THOUGHTS AND PERSPECTIVES ON FAST-LOOP COOKING

Alejandro Lorca

Freiburg University

Contents

1. Cooking fast-loop:
   Canteen vs. 'High-class cuisine'
2. Automated canteen logistics
3. Towards the free-open receipt
4. Nice menus to take away

I. COOKING FAST-LOOP

. . . is all about loops, not food

Cooking fast-loop: Origin

We all are aware of the importance of $\boxed{\text{radiative corrections}}$ on determining precise theoretical predictions.

Cooking fast-loop: Origin

We all are aware of the importance of $\boxed{\text{radiative corrections}}$ on determining precise theoretical predictions.

During the last decades we learnt a lot about how to cook loops properly

Cooking fast-loop: Origin

We all are aware of the importance of $\boxed{\text{radiative corrections}}$ on determining precise theoretical predictions.

During the last decades we learnt a lot about how to cook loops properly

Besides loops it came out the whole machinery around

- Feynman diagrams and rules
- Evaluation of matrix elements
- Renormalization schemes
- Regularization of infrared, collinear and ultraviolet divergences
- Treatment of unstable particles and more . . .

Cooking fast-loop: Why canteen loops?

My personal opinion is that . . .

Cooking fast-loop: Why canteen loops?

My personal opinion is that . . .

$$\boxed{\text{Perturbation theory beyond ONE-LOOP } \underline{\text{is not}} \text{ exciting physics}}$$

Cooking fast-loop: Why canteen loops?

My personal opinion is that . . .

$\boxed{\text{Perturbation theory beyond ONE-LOOP }\underline{\text{is not}}\text{ exciting physics}}$

. . . but necessary!!

Cooking fast-loop: Why canteen loops?

My personal opinion is that . . .

Perturbation theory beyond ONE-LOOP is not exciting physics

. . . but necessary!!

To facilitate the task, automated loop calculators are like *canteens*: the output is delivered

- In an automatic way
- Instantly
- In large quantities or individually
- Following standard algorithms and techniques

Cooking fast-loop: Canteen vs. ‘Haute cuisine’

