ParInt 1.3

Kendrick Cline
Lawrence Cuneaz
Dillon Daudert
Austin Jones

ParInt Group 2016
The Team

Kendrick Cline  Dillon Daudert  Lawrence Cuneaz  Austin Jones
Outline

- Background
- Project Needs
- ParInt
- Design Decisions
  - Compiling
  - Random Number Generators
  - Test Pack
- Maintenance & Security
Background

- ParInt - Parallel Integration
- A numeric integration package for computing single and multidimensional integrals
- One of the best in the world
- Original Project Leads: Elise de Doncker and dozens of faculty and students at WMU with a few external collaborators
ParInt Needs

- Automated testing
- Removing support for obsolete environments
- Survey current cluster systems
- Place a layer in ParInt to use other RNG's
ParInt - How it works

- Approximates single and multidimensional integrals in parallel
- Interprocess communication via the Message Passing Interface (MPI)
- Adaptive domain partitioning and load balancing
- 16 Integration methods
ParInt - How it works

I ask ParInt for an answer to a degree of error (E)

Work for integration gets subdivided

Integral and error estimates are kept in a heap
ParInt - How it works

This portion of the tree has higher error values, therefore more work needs to be done.
ParInt - Integration Methods

● Fixed-point
  ○ Uses fixed integration points to evaluate integral
  ○ Can provide exact solution for low-degree polynomials

● Stochastic
  ○ Generates large numbers of random points to evaluate integral
  ○ Work well for high dimensionality (d >= 8)
## Integration Rules

- **16 Methods**
- **Specialized and general purpose rules**

<table>
<thead>
<tr>
<th>Rule#</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PI_IRULE_DIM2_DEG13</td>
<td>2-dimensional degree 13 rule for rectangular regions using 65 evaluation points</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PI_IRULE_DEG9_OSCIL</td>
<td>Degree 9 rule for n-dimensional hyper-rectangular regions, especially suited for oscillatory integrands.</td>
</tr>
<tr>
<td>4</td>
<td>PI_IRULE-DEG7</td>
<td>Degree 7 rule, recommended general purpose rule for n-dimensional hyper rectangular regions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PI_IRULE_MC</td>
<td>Simple Monte Carlo method</td>
</tr>
</tbody>
</table>
Fixed Point Methods

- Evaluates fixed points along the curve
- Approximates the integrand by fitting a curve to these fixed points
- Adds the area under these new curves to approximate the integral

Simpson’s Rule - an integration method using quadratic curves

<table>
<thead>
<tr>
<th></th>
<th>PL_IRULE_DIM2_DEG13</th>
<th>2-dimensional degree 13 rule for rectangular regions using 65 evaluation points</th>
</tr>
</thead>
</table>
Stochastic (Randomly Determined) Methods

Monte Carlo Method

- Randomly generates points in the region
- Evaluates the function at the points generated under the curve
- Averages these rectangular regions, approximating the integral
- Converges slowly to the integral as $n$ increases
Parallelization with the Rules

- **User Specifies:**
  - Max # of function evaluations
  - Error Tolerance

- **Each Rule Returns:**
  - Estimated Integral
  - Estimated Error

- **Divides region and repeats until one of:**
  - estimated error < error tolerance
  - The Max evaluation limit is surpassed

Simpson’s rule with 1 region, high error

Simpson’s rule with two regions, much lower error
Demonstration

\[ \int_0^1 \int_0^1 \int_0^1 \frac{1}{(x+y+z)^2} \, dz \, dy \, dx = \log \left( \frac{64}{27} \right) \approx 0.863046 \]

ParInt Demonstration
Project objective: Update ParInt to be flexible, user friendly, and well tested
Design Decisions

1. Configuration
2. Random Number Generators
3. Test Suite
Configuration

- Learn how Autoconf is deployed within ParInt
- Correct configuration bugs
  - ParInt and SPRNG
- Investigate target architectures for distributed jobs
- Test ParInt on chosen architectures
Random Number Generators

- Stochastic methods need a good parallel RNG
- Scalable Parallel Random Number Generators (SPRNG)
  - "Deprecated" C interface
  - Newest versions C++/Fortran only
- Don’t want to be dependent on a specific library...
Other PRNG Libraries

- **Generic Interface**
  - Allow any RNG to be “plugged in” to ParInt
  - Gives power and flexibility to user
  - Avoid future incompatibility issues

- **Random123**
  - Header files - fewer compilation concerns
  - Excellent statistical properties
  - Parallelizes well
Test Suite

