XLOOPS - GiNaC

H. Spiesberger (Mainz)
CA mini-workshop
Karlsruhe, 19.5.2004

The GiNaC group:
C. Bauer, A. Frink,
R. Kreckel, J. Vollinga

The XLOOPS group:
C. Bauer, H.S. Do, A. Frink,
H. S., J. Vollinga

with contributions from:
L. Brücher, J. Franzkowski,
R. Kreckel, D. Kreimer

GiNaC: computer algebra in C++
C++ library for symbolic manipulations

History: development started in 1999
first release 0.4.0: Nov 26, 1999
since then $\sim$ 50 updates
present version 1.2.1, April 23, 2004
$\rightarrow$ www.ginac.de

Motivation: to replace Maple as programming language for the *xloops* project

GiNaC based projects

- nestedsums: series expansion of transcendental functions (S. Moch, P. Uwer, S. Weinzierl)
- gTybalt: interactive computer algebra system (interfacing GiNaC with ROOT and TeXmacs, S. Weinzierl)
- PyGiNaC interface to Python (P. Peterson)
- feelfem FEM code generation for parallel computation (H. Fujio)
- PURRS The Parma University's Recurrence Relation Solver (R. Bagnara et al)

Why C++?

- standardized and common language
- structured data types, modularization
- infrastructure: compilers, libraries, development tools
- operator overloading
- free software

Requirements

- ANSI compliant C++ compiler (gcc > 2.95.3)
- CLN (class library for numbers, B. Haible) for arbitrary precision arithmetic

Usage

- C++ program: edit, compile, link, execute
- ginsh: a simple interactive shell (without loops and conditionals)
- interface to the C++ interpreter Cint (M. Goto)

Requirements for a CAS (a selection)

- Open system
  - to allow proving validity
  - to allow extensions, improvements, bug fixes (users and developers are identical!)
- Development tools
  - to allow distributed development by developers working in parallel or in continuation

  modularity, object oriented, orthogonality
- No hidden limitations
  no hidden assumptions, no automatic expansion or simplification
- Extensibility, allow linkage to modules written in different languages
- Persistence
- Compiled language: efficiency for numerial evaluation

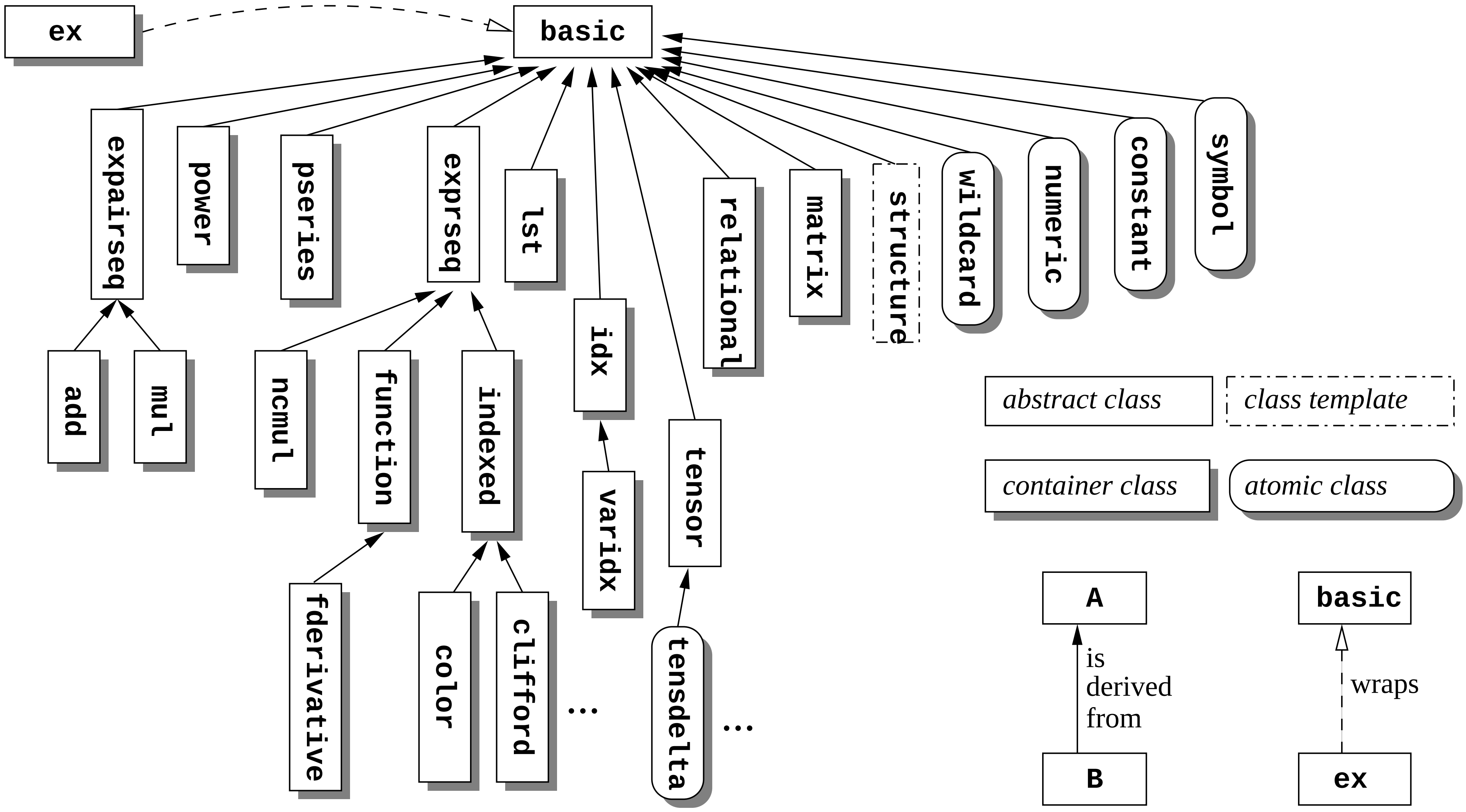
GiNaC features

- arbitrary precision arithmetic (from CLN: integer, floating point, complex numbers, state-of-the-art implementation of some functions and constants)
- simple manipulations: expansion of products, collect, substitutions, pattern matching
- normalization of rational functions, polynomial division
- matrix operations, solve system of linear equations
- indexed objects: momenta, Dirac and color algebra
- many special functions (including multiple polylogarithms)
- differentiation and series expansion
- numerous and flexible output formats and archiving
- extensibility for user-defined classes and functions, easy modification of existing classes
- numerical integration

missing things

- solve for non-linear (system of) equations
- calculation of limits
- symbolic integration
- domains: assumptions about the range of parameter values
- full support for factorization, non-trivial simplifications