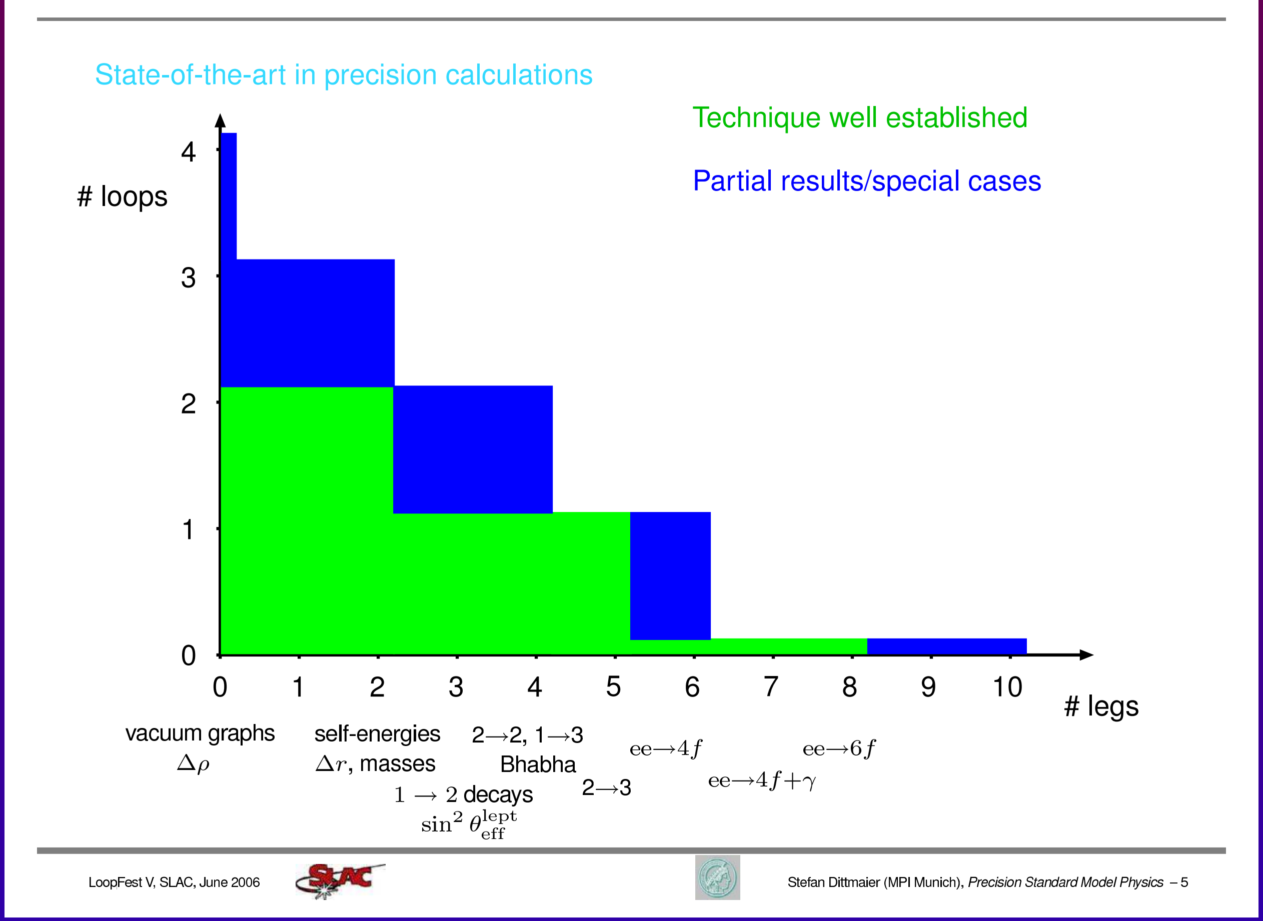
Canteen

High-class cooking

Cooking fast-loop: Canteen vs. ‘Haute cuisine’

Canteen

High-class cooking



State-of-the-art in precision calculations

