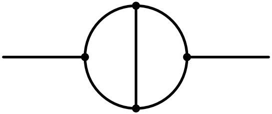
```
phi3_j2_h2 :=
+G[[1,0],[1,0],[2,1],[3,0],[3,2],[4,2],[5,3]]/2
+G[[2,0],[2,1],[3,0],[3,1],[3,2],[4,0],[5,1]]/2
;
```

φ^3, 1PI graph generation

- Suppose all two loop, propagator, 1PI φ^3 diagrams shall be generated.
- The call `./feyngen 2 -p -k3 -j2` will yield

```
phi3_j2_h2 :=
+G[[1,0],[1,0],[2,1],[3,0],[3,2],[4,2],[5,3]]/2
+G[[2,0],[2,1],[3,0],[3,1],[3,2],[4,0],[5,1]]/2
;
```

- Corresponding to the sum of graphs.

$\frac{1}{2}$  $+\frac{1}{2}$  .

Check of validity

- **feyngen** uses zero-dimensional quantum field theory to check the validity of the generated graphs.

Check of validity

- **feyngen** uses zero-dimensional quantum field theory to check the validity of the generated graphs.
- For φ^k-theory the partition function of the zero-dimensional QFT is given by the integral

$$Z_{\varphi^k}(a,\lambda,j) := \int_{\mathbb{R}} \frac{d\varphi}{\sqrt{2\pi a}}\, e^{-\frac{\varphi^2}{2a}+\lambda\frac{\varphi^k}{k!}+j\varphi},$$

where a counts the number of edges, λ the number of vertices and j the number of external edges.

Check of validity

- **feyngen** uses zero-dimensional quantum field theory to check the validity of the generated graphs.
- For φ^k-theory the partition function of the zero-dimensional QFT is given by the integral

$$Z_{\varphi^k}(a,\lambda,j) := \int\limits_{\mathbb{R}} \frac{d\varphi}{\sqrt{2\pi a}}\, e^{-\frac{\varphi^2}{2a}+\lambda\frac{\varphi^k}{k!}+j\varphi},$$

  where a counts the number of edges, λ the number of vertices and j the number of external edges.
- This integral is calculated perturbatively, i.e. by termwise integration:

$$\widetilde{Z}_{\varphi^k}(a,\lambda,j) := \sum_{n,m\geq 0} \int\limits_{\mathbb{R}} \frac{d\varphi}{\sqrt{2\pi a}} \left\{ e^{-\frac{\varphi^2}{2a}} \frac{1}{n!m!} \left(\frac{\lambda\varphi^k}{k!}\right)^n (j\varphi)^m \right\}.$$

- The perturbative expansion of Z_{φ^k} can be obtained by integration or using Feynman diagrams.

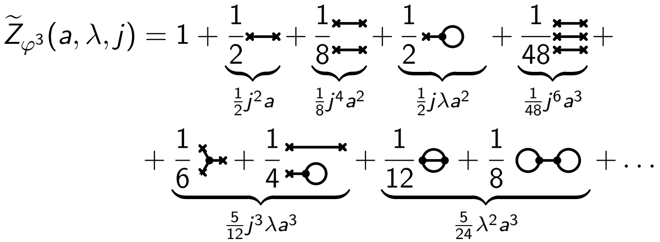
- The perturbative expansion of Z_{φ^k} can be obtained by integration or using Feynman diagrams.
- For φ^3 theory the first few terms after integration are:

$$\begin{aligned}\widetilde{Z}_{\varphi^3}(a,\lambda,j) = 1 + \frac{1}{2}j^2 a + \left(\frac{1}{8}j^4 + \frac{1}{2}j\lambda\right) a^2 + \\ + \left(\frac{1}{48}j^6 + \frac{5}{12}j^3\lambda + \frac{5}{24}\lambda^2\right) a^3 + \dots\end{aligned} \tag{1}$$

- The perturbative expansion of Z_{φ^k} can be obtained by integration or using Feynman diagrams.
- For φ^3 theory the first few terms after integration are:

$$\widetilde{Z}_{\varphi^3}(a,\lambda,j) = 1 + \frac{1}{2}j^2 a + \left(\frac{1}{8}j^4 + \frac{1}{2}j\lambda\right)a^2 + \\ + \left(\frac{1}{48}j^6 + \frac{5}{12}j^3\lambda + \frac{5}{24}\lambda^2\right)a^3 + \dots \tag{1}$$

- Diagrammatically the series is:



(2)

Coproduct calculations with **feyncop**

- **feyncop** can calculate the reduced coproduct of given 1PI diagrams.

Coproduct calculations with **feyncop**

- **feyncop** can calculate the reduced coproduct of given 1PI diagrams.
- Compatible with **feyngen** and **maple**.

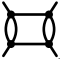
Coproduct calculations with **feyncop**

- **feyncop** can calculate the reduced coproduct of given 1PI diagrams.
- Compatible with **feyngen** and **maple**.
- Can calculate superficially divergent subgraphs, cographs and the tensor products.

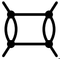
Coproduct calculations with **feyncop**

- **feyncop** can calculate the reduced coproduct of given 1PI diagrams.
- Compatible with **feyngen** and **maple**.
- Can calculate superficially divergent subgraphs, cographs and the tensor products.
- Additionally, **feyncop** can filter a list of graphs for primitive ones.