Class hierarchy of GiNaC



Mathematical objects in GiNaC

`symbol`	Algebraic symbols a, x, $y\dots$
`constant`	Constants like π
`numeric`	All kinds of numbers, 42, $7/3*I$, $3.14159\dots$
`add`	Sums like $x{+}y$ or $a-(2*b){+}3$
`mul`	Products like $x*y$ or $2*a\hat{\ }2*(x{+}y{+}z)/b$
`ncmul`	Products of non-commutative objects
`power`	Exponentials such as $x\hat{\ }2$, $a\hat{\ }b$, $\sqrt{2}\ \dots$
`pseries`	Power Series, e.g. $x-1/6*x\hat{\ }3{+}1/120*x\hat{\ }5{+}O(x\hat{\ }7)$
`function`	A symbolic function like $\sin 2x$
`lst`	Lists of expressions $\{x,\ 2*y,\ 3{+}z\}$
`matrix`	mxn matrices of expressions
`relational`	A relation like the identity x==y
`indexed`	Indexed object like A_ij
`tensor`	Special tensor like the delta and metric tensors
`idx`	Index of an indexed object
`varidx`	Index with variance
`spinidx`	Index with variance and dot (used in Weyl-van-der-Waerden spinor formalism)
`wildcard`	Wildcard for pattern matching
`structure`	Template for user-defined classes

Test on numbers

Method	**Returns true if the object is...**
`.is_zero()`	...equal to zero
`.is_positive()`	...not complex and greater than 0
`.is_integer()`	...a (non-complex) integer
`.is_pos_integer()`	...an integer and greater than 0
`.is_nonneg_integer()`	...an integer and greater equal 0
`.is_even()`	...an even integer
`.is_odd()`	...an odd integer
`.is_prime()`	...a prime integer (probabilistic primality test)
`.is_rational()`	...an exact rational number (integers are rational, too)
`.is_real()`	...a real integer, rational or float (i.e. is not complex)
`.is_cinteger()`	...a (complex) integer (such as $2-3*I$)
`.is_crational()`	...an exact (complex) rational number (such as $2/3+7/2*I$)

GiNaC methods (incomplete overview)

implemented as C++ functions or methods with wrapper functions

- automatic evaluation and simplification
- numerical evaluation: `evalf`, `Digits`, tests on numbers
- accessing subexpressions and substitutions: `nops`, `op(i)`, iterators, `has`, `find`, `info`, wildcard, `subs`; `map_function`
- polynomial arithmetic
  `expand`, `collect`, `degree`, `divide`, `gcd`;
  `normal` (cancel gcd, fraction addition)
- matrix operations: `evalm`, `transpose`, `determinant`, `charpoly`, `inverse`, `solve`
- indexed objects: substitutions, symmetries, `simplify_indexed`, predefined tensors
- non-commutative objects: `ncmul`, clifford, color, `dirac_gamma`, `color_T`, trace
- functions: `diff`, `series`

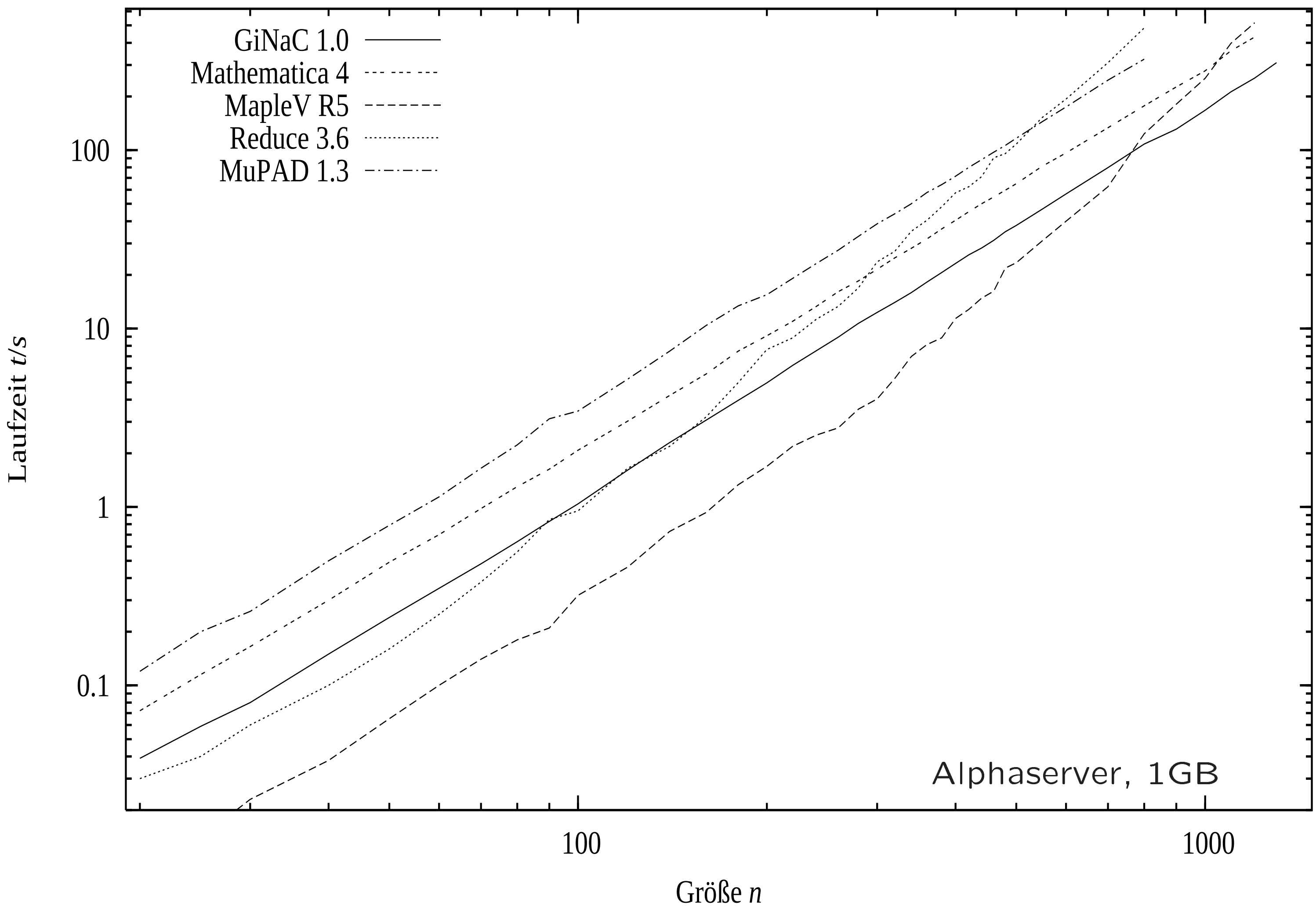
Attributes of expressions (info() method)