Technique well established

Partial results/special cases

LoopFest V, SLAC, June 2006

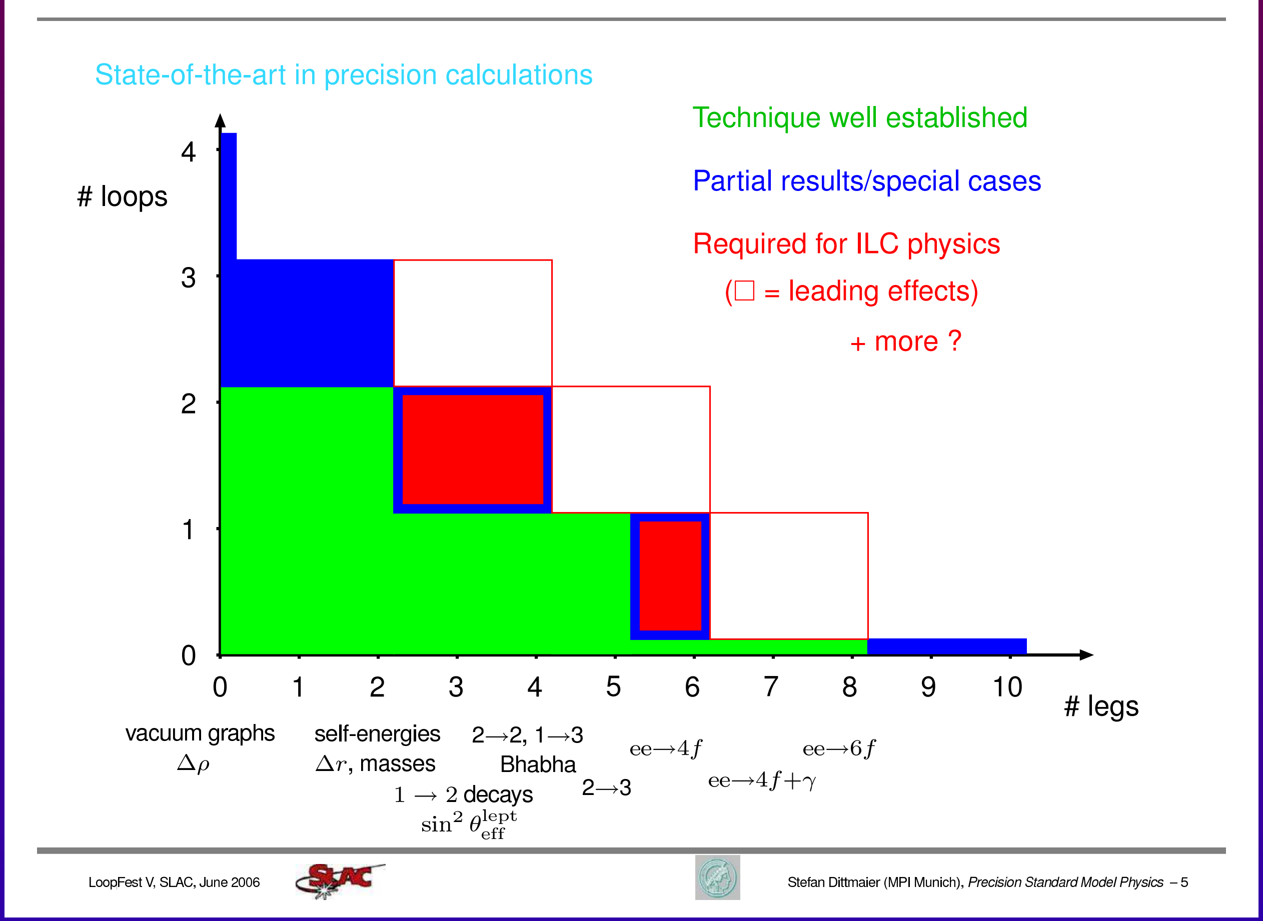
Stefan Dittmaier (MPI Munich), *Precision Standard Model Physics* – 5

