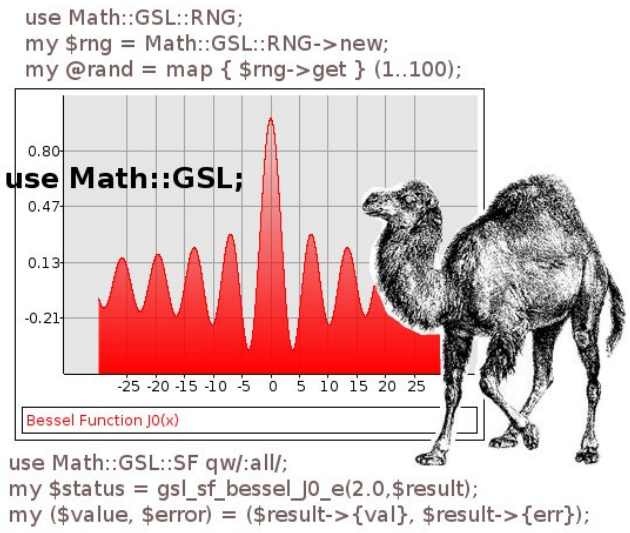
Scientific Computing With Perl and Math::GSL

Jonathan Leto - jonathan@leto.net



Why do "Science" with Perl?

- Minimize development time
- CPAN
- Data Munging
- Camels are cool, FORTRAN isn't

Basic Tools

- PDL - Perl Data Language
  - Basic datatype is an N-dimensional matrix
  - Very fast
  - Low level
  - Overwhelming for a beginner
- Math::GSL - Interface to the GNU Scientific Library
  - 422 special functions, hundreds of statistical distributions, descriptive statistics, splines, root solving, random and quasi-random number generators, FFT, numerical derivative/integration, etc...
  - Very fast
  - Needs more syntax sugar

Random Number Generators

Math::GSL currently has support for 64 different random number generators.

```
my $rng  = Math::GSL::RNG->new;
my @nums = map { $rng->get } (1 .. 1000);
```

```
my $rng  = Math::GSL::RNG
             ->new($gsl_rng_knuthran, 31337);
while ( my $num = $rng->get ){
    ...
}
```

Vectors and Matrices

```
my $v = Math::GSL::Vector->new([1  .. 10]);
my $w = Math::GSL::Vector->new([10 .. 20]);
my $dot_product = $v * $w;
my $scaled_v = 5 * $v;
my ($min,$max) = ($v->min,$v->max);
```

```
my $matrix  = Math::GSL::Matrix->new(5,5);
$matrix->set_col(0, [1 .. 5 ])
       ->set_row(2, [6 .. 10]);
my $first_row = $matrix->row(0);
print "First row is: " .
    join(' ',$first_row->as_list) . "\n";
```

Numerical Derivatives and Integration

$$\frac{d}{dx}\left(x^3\right) = 3x^2$$

```
($status, $result) = gsl_deriv_central ( sub { $_[0] ** 3 }, $x, $h,);
my ($value,$error) = @$result;
```

$$\int_0^1 x^\alpha \log(1/x) dx = \frac{1}{(\alpha+1)^2}$$

When $\alpha = 2$, this reduces to $\frac{1}{9}$.

```
my ($status, $result, $abserr) = gsl_integration_qags (
                                 sub { my $x=shift; $x ** 2 * log (1/$x) } ,
                                 0, 1, 0, 1e-7, 1000,
                                 $gsl_integration_workspace,
                             );
my ($value,$error) = @$result;
```

My Background in Scientific Computing

- Undergraduate research in Fluid Mechanics
- Required numerical solution of nonlinear boundary value equation

$$v'' + \frac{v'}{r} + \frac{v}{r^2} + \epsilon \left(v' - \frac{v}{r} \right)^2 \left(6v'' - \frac{2v}{r} + \frac{2v}{r^2} \right) = 0$$

$v(1) = R$, $v(\omega) = R$ for various R and ω.

- The Runge-Kutta integrator that I wrote in Perl to solve this later became Math::ODE.

But really, what's inside?

Math::GSL::BLAS Math::GSL::BSpline Math::GSL::CBLAS Math::GSL::CDF Math::GSL::Chebyshev

Math::GSL::Combination Math::GSL::Complex Math::GSL::Const Math::GSL::DHT

Math::GSL::Deriv Math::GSL::Eigen Math::GSL::Errno Math::GSL::FFT Math::GSL::Fit

Math::GSL::Heapsort Math::GSL::Histogram Math::GSL::Histogram2D Math::GSL::Integration

Math::GSL::Interp Math::GSL::Linalg Math::GSL::Machine Math::GSL::Matrix Math::GSL::Min

Math::GSL::Monte Math::GSL::Multifit Math::GSL::Multimin Math::GSL::Multiroots

Math::GSL::NTuple Math::GSL::ODEIV Math::GSL::Permutation Math::GSL::Poly Math::GSL::PowInt

Math::GSL::QRNG Math::GSL::RNG Math::GSL::Randist Math::GSL::Roots Math::GSL::SF

Math::GSL::Siman Math::GSL::Sort Math::GSL::Spline Math::GSL::Statistics Math::GSL::Sum

Math::GSL::Sys Math::GSL::Vector Math::GSL::Wavelet Math::GSL::Wavelet2D

Scientific Computing Environment

- Computer Algebra System - gsl_repl and Math::*
- Numerics - Math::GSL and PDL
- Visualization - Cairo, Chart::Clicker, ?
- Data Assistant - ?

Active Development Continues

- Darwin support
- Scientific Computing applications built on top of Math::GSL
- Math::Symbolic integration
- PDL integration
- Callbacks and threaded Perls

Thanks

- Eric Wilhelm
- Thierry Moisan
- #pdx.pm

More Info

- http://leto.net/gitweb/
- http://leto.net/code/Math-GSL/
- http://groups.google.com/group/math-gsl-dev
- http://groups.google.com/group/perl-scientific-computing