Flag	Returns true if the object is...
numeric	...a number (same as is_<numeric>(...))
real	...a real integer, rational or float (i.e. is not complex)
rational	...an exact rational number (integers are rational, too)
integer	...a (non-complex) integer
crational	...an exact (complex) rational number (such as $2/3+7/2*I$)
cinteger	...a (complex) integer (such as $2-3*I$)
positive	...not complex and greater than 0
negative	...not complex and less than 0
nonnegative	...not complex and greater than or equal to 0
posint	...an integer greater than 0
negint	...an integer less than 0
nonnegint	...an integer greater than or equal to 0
even	...an even integer
odd	...an odd integer
prime	...a prime integer (probabilistic primality test)
relation	...a relation (same as is_a<relational>(...))
relation_equal	...a == relation
relation_not_equal	...a != relation
relation_less	...a < relation
relation_less_or_equal	...a <= relation
relation_greater	...a > relation
relation_greater_or_equal	...a >= relation
symbol	...a symbol (same as is_a<symbol>(...))
list	...a list (same as is_a<lst>(...))
polynomial	...a polynomial (i.e. only consists of sums and products of numbers and symbols with positive integer powers)
integer_polynomial	...a polynomial with (non-complex) integer coefficients
cinteger_polynomial	...a polynomial with (possibly complex) integer coefficients (such as $2-3*I$)
rational_polynomial	...a polynomial with (non-complex) rational coefficients
crational_polynomial	...a polynomial with (possibly complex) rational coefficients (such as $2/3+7/2*I$)
rational_function	...a rational function ($x+y$, $z/(x+y)$)
algebraic	...an algebraic object ($sqrt(2)$, $sqrt(x)-1$)

Pre-defined functions in GiNaC

Name	**Function**
`abs(x)`	absolute value
`csgn(x)`	complex sign
`conjugate(x)`	complex conjugation
`sqrt(x)`	square root (not a GiNaC function, rather an alias for `pow(x, numeric(1, 2))`)
`sin(x)`	sine
`cos(x)`	cosine
`tan(x)`	tangent
`asin(x)`	inverse sine
`acos(x)`	inverse cosine
`atan(x)`	inverse tangent
`atan2(y, x)`	inverse tangent with two arguments
`sinh(x)`	hyperbolic sine
`cosh(x)`	hyperbolic cosine
`tanh(x)`	hyperbolic tangent
`asinh(x)`	inverse hyperbolic sine
`acosh(x)`	inverse hyperbolic cosine
`atanh(x)`	inverse hyperbolic tangent
`exp(x)`	exponential function
`log(x)`	natural logarithm
`Li2(x)`	dilogarithm
`Li(m, x)`	classical polylogarithm as well as multiple polylogarithm
`S(n, p, x)`	Nielsen's generalized polylogarithm
`H(m, x)`	harmonic polylogarithm
`zeta(m)`	Riemann's zeta function as well as multiple zeta value
`zeta(m, s)`	alternating Euler sum
`zetaderiv(n, x)`	derivatives of Riemann's zeta function
`tgamma(x)`	gamma function
`lgamma(x)`	logarithm of gamma function
`beta(x, y)`	beta function (`tgamma(x)*tgamma(y)/tgamma(x+y)`)
`psi(x)`	psi (digamma) function
`psi(n, x)`	derivatives of psi function (polygamma functions)
`factorial(n)`	factorial function
`binomial(n, m)`	binomial coefficients
`Order(x)`	order term function in truncated power series

Runtime test I (Denny Fliegner test)

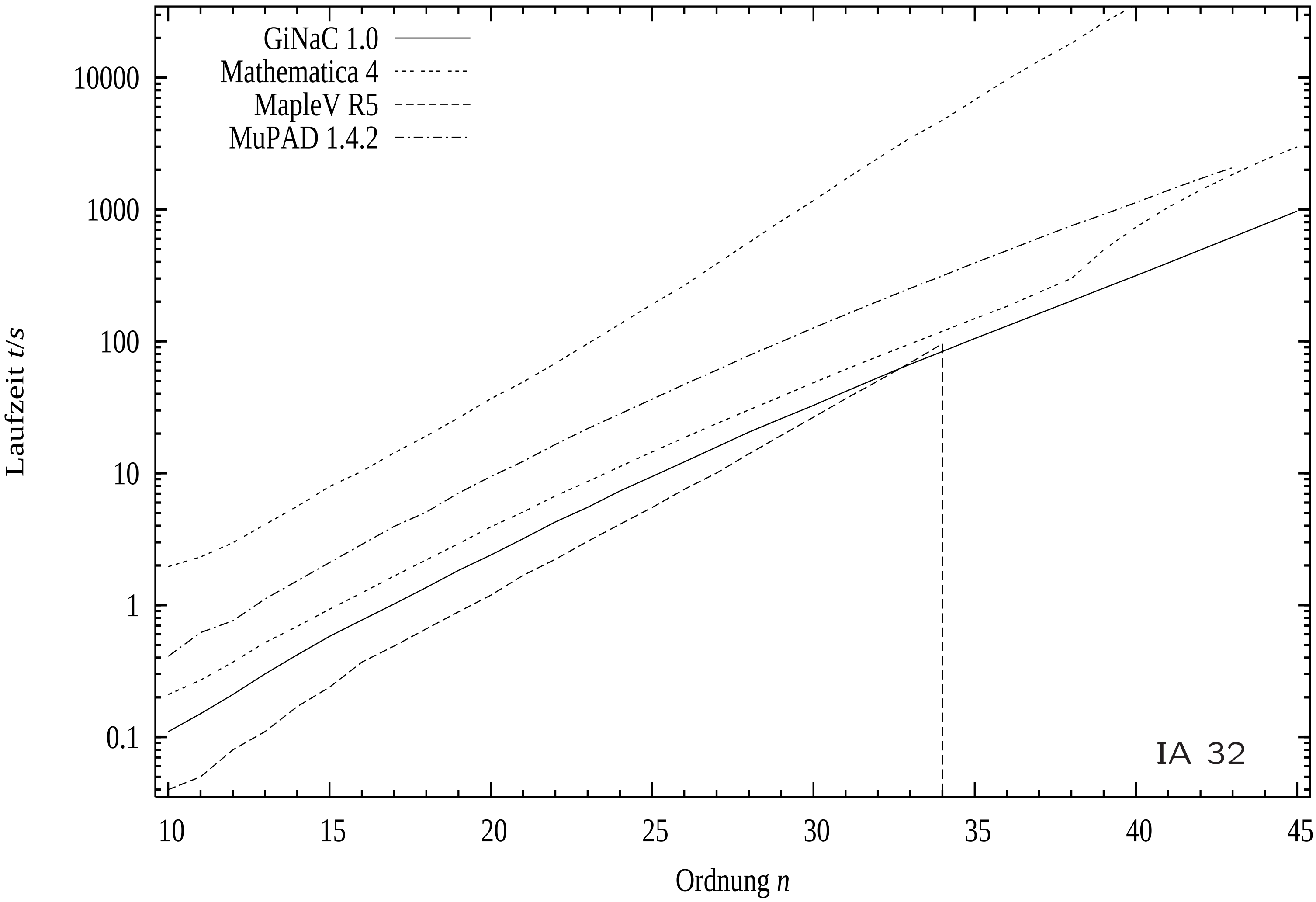


expand expression of n variables:

1) $e = \left(\sum_{i=0}^{n-1} a_i\right)^2$, expand
2) substitute $a_0 = -\sum_{i=2}^{n-1} a_i$
3) expand $\to e = a_1^2$

match terms in a sum of $\sim 2n^2$ terms

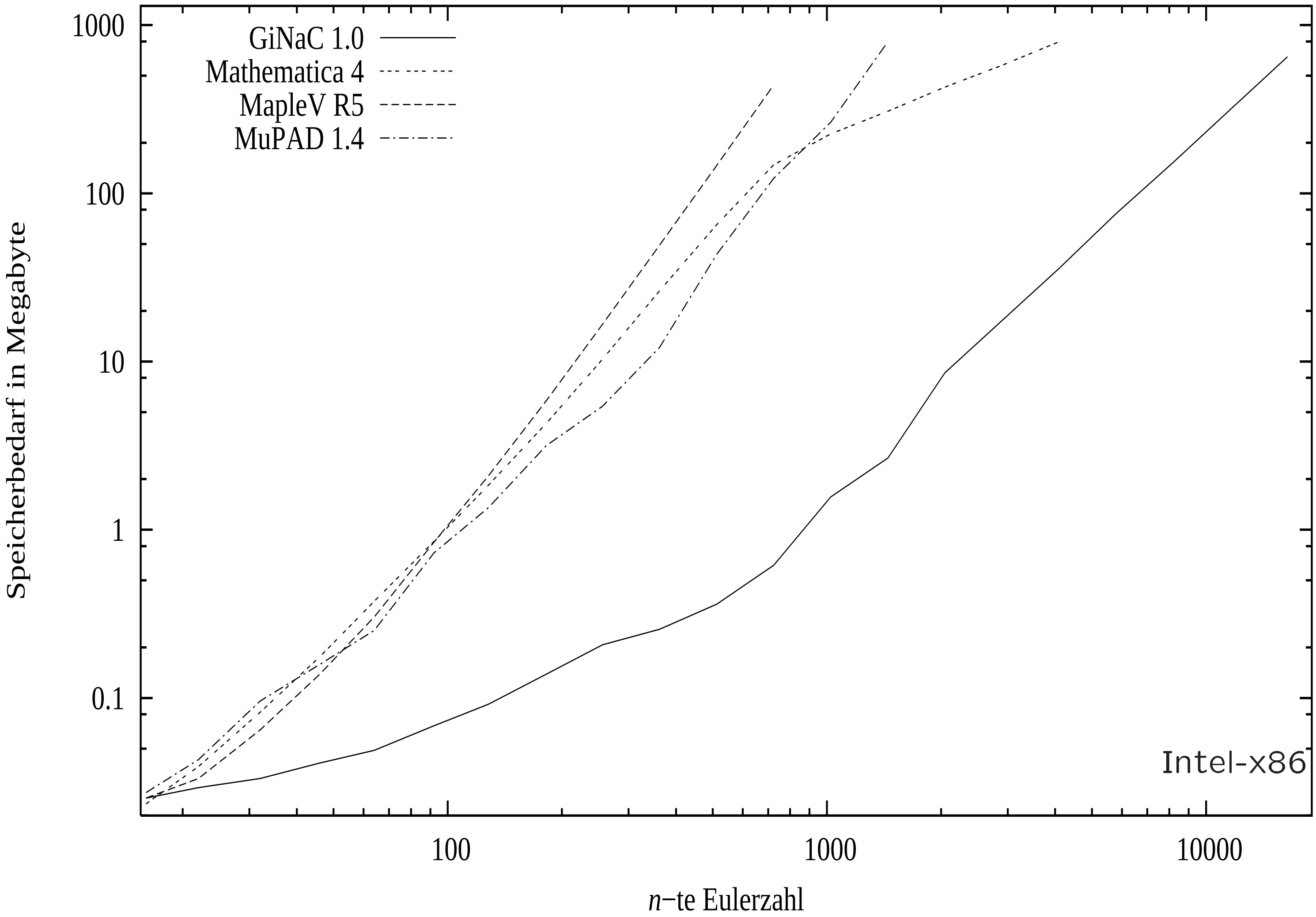
Runtime test II



expand $\Gamma(x)$ around $x = 0$

(Mathematica, lower curve: unevaluated polygamma function
upper curve: forced evaluation of $\Psi_n(1)$)

Memory usage



Calculation of Euler numbers (expansion coefficients of $1/\cosh(x)$)

Runtime of some benchmark tests

	System:	**GiNaC**	**MapleV**	**MuPAD**	**Pari-GP**	**Singular**
	Version:	**1.0.0**	**R5**	**2.0.0**	**2.0.19β**	**2.0.1**
Benchmark	Erscheinungsdatum:	11/2001	9/1997	5/2001	3/2000	6/2001
A:	Teile Fakultäten $\frac{(1000+i)!}{(900+i)!}\Big\|_{i=1}^{100}$	0.20	6.66	1.13	0.37	0.21
B:	$\sum_{i=1}^{1000} 1/i$	0.019	0.08	0.10	0.041	0.51
C:	ggT(große Zahlen)	0.25	10.2	3.01	1.65	0.31
D:	$\sum_{i=1}^{10} iyt^i/(y+it)^i$	0.68	0.13	2.20	0.20	0.11
E:	$\sum_{i=1}^{10} iyt^i/(y+\|5-i\|t)^i$	0.56	0.05	2.20	0.11	2.30
F:	ggT(bivariate Polynome)	0.07	0.08	1.36	0.057	0.09
G:	ggT(trivariate Polynome)	2.01	2.89	5.43	99.5	0.27
H:	det(Rang 80 Hilbert-Matrix)	9.12	33.5	19.5	3.97	15.9
I:	Invertiere Rang 40 Hilbert-Matrix	2.73	6.41	5.10	0.62	0.35
J:	Verifiziere I	1.52	2.28	1.90	0.22	0.02
K:	Invertiere Rang 70 Hilbert-Matrix	17.8	92.0	32.2	5.90	1.84
L:	Verifiziere K	8.69	21.6	10.6	1.57	0.058
M_1:	det(symbolische 26×26 Matrix)	0.36	0.40	0.51	0.016	0.003
M_2:	det(symbolische 101×101 Matrix)	1517.3	*GU*	*CR*	*CR*	28.2
N:	$\sum$ rationaler Funktionen vereinfachen	704.4	*GU*	*CR*	*CR*	*CR*
O_1:	Drei Rang 15 Determinanten (Mittel)	39.5	*GU*	*CR*	*CR*	53.8
O_2:	ggT der Ergebnisse aus O_1	*CR*	*UN*	*UN*	*UN*	*CR*
P:	det(Rang 101, dünn besetzte Matrix)	1.10	12.6	4.3	0.09	0.023
P':	det(Rang 101, weniger dünn besetzt)	5.61	13.3	11.3	0.38	2.85
Q:	charpoly(P)	103.1	1429.7	2751.2	0.15	0.14
Q':	charpoly(P')	209.8	1497.3	2796.1	*CR*	3.82