Chart from S.Dittmaier’s talk, Loopfest V (June 2006)

Cooking fast-loop: Canteen vs. ‘Haute cuisine’

Canteen

High-class cooking



State-of-the-art in precision calculations

Technique well established

Partial results/special cases

Required for ILC physics

(□ = leading effects)

+ more ?

LoopFest V, SLAC, June 2006

Stefan Dittmaier (MPI Munich), *Precision Standard Model Physics* – 5

Chart from S.Dittmaier’s talk, Loopfest V (June 2006)

II. AUTOMATED CANTEEN LOGISTICS

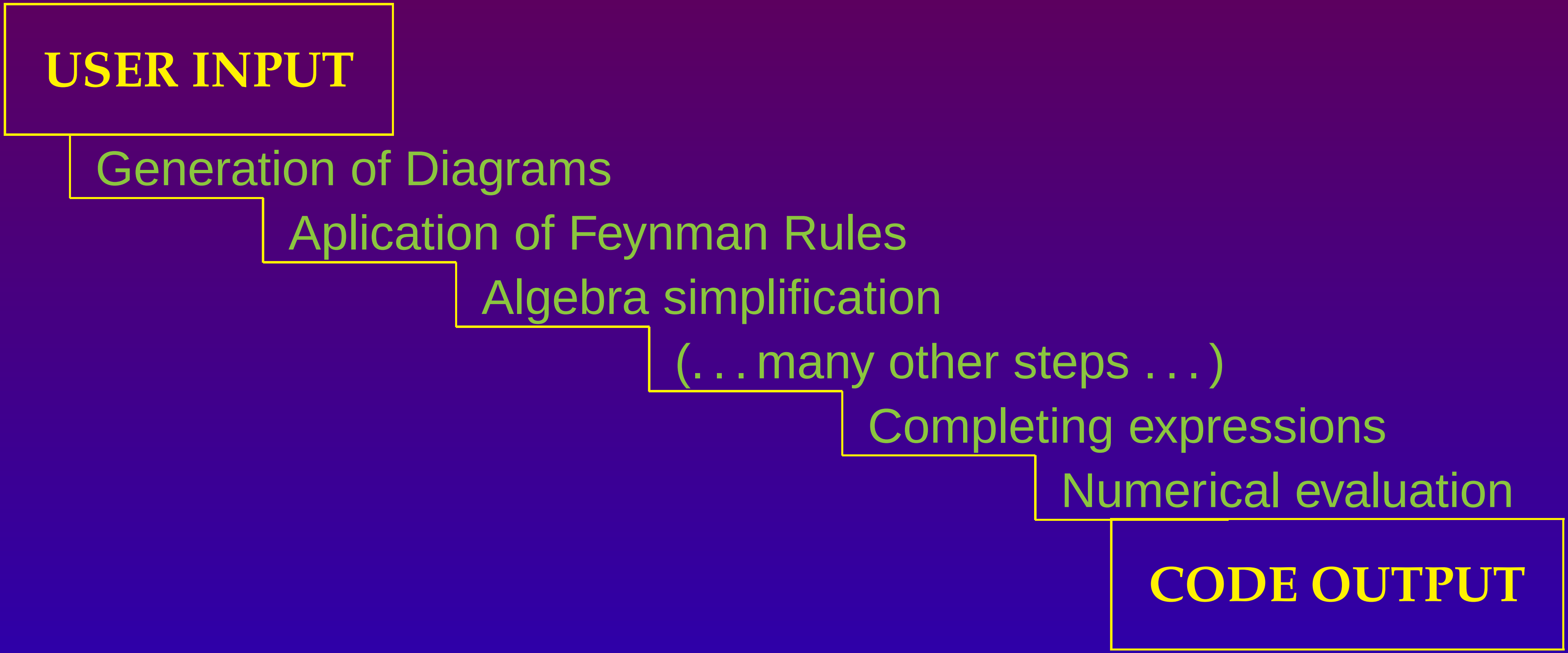
Canteen logistics: Underlying scheme

Logistics := the practical arrangements that are needed in order to make a plan or activity successful.

Canteen logistics: Underlying scheme

Logistics := the practical arrangements that are needed in order to make a plan or activity successful.

➡ For a fully automated tool for perturbative calculations means:



USER INPUT

Generation of Diagrams

Aplication of Feynman Rules

Algebra simplification

(. . . many other steps . . .)

Completing expressions

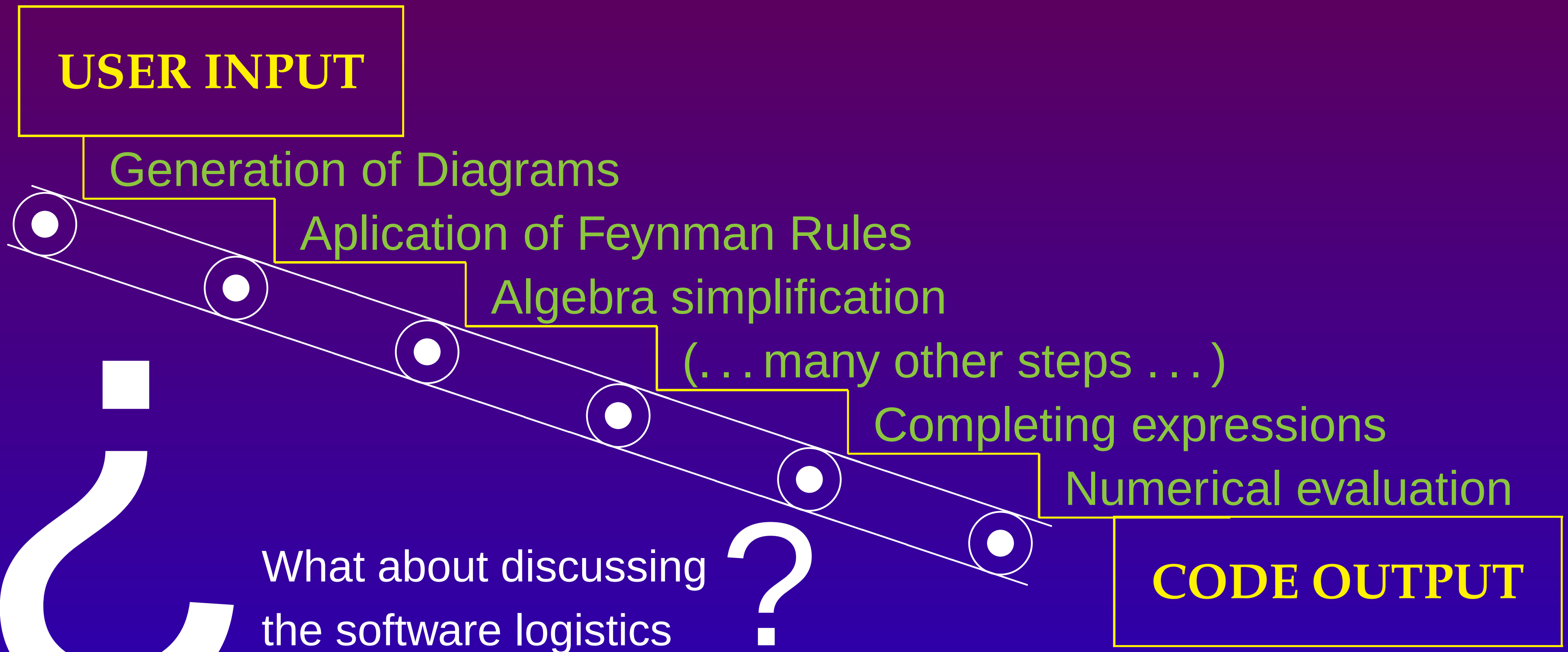
Numerical evaluation

CODE OUTPUT

Canteen logistics: Underlying scheme

Logistics := the practical arrangements that are needed in order to make a plan or activity successful.