Goals

- Installation Checking (System Admins)
- Regression Testing (Developers)
- New Test Integrands (Scientists)
Target Integrands & Data Report Overview

- Simple Integrands (Installation Test)
  - Set dimension & parameter
  - Absolute and Relative Error

- Family Integrands (Regression Test)
  - Variable dimension & Parameters
  - Confidence intervals of 20 samples
  - Correct and wrong digits
Simple Integrands

1 dimensional integrand

\[ \int_0^1 \left| x - \frac{1}{3} \right|^a \, dx = \frac{\frac{2a+1}{3} + \frac{1a+1}{3}}{a+1} \]

2 dimensional integrand

\[ \int_{-1}^1 \int_{-1}^1 \frac{1}{\sqrt{3 - x - 2y}} \, dx \, dy = 2.57901 \]

Asymptotic singularity at (1, 1)

Set parameter
Asymptotic Singularity Effect

\[ \int_{-1}^{1} \frac{1}{\sqrt{1-x}} \, dx = 4\sqrt{2} \]

Singularity at \( x = 1 \)
Family Integrands

- Genz Integrand Families
- N-dimensional
- Affective and Un-affective parameters
- Exact Integrals are computable

Oscillatory

\[ f_1(x) = \cos(2\pi u_1 + \sum_{i=1}^{n} a_i x_i) \]

Product Peak

\[ f_2(x) = \prod_{i=1}^{n} \frac{1}{a_i^{-2} + (x_i - u_i)^2} \]

Discontinuous

\[ f_3(x) = \frac{1}{(1 + \sum_{i=1}^{n} a_i x_i)^{- (n+1)}} \]
Family Integrands (cont.)

- Affective parameters
  - Change difficulty level
  - Affect the magnitude of the integrand family trait

- Un-affective parameters
  - no effect on integration difficulty
  - Just a displacement parameter.
Family Integrands (cont.)

Un-Affective Parameter

$$\cos(2\pi u_1 + a_i x_1)$$

Affective Parameter

$$\cos(2\pi u_1 + a_i x_1)$$
Data Report (Simple Integrands)

**fcn32 with rule #4**

- **ACTABS**: 0.000013168516796
- **ACTVAL**: 0.000001287511147
- **RESULT**: 1.44560279429824E-05
- **ESTABS**: 0.555 (Relative Error 3.84E+04)
- **STATUS**: Fcn count: 4000005; Rgn count: 38835; Fcn count flag: 1
  - Time: 0.359; Time/IM: 0.08984; Time/Region: 0.00000925

**Test Failed**

- **Total Time**: 15.829761 seconds
- **Total Passed Tests**: 7
- **Total Okay Tests**: 19
- **Total Failed Tests**: 5

**System Information**

Linux version 2.6.32-358.23.2.el6.647g0000.x86_64 (root@-home-1gu-build_cw6-olaf-
-RHEL-6ES-4-x86_64.localhost.localdomain) (gcc version 4.4.7 20120313 (Red Hat 4.4.7-3) (GCC) ) #1 SMP Thu Oct 17 16:56:53 EDT 2013
Data Report (Family Integrands)

<table>
<thead>
<tr>
<th>Variables</th>
<th>Integrand</th>
<th>Correct digits</th>
<th>Reliability</th>
<th>Wrong Digits</th>
<th>Integrand Values</th>
<th>Quality</th>
<th>Total Fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Oscillatory</td>
<td>1.2</td>
<td>2.8</td>
<td>7.1</td>
<td>8.6</td>
<td>1.00</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Product Peak</td>
<td>4.6</td>
<td>5.0</td>
<td>6.3</td>
<td>7.1</td>
<td>1.00</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Corner Peak</td>
<td>6.0</td>
<td>6.0</td>
<td>9.5</td>
<td>9.7</td>
<td>1.00</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian</td>
<td>6.0</td>
<td>6.0</td>
<td>6.7</td>
<td>7.7</td>
<td>0.90</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>C0 Function</td>
<td>5.2</td>
<td>5.9</td>
<td>5.6</td>
<td>6.9</td>
<td>0.65</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Discontinuous</td>
<td>6.0</td>
<td>6.0</td>
<td>3.4</td>
<td>6.6</td>
<td>0.40</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>Medians</td>
<td>5.6</td>
<td>5.9</td>
<td>6.5</td>
<td>7.4</td>
<td>0.95</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Maintenance & Security

- Instructions on how to build, configure, and compile
- Directions on how to add Random Number Generators
- Directions on how to add new integrand functions
- Security is not a factor in this project
Thank You