Intel P-III 450 MHz, 384 MB RAM, runtimes in sec

"GA" = gave up
"CR" = crashed
"UN" = unable

Xloops: Toolkit for the automatic calculation of one- and two-loop Feynman diagrams
no restrictions on masses,
external momenta, tensor degree
analytical + numerical results

integration techniques based on work by D. Kreimer (1991-93)

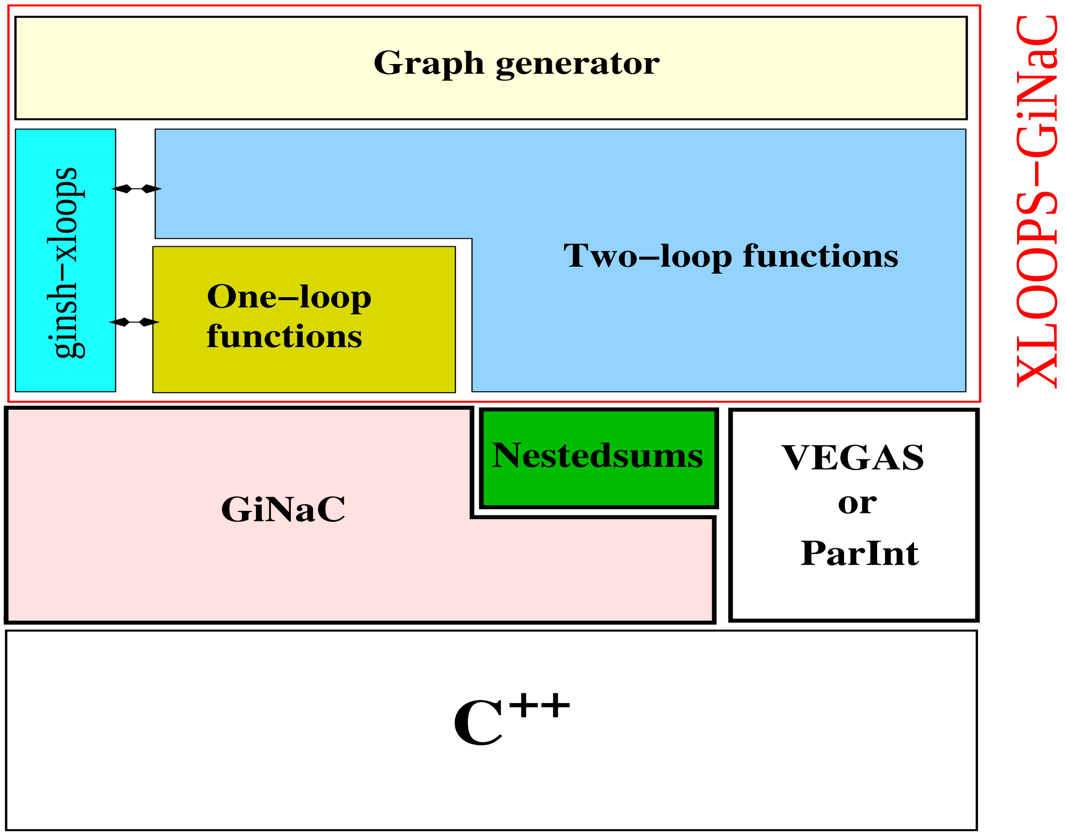
contributions from U. Kilian, J. Franzkowski
A. Frink, L. Brücher,
R. Kreckel

$\rightarrow$ Xloops-MapleV (Oct 1997)

The Xloops group now C. Bauer, H.S. Do,
H.S., J. Vollinga

$\rightarrow$ Xloops (GiNaC) version 0.1.3 (Nov 2001)

Xloops Concept



Status

- One-loop: up to 4 external legs
  analytical results (dim. reg.)
- Two-loop: 2-point functions
  analytical results (divergent part) + numerical results (finite part)
  3-point functions in preparation

Graph generation: working, but not yet released

Manual in preparation

Integration technique

- parallel-orthogonal space of loop momenta

  example: 3-point functions
  two-dimensional parallel space, choose $q_1^\mu = (e_1, q_z, \ldots)$, $q_2^\mu = (e_2, -q_z, \ldots)$, $D-2$-dimensional orthogonal space

$$l^\mu = (l^0, l^1, \vec{l}_\perp), \quad k^\mu = (k^0, k^1, \vec{k}_\perp)$$

$$\int d^4l \int d^4k = V \int_{-\infty}^{+\infty} dl^0 \int_{-\infty}^{+\infty} dk^0 \int_{-\infty}^{+\infty} dl^1 \int_{-\infty}^{+\infty} dk^1 \times \int_0^{+\infty} k_\perp dk_\perp \int_0^{+\infty} l_\perp dl_\perp \int_{-1}^{+1} \frac{dz}{\sqrt{1-z^2}}$$

  with $z = \angle(\vec{l}_\perp, \vec{k}_\perp)$

- tensor reduction

  cancel components of loop momenta in the numerator against propagators, partial fractioning

- UV-divergences

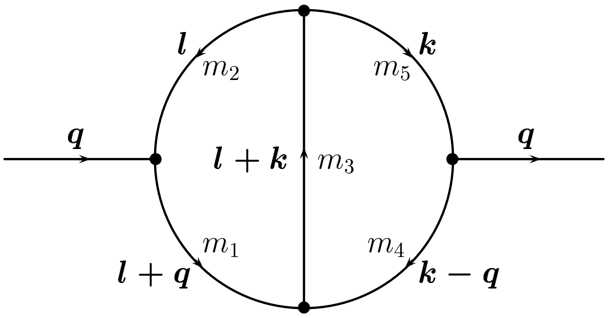
  isolated by subtraction terms

- analytical integrations

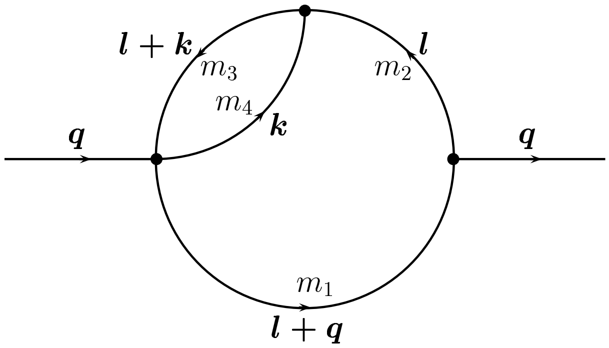
  Cauchy's theorem, elementary integrations (orthogonal space variables)

- one-, or two-parameter integrals over finite domains for numerical integration
  (using `pvegas` or `parint`)