➡ For a fully automated tool for perturbative calculations means:



USER INPUT

Generation of Diagrams

Aplication of Feynman Rules

Algebra simplification

(. . . many other steps . . .)

Completing expressions

Numerical evaluation

CODE OUTPUT

¿ What about discussing the software logistics ?

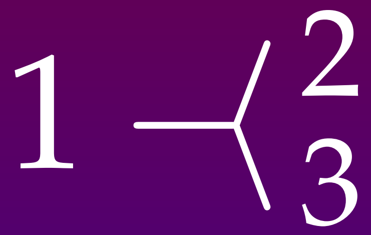
Canteen logistics: Diagram resolution

Let’s imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

Canteen logistics: Diagram resolution

Let’s imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

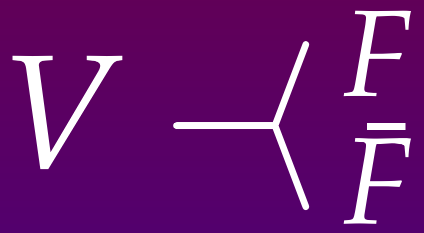
- Graph level (vertex)



Canteen logistics: Diagram resolution

Let’s imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

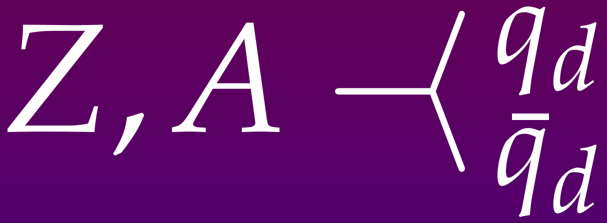
- Graph level (vertex)
- Class level (spin, lorentz)



Canteen logistics: Diagram resolution

Let's imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

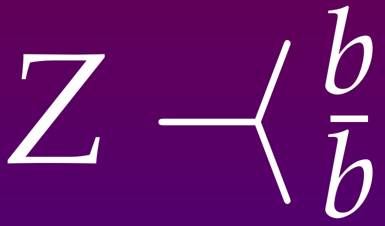
- Graph level (vertex)
- Class level (spin, lorentz)
- Family level (charge, isospin)



Canteen logistics: Diagram resolution

Let’s imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

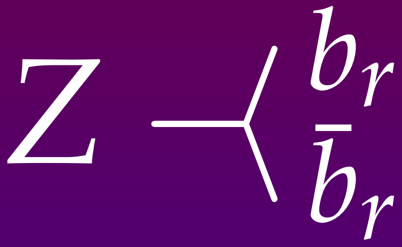
- Graph level (vertex)
- Class level (spin, lorentz)
- Family level (charge, isospin)
- Generation level (mass)



Canteen logistics: Diagram resolution

Let’s imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

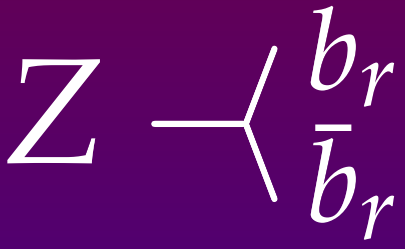
- Graph level (vertex)
- Class level (spin, lorentz)
- Family level (charge, isospin)
- Generation level (mass)
- Color level (color)



Canteen logistics: Diagram resolution

Let's imagine: N Feynman diagrams need to be evaluated correctly, but how efficient are our evaluation algorithms?

- Graph level (vertex)
- Class level (spin, lorentz)
- Family level (charge, isospin)
- Generation level (mass)
- Color level (color)



Does the code use a tree structure for evaluation? $\mathcal{O}(N) \rightarrow \frac{\mathcal{O}(N)}{deg}$

Does the code allow for different renormalization schemes?

Does the code implement Feynman Rules with the Model altogheter?

Does the code accept approximations (i.e. $m_e = 0$) on a diagram level?