Calculating the relevant subgraphs

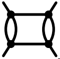
- The graph  is represented as an edge list using an auxillary vertex labeling `G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],[4,0],[5,1],[6,2],[7,3]]`.

Calculating the relevant subgraphs

- The graph  is represented as an edge list using an auxillary vertex labeling
  `G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],`
  `[4,0],[5,1],[6,2],[7,3]]`.
- This can be used as input for **feyncop**:

```
$ echo "G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],
[4,0],[5,1],[6,2],[7,3]]" | ./feyncop -D4
```

Calculating the relevant subgraphs

- The graph  is represented as an edge list using an auxillary vertex labeling
  `G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],[4,0],[5,1],[6,2],[7,3]]`.
- This can be used as input for **feyncop**:

```
$ echo "G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],
[4,0],[5,1],[6,2],[7,3]]" | ./feyncop -D4
```

- And will yield the output:

```
+ D[G[[1,0],[2,0],[2,0],[3,1],[3,1],[3,2],
[4,0],[5,1],[6,2],[7,3]],
[{{1,2}}, {{3,4}}, {{1,2},{3,4}}]]
;
```

- The output line

```
[{{1,2}}, {{3,4}}, {{1,2},{3,4}}]
```

- The output line

  `[{{1,2}}, {{3,4}}, {{1,2},{3,4}}]`

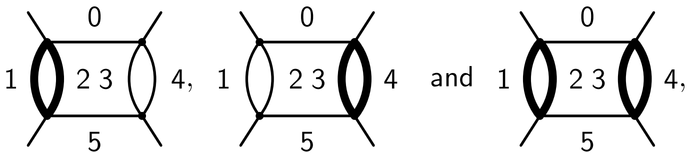
- corresponds to the subgraphs which are composed of superficially divergent, 1PI graphs, represented a by their **edge sets**. The edges are indexed by their order of appearance in the edge list.

- The output line

  `[{{1,2}}, {{3,4}}, {{1,2},{3,4}}]`

- corresponds to the subgraphs which are composed of superficially divergent, 1PI graphs, represented a by their **edge sets**. The edges are indexed by their order of appearance in the edge list.

-



represented as the sets of sets,

`{{1,2}},` `{{3,4}}` and `{{1,2},{3,4}}.`

Check of validity

- Validation using an identity[5] on sums of Feynman graphs:

[5] Suijlekom 2007.

Check of validity

- Validation using an identity[5] on sums of Feynman graphs:

$$\sum_{\Gamma \in \mathcal{T}} \frac{\Delta \Gamma}{|\mathrm{Aut}(\Gamma)|} = \sum_{\substack{\gamma = \left(\prod_i \gamma_i\right) \in \mathcal{F} \\ \omega(\gamma_i) \leq 0}} \sum_{\widetilde{\Gamma} \in \mathcal{T}} \frac{\left|\mathcal{I}(\widetilde{\Gamma}|\gamma)\right|}{|\mathrm{Aut}(\gamma)| \left|\mathrm{Aut}(\widetilde{\Gamma})\right|} \gamma \otimes \widetilde{\Gamma},$$

where $\left|\mathcal{I}(\widetilde{\Gamma}|\gamma)\right|$ is the number of insertions of γ into $\widetilde{\Gamma}$, $\mathcal{T}$ the set of all 1PI graphs and $\mathcal{F}$ the set of all products of 1PI graphs.

[5] Suijlekom 2007.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.
- **feyncop** automizes the combinatorics behind BPHZ renormalization.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.
- **feyncop** automizes the combinatorics behind BPHZ renormalization.
- Easy to extend and fast (because of **nauty**).

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.
- **feyncop** automizes the combinatorics behind BPHZ renormalization.
- Easy to extend and fast (because of **nauty**).
- Both programs were validated.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.
- **feyncop** automizes the combinatorics behind BPHZ renormalization.
- Easy to extend and fast (because of **nauty**).
- Both programs were validated.
- Zero-dimensional quantum field theory was used to check the graph generation.

Conclusion

- Two **python** programs were developed **feyngen** and **feyncop**.
- **feyngen** generates high loop order Feynman graphs using **nauty**.
- **feyncop** automizes the combinatorics behind BPHZ renormalization.
- Easy to extend and fast (because of **nauty**).
- Both programs were validated.
- Zero-dimensional quantum field theory was used to check the graph generation.
- A combinatorical identity was used to validate the coproduct computation.

References

- Borinsky, Michael (2014). "Feynman graph generation and calculations in the Hopf algebra of Feynman graphs". In: *Computer Physics Communications*, (in press).
- Brown, Francis and Dirk Kreimer (2013). "Angles, Scales and Parametric Renormalization". In: *Letters in Mathematical Physics* 103.9, pp. 933–1007.
- McKay, Brendan D. (1981). "Practical Graph Isomorphism". In: *10th. Manitoba Conference on Numerical Mathematics and Computing; Congressus Numerantium, 30*, pp. 45–87.
- Panzer, Erik (2014). "On hyperlogarithms and Feynman integrals with divergences and many scales". In: *Journal of High Energy Physics* 2014.3.
- Suijlekom, Walter D. van (2007). "Renormalization of Gauge Fields: A Hopf Algebra Approach". In: *Communications in Mathematical Physics* 276.3, pp. 773–798.