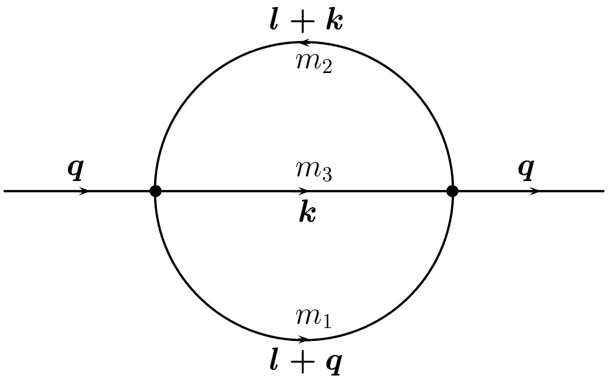
2-loop 2-point topologies



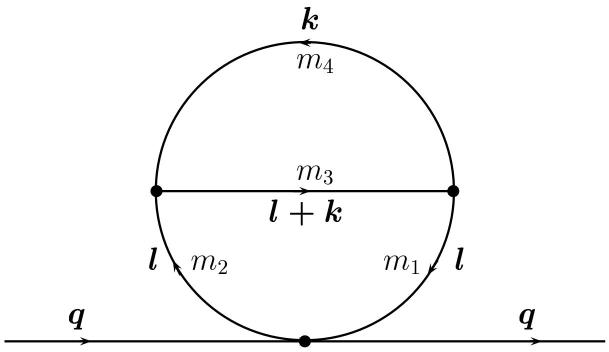
(a)



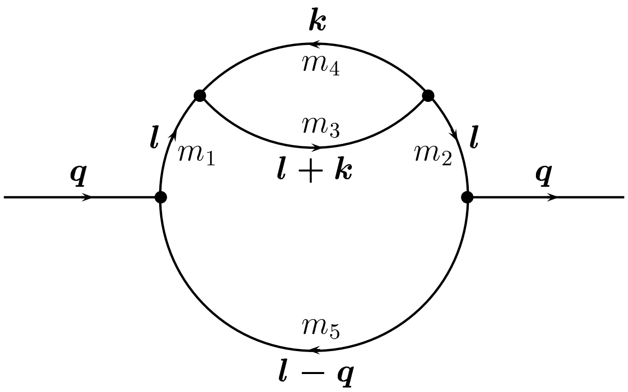
(b)



(c)



(d)



+ factorizing topologies

Example I

one-loop two-point tensor integral

$$\int d^D l \frac{l^{\mu_1} l^{\mu_2}}{[(l+q)^2 - m_1^2 + i\rho][l^2 - m_2^2 + i\rho]}$$

code

```
#include <iostream>
#include "xloops.h"
using namespace std;
using namespace GiNaC;
using namespace xloops;

int main()
{
  symbol  q("q"), m1("m1"), m2("m2"), eps("eps"), rho("rho");
  lst C;
  ex T = OneLoopTens2Pt(2, q, m1, m2, 1, 1, rho, eps, C);
  cout << "Tensor structure: " << T << endl;
  cout << "Coefficients: " << C << endl;
  return 0;
}
```

output

```
Tensor structure: q~mu1*q~mu2*C20 + eta~mu1~mu2*C21
Coefficients: { C21 == -(3-2*eps)^(-1)
                            *OneLoop2Pt(0,2,q,m1,m2,1,1,rho,eps),
                C20 == (3-2*eps)^(-1)*q^(-2)
                          *OneLoop2Pt(0,2,q,m1,m2,1,1,rho,eps)
                        + q^(-2)*OneLoop2Pt(2,0,q,m1,m2,1,1,rho,eps)
              }
```

calculate coefficient

```
ex res = ((C.op(1)).op(1)).subs(m1==m2);
cout << "Coefficient C20: " << endl;
cout << ((res.series(eps==0,2))).normal() << endl;
```

```
Coefficient C20:
(-(1/12*I)*Pi^2*q^2+(1/2*I)*Pi^2*m2^2)*eps^(-1)
+ (-(1/2*I)*Pi^2*m2^2*log(Pi)+1/12*Pi^3*q^2
  -(1/6*I)*Pi^2*m2^2*log(m2^2)
  -(1/2*I)*Pi^2*m2^2*Euler+(1/12*I)*Pi^2*Euler*q^2-(2/9*I)*Pi^2*q^2
  +(7/6*I)*Pi^2*m2^2-1/3*Pi^3*m2^2+(1/12*I)*Pi^2*log(Pi)*q^2
  -(1/12*I)*Pi^2*R2ex1(I*rho-m2^2,-1/4*q^2)*q^2
  +(1/3*I)*Pi^2*m2^2*R2ex1(I*rho-m2^2,-1/4*q^2))
+ Order(eps)
```

Example II: sunrise two-loop integral

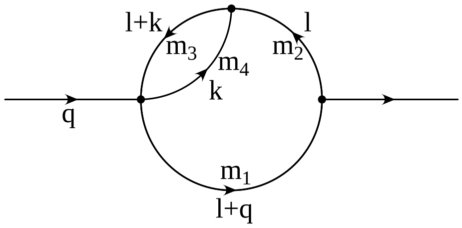
$$\int d^D l d^D k \frac{1}{[(l+q)^2 - m_1^2][(l+k)^2 - m_2^2][k^2 - m_3^2]}$$

code

```
#include <iostream>
#include "xloops.h"
using namespace std;
using namespace GiNaC;
using namespace xloops;

int main()
{
  symbol m1("m1"), m2("m2"), m3("m3"), q("q"), eps("eps"), rho("rho");
  symbol x("x[0]"), y("x[1]");
  lst NumParm(x, y), m(m1, m2, m3), t(1,1,1);

  ex c = TwoLoop2Pt4(0,0, 0, 0, 0, q, m, t , rho, eps, NumParm);

  ex UV = (c.op(1)).series(eps==0,4);
  ex res1 = normal(UV.coeff(eps,-2))*pow(eps,-2)
            + normal((UV.coeff(eps,-1)).subs(rho==0))*pow(eps,-1);
  ex temp = (UV.coeff(eps,0)).subs(lst(q == 175, m1==175, m2==175,
                                       m3==175, rho==pow(10,-15)));

  ex finite = (c.op(0)).subs(lst(q == 175, m1==175, m2==175,
                                 m3==175, rho==pow(10,-15)));

  ex res = VNumInt(finite, 10000, 5, 100000, 5, NumParm);

  cout << "The divergent part is: " << endl;
  cout << res1 << endl;
  cout << "Finite part at pseudo-threshold: "
       << (res.op(0)+res.op(2)*I+temp).evalf()  << endl;
  cout << "with errors (real and imaginary part): "
       <<  "+/-" << abs(res.op(1)) << "and +/-"
       << abs(res.op(3))*I << endl;
}
```