Does the code produce graphical quality output (eps) at each level?

Canteen logistics: Math engine

A mathematical engine powers the ability to handle expressions.

➔ When loops enter the game, FORM is the favourite

pro	contra
✔ suited for LARGE expressions	✘ Interaction with other tools via file i/o (v3.2 improves!)
✔ Designed for QFT	✘ No numerical support
✔ Portable	✘ Bug fixing
✔ Free (no usage cost)	✘ Close (no source available)

Canteen logistics: Math engine

A mathematical engine powers the ability to handle expressions.

➔ When loops enter the game, FORM is the favourite

pro

- ✔ suited for LARGE expressions
- ✔ Designed for QFT
- ✔ Portable
- ✔ Free (no usage cost)

contra

- ✘ Interaction with other tools via file i/o (v3.2 improves!)
- ✘ No numerical support
- ✘ Bug fixing
- ✘ Close (no source available)

Alternatives without these contras? GiNaC (framework for symbolic C++)
If it can be as ‘good’ as FORM is still on discussion . . .

Canteen logistics: Applicability domain

Difficult issues to integrate:

1. Electroweak corrections require different treatment than QCD
2. Mathematical techniques often apply only in specific cases or are known to work better for a given situation
3. CPU power, memory or computing time can be also a restriction

Canteen logistics: Applicability domain

Difficult issues to integrate:

1. Electroweak corrections require different treatment than QCD
2. Mathematical techniques often apply only in specific cases or are known to work better for a given situation
3. CPU power, memory or computing time can be also a restriction

Applicability domain algorithm := Set of instructions required by the code to solve a generic problem in an optimal way. The solution is found by analizing the difficulty of the given enunciate and matching the applicability regions of the different implemented techniques.

Canteen logistics: Applicability domain

Difficult issues to integrate:

1. Electroweak corrections require different treatment than QCD
2. Mathematical techniques often apply only in specific cases or are known to work better for a given situation
3. CPU power, memory or computing time can be also a restriction

Applicability domain algorithm := Set of instructions required by the code to solve a generic problem in an optimal way. The solution is found by analizing the difficulty of the given enunciate and matching the applicability regions of the different implemented techniques.

⟶ Not solved yet, at least in fully automated loop calculators. We need new ideas and support here: coming from adaptative algorithm theory, artificial intelligence, robotics, . . . Moment for an interdisciplinar coffee?

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information → less memory, less lines

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information ⟶ less memory, less lines
4. Portability: Available for different hardware and operating systems

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information → less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information $\longrightarrow$ less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information ⟶ less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list
7. Testability: To be built-in at design phase: local and overall behaviour

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information $\longrightarrow$ less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list
7. Testability: To be built-in at design phase: local and overall behaviour
8. Usability: Practical and convenient to use. A GUI is very welcome

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information → less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list
7. Testability: To be built-in at design phase: local and overall behaviour
8. Usability: Practical and convenient to use. A GUI is very welcome
9. Robustness: Correct behaviour and trustable, neither crashing nor hanging

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information → less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list
7. Testability: To be built-in at design phase: local and overall behaviour
8. Usability: Practical and convenient to use. A GUI is very welcome
9. Robustness: Correct behaviour and trustable, neither crashing nor hanging
10. Efficiency: Resources (cpu/memory/time) not wasted for a given task

Canteen logistics: Software quality

Users want to use high quality tools. Some of the factors with major impact are

1. Understandability: Clear purpose of the product (within user context)
2. Completeness: Fully develop, missing ingredients is not desirable
3. Conciseness: No excesive information ⟶ less memory, less lines
4. Portability: Available for different hardware and operating systems
5. Consistency: Uniform notation, symbols and terminology
6. Mantainability: Updates, extensions, bug-fixing, mailing-list
7. Testability: To be built-in at design phase: local and overall behaviour
8. Usability: Practical and convenient to use. A GUI is very welcome
9. Robustness: Correct behaviour and trustable, neither crashing nor hanging
10. Efficiency: Resources (cpu/memory/time) not wasted for a given task
11. Security: Data protection and lost, access control against bad management

III. TOWARDS THE FREE-OPEN RECEIPT

Towards free-open receipt: Why free?

Public scientific community is ‘mantained’ by goverments, fundations and other institutions . . .

. . . so it seems reasonable not to sell the codes we write.