output

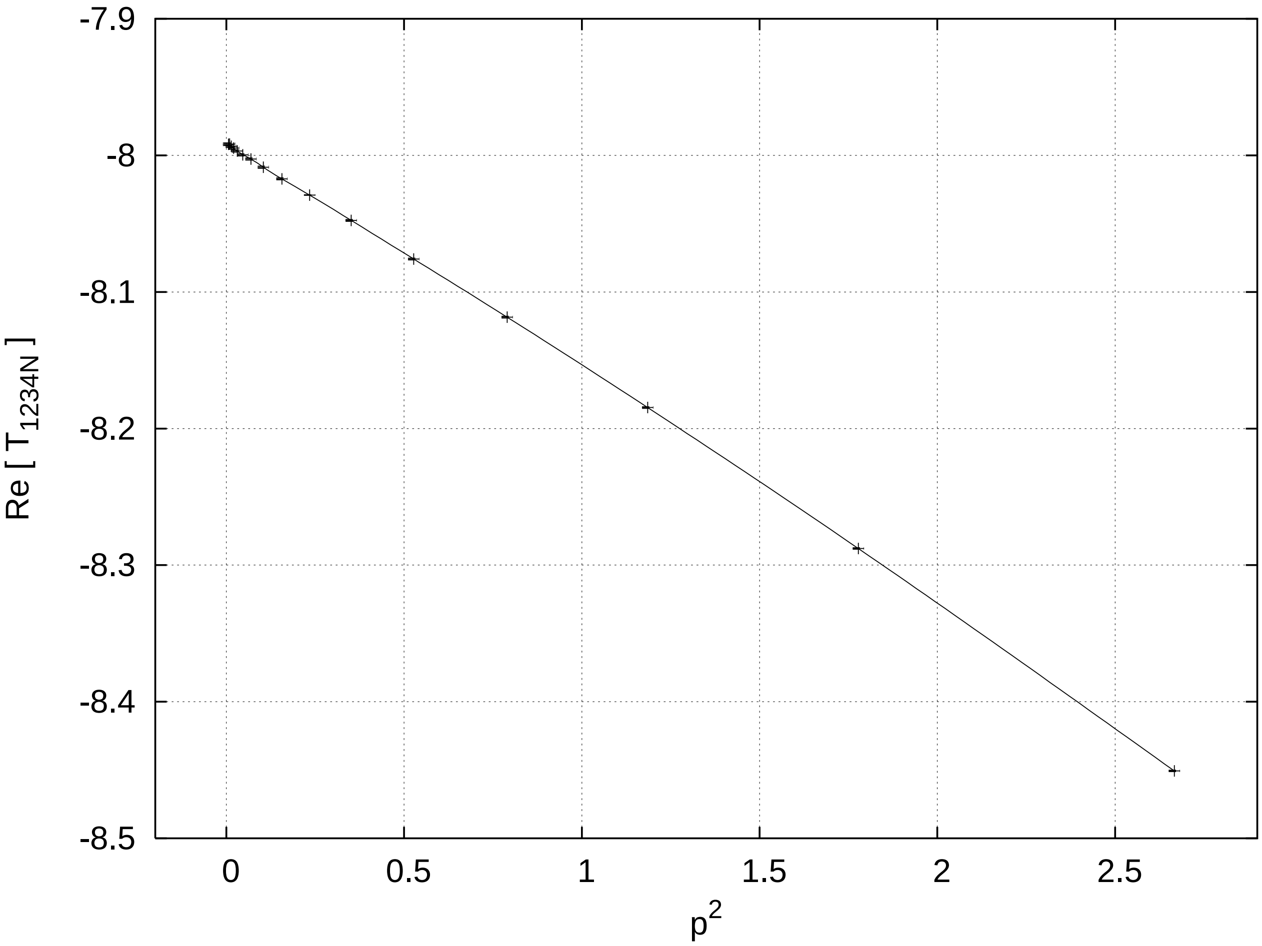
```
The divergent part is:

(-1/2*Pi^4*m2^2-1/2*Pi^4*m1^2-1/2*Pi^4*m3^2)*eps^(-2)+
(-3/2*Pi^4*m2^2+Pi^4*m3^2*log(Pi)+Pi^4*log(m1^2)*m1^2
 -3/2*Pi^4*m1^2+Pi^4*m3^2*log(m3^2)+1/4*Pi^4*q^2
 +Pi^4*m2^2*log(m2^2)+Pi^4*log(Pi)*m2^2+Pi^4*m3^2*Euler
 +Pi^4*Euler*m1^2-3/2*Pi^4*m3^2+Pi^4*log(Pi)*m1^2
 +Pi^4*m2^2*Euler)*eps^(-1)

Finite part at pseudo-threshold: -1.0772268158E9+51943.5*I

with errors (real and imaginary part): +/-136.3 and +/-165.7*I
```

Example: topology 2 $\quad T_{1234} =$ 

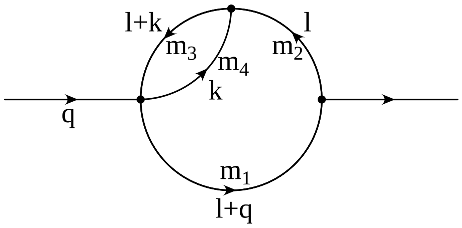
scalar subtracted

$T_{1234N} = T_{1234}(p^2, m_1^2, m_2^2, m_3^2, m_4^2) - T_{1234}(p^2, m_1^2, m_2^2, 0, 0)$