The reward is actually coming from the acknowledgement of the community itself,

$$\boxed{\text{+ citations} \Rightarrow \text{+ success}}$$

so make it free and let the people enjoy!

Towards free-open receipt: Why free?

Public scientific community is ‘mantained’ by goverments, fundations and other institutions . . .

. . . so it seems reasonable not to sell the codes we write.

The reward is actually coming from the acknowledgement of the community itself,

$$\boxed{\text{+ citations} \Rightarrow \text{+ success}}$$

so make it free and let the people enjoy!

A second weaker requisite would be that the code is also free-based.

☞ Free OS (GNU/LINUX FREEBSD) are preferred over $Windows

☞ Underlying packages with $payment licenses are discorauged

Towards free-open receipt: Why free?

Public scientific community is ‘mantained’ by goverments, fundations and other institutions . . .

. . . so it seems reasonable not to sell the codes we write.

The reward is actually coming from the acknowledgement of the community itself,

$$\boxed{\text{+ citations} \Rightarrow \text{+ success}}$$

so make it free and let the people enjoy!

A second weaker requisite would be that the code is also free-based.

☞ Free OS (GNU/LINUX FREEBSD) are preferred over $Windows

☞ Underlying packages with $payment licenses are discorauged

♣ This attitude lowers costs and strengths solidarity ♡

Towards free-open receipt: Why open?

Open-Source projects are nowadays very frequent, and a revulsive against software patents.

- Increase transparency (everyone can look inside)
- Users might understand better the *modus-operandi*
- Feed-back turns easier (bug reports, etc)
- Allow suitable adaptation, transformation, evolution

In our community, the lack of manpower stops many desirable improvements.
☞ By making open a project it helps users and invite potential collaborators ☺.

IV. NICE MENUS TO TAKE AWAY

Menus to take away

- Russian Bliny with web-caviar: SANC
  (ask D.Bardin, web-based, JAVA 1.5, FORM, FORTRAN)
- German-Spanish fermionic tapas: *a*I̊TALC (ask me)
  (DIANA(ask M.Tentyukov), FORM, LOOPTOOLS, FORTRAN)
- German Wolframspätzle: FEYNARTS & co.
  (ask T.Hahn, MATHEMATICA, FORM, FORTRAN)
- Japanesse SUSHY a la multileg: GRACE
  (C, FORTRAN, MOTIF, FORM)

For vegetarians (no loops inside!)

- Russian mushrooms' Solyanka: CALCHEP (ask A.Pukhov, C)
- American vegetarian burger: MADGRAPH (web-support, FORTRAN, PERL)

Summary & Outlook

- Automating loop calculations is an important subject
  - it is the quick&safe way to have calculations done
  - we are fit for 5legs-1loop, 2legs-2loops
  - next goal: 6legs-1loop, 4legs-2loops

Summary & Outlook

- Automating loop calculations is an important subject
  - it is the quick&safe way to have calculations done
  - we are fit for 5legs-1loop, 2legs-2loops
  - next goal: 6legs-1loop, 4legs-2loops
- Our codes are like any other software tool
  - efficient algorithms and the math engine are crucial
  - domain of applicability problem to be solved
  - rigorous software quality checks are needed

Summary & Outlook

- Automating loop calculations is an important subject
  - ► it is the quick&safe way to have calculations done
  - ► we are fit for 5legs-1loop, 2legs-2loops
  - ► next goal: 6legs-1loop, 4legs-2loops
- Our codes are like any other software tool
  - ► efficient algorithms and the math engine are crucial
  - ► domain of applicability problem to be solved
  - ► rigorous software quality checks are needed

♣ Remember: Free-Open receipt for all your plates!

Summary & Outlook

- Automating loop calculations is an important subject
  - ▶ it is the quick&safe way to have calculations done
  - ▶ we are fit for 5legs-1loop, 2legs-2loops
  - ▶ next goal: 6legs-1loop, 4legs-2loops
- Our codes are like any other software tool
  - ▶ efficient algorithms and the math engine are crucial
  - ▶ domain of applicability problem to be solved
  - ▶ rigorous software quality checks are needed

♣ Remember: Free-Open receipt for all your plates!

☞ If you are interested try a menu from the market

Visual summary

Physicists will still cook hand-made loops

Visual summary

but automated alternatives exists,

Visual summary

Color loops (QCD) are still a quest

Visual summary

Multiloop is too much for us!