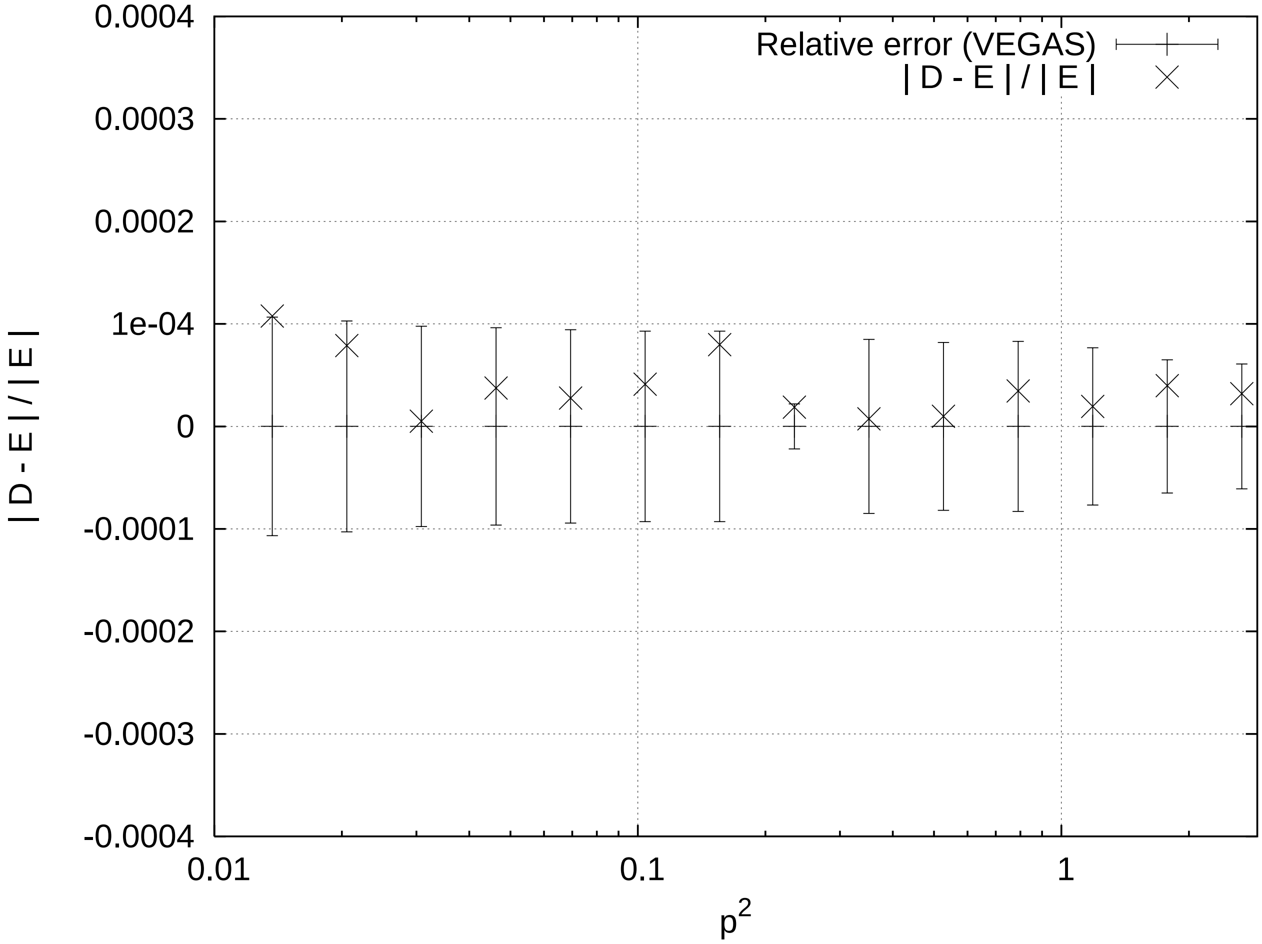
using $m_1 = 1$, $m_2 = 3$, $m_3 = 5$, $m_4 = 7$ GeV



Example: topology 2

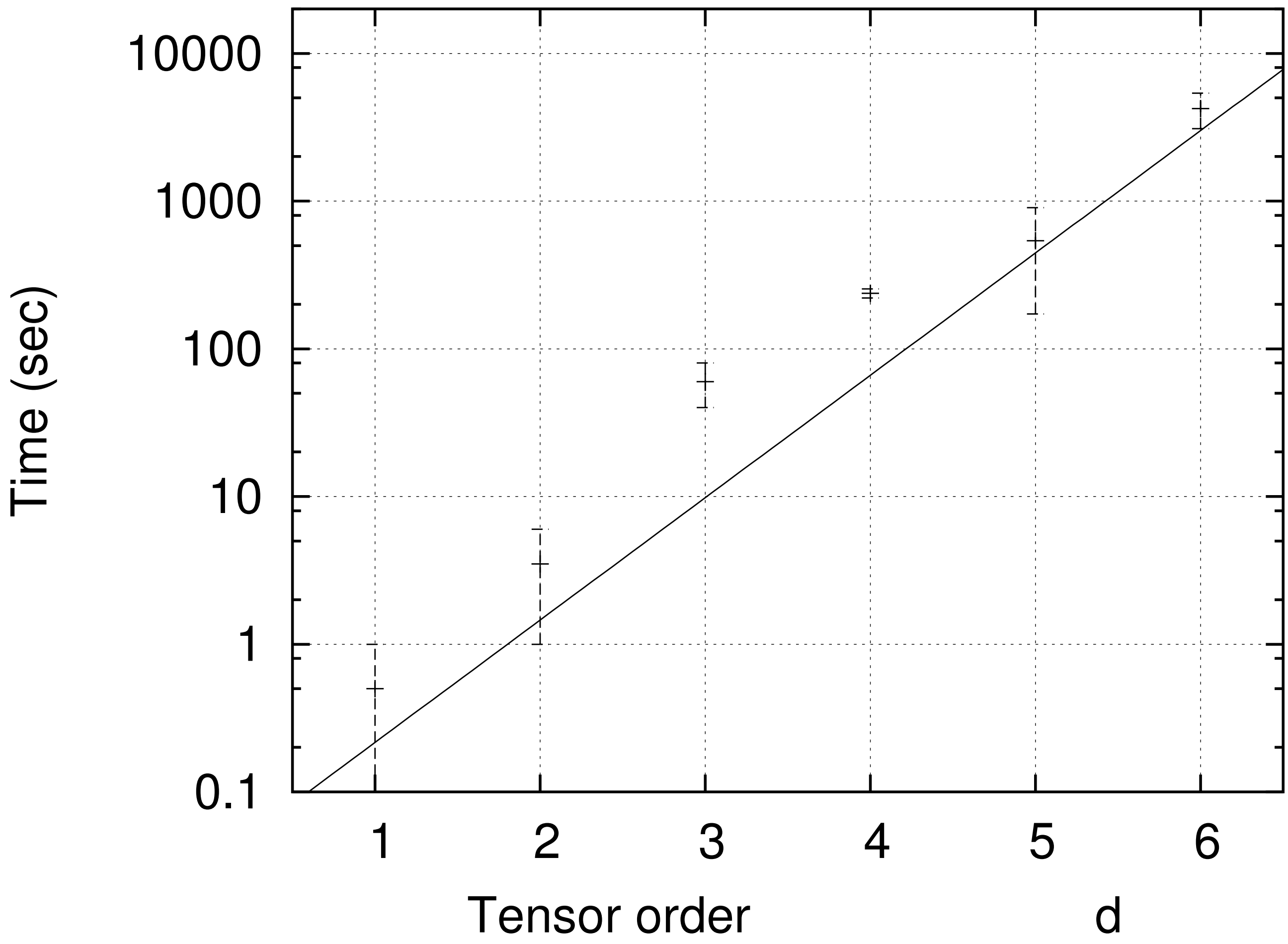


comparison with Bauberger et al.



Performance

two-loop master integral: tensor degree



Further information:

- GiNaC homepage http://www.ginac.de (download, developer's reference, FAQ, mailing lists)
- GiNaC Tutorial http://www.ginac.de/tutorial/
- Theses by C. Bauer, A. Frink, and R. Kreckel http://wwwthep.physik.uni-mainz.de/
  $\rightarrow$ Publication Database $\rightarrow$ diploma and doctoral theses