Recent Work on Computing Lorentzian Spin Foams

Joshua L. Willis
Abilene Christian University

26 June 2007
Outline of Talk

- Review computational challenges for Lorentzian spin foam models
- Summary of existing method for the tetrahedral network (6J)
- Recoupling Theory for SL(2,C)
- The analogue of the Christensen-Egan algorithm for the Lorentzian 10J
- Outlook for numerical implementation
Spin Foam Models of Quantum Gravity

- Assign partition function to 2-complexes in spacetime

\[ Z = \sum_{\text{colorings}} \prod_{\text{faces}} A_F(f) \prod_{\text{edges}} A_E(e) \prod_{\text{vertices}} A_V(v) \]

- Would like to study numerically to investigate phase structure, semiclassical limit

- Definition involves a sum over labellings of 2-complexes, and possibly over different complexes as well.

- But evaluating summand is computationally hard for just one labeling, because the vertex amplitude \( A_V \) is hard to compute
Why are they computationally hard?

- For Riemannian models (Spin(4) gauge group) efficient algorithm known that re-expresses $10J$ as a sum over $6J$ symbols (Racah coefficients).
- For Lorentzian models (SL(2,C) gauge group) no such efficient algorithm was known; $6J$ symbols themselves are hard.
- Why? They are defined by integrals that are high-dimensional with oscillatory integrands.

\[ 6J = \int_{H^4} \prod_{i=1}^{4} dx_i \ K_{\rho_1}(x_1, x_2) \cdots K_{\rho_6}(x_3, x_4) \]

\[ 10J = \int_{H^5} \prod_{i=1}^{5} dx_i \ K_{\rho_1}(x_1, x_2) \cdots K_{\rho_{10}}(x_4, x_5) \]

\[ K_{\rho}(x, y) = \frac{\sin (\rho r)}{\rho \sinh r} \quad r = d_{\text{hyp}}(x, y) \]
A Better Algorithm for Lorentzian 6J

- Using group-theoretic techniques, can re-express the Lorentzian 6J as a sum of products of Clebsch-Gordan coefficients for \( SL(2,\mathbb{C}) \). Analogous to similar formula for \( SU(2) \) Racah coefficients:

\[
6J \propto \sum_{J}(2J + 1)C^{0\rho_1}_{00}C^{0\rho_5}_{J0}C^{0\rho_4}_{J0}C^{0\rho_6}_{00}C^{0\rho_4}_{J0}C^{0\rho_3}_{J0}C^{0\rho_2}_{00}C^{0\rho_5}_{J0}C^{0\rho_3}_{J0}
\]

- These coefficients can be calculated recursively; thus, very efficiently.

- Much more efficient than direct integration, but convergence can still require many terms.

- Can further speed convergence by using asymptotic form of Clebsch-Gordan coefficients (this is the hard part of both the derivation and coding).
**Tet(1,1,1,1,1,1)**

### Vegas Monte-Carlo Integration

<table>
<thead>
<tr>
<th>Calls</th>
<th>Value</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td>0.041267 ± 21.4%</td>
<td>0.0070</td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.126242 ± 4.03%</td>
<td>0.0430</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.122350 ± 1.53%</td>
<td>0.4309</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.118190 ± 0.490%</td>
<td>4.933</td>
</tr>
<tr>
<td>$10^7$</td>
<td>0.117902 ± 0.192%</td>
<td>69.99</td>
</tr>
<tr>
<td>$10^8$</td>
<td>0.118459 ± 0.0532%</td>
<td>436.6</td>
</tr>
</tbody>
</table>

### Summation Algorithm

<table>
<thead>
<tr>
<th>Terms</th>
<th>Value</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^2$</td>
<td>0.118087292</td>
<td>≈ 0.00002</td>
</tr>
<tr>
<td>$10^3$</td>
<td>0.118283570</td>
<td>≈ 0.0002</td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.118306260</td>
<td>0.002</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.118299794</td>
<td>0.0200</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.118300212</td>
<td>0.198</td>
</tr>
<tr>
<td>$10^7$</td>
<td>0.118300200</td>
<td>1.98</td>
</tr>
<tr>
<td>$10^8$</td>
<td>0.118300196</td>
<td>19.9</td>
</tr>
</tbody>
</table>

### Accelerated Summation Algorithm

<table>
<thead>
<tr>
<th>Terms</th>
<th>Value</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^2$</td>
<td>0.1183001969</td>
<td>≈ 0.0002</td>
</tr>
<tr>
<td>$10^3$</td>
<td>0.1183001969</td>
<td>≈ 0.002</td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.1183001969</td>
<td>0.0170</td>
</tr>
</tbody>
</table>
The reason for calculating the $6J$ is to use it in calculating the $10J$, hoping that this method is more efficient or more accurate than the direct integration.

To do this, we need to use recoupling theory for $SL(2,\mathbb{C})$ in the same way that the Riemannian algorithms rely on recoupling theory for $SU(2)$.

This can be done, and leads to diagrammatic techniques similar to those used for $SU(2)$ spin networks.

Such techniques can be proven using known identities for $SL(2,\mathbb{C})$ matrix elements and Clebsch-Gordan coefficients.
Example: Expanding the 4-Valent Vertex

- All manipulations are based on re-expressing the kernels for Lorentzian spin networks in terms of matrix elements on $\text{SL}(2,\mathbb{C})$.
- Use the identity:

$$D^{0\alpha}_{J\alpha M\alpha 00}(g)D^{0\beta}_{J\beta M\beta 00}(g) = \int \lambda^2 d\lambda \overline{C}^{0\alpha}_0^{0\beta}_0 \sum_{J M} C^{0\alpha}_{J\alpha M\alpha}C^{0\beta}_{J\beta M\beta}D^{0\lambda}_{J M 00}(g)$$

twice to prove the diagrammatic relation:
After suitably renormalizing the 3-valent vertex, can prove recoupling for \( SL(2,\mathbb{C}) \) spin networks.

Note the appearance of a non-simple representation in the recoupling formula: this is unavoidable and an exactly analogous situation occurs in the Riemannian case, when we consider recoupling for \( \text{Spin}(4) \) spin networks.
Evaluating the 10J
A Formula for 10J's in terms of 6J's

- Combining all of these steps we get a formula analogous to the Christensen-Egan algorithm for the Riemannian 10J:

\[ 10J = \int \prod_{i=0}^{4} (\lambda_i^2 d\lambda_i) \sum_{\nu} \int (\mu^2 + 4\nu^2) d\mu \text{ (Prod of CG's)} \]

\[ \left\{ \rho_{20} \lambda_0, \nu, \mu \right\} \left\{ \rho_{21} \lambda_1, \nu, \mu \right\} \left\{ \rho_{22} \lambda_2, \nu, \mu \right\} \left\{ \rho_{23} \lambda_3, \nu, \mu \right\} \left\{ \rho_{24} \lambda_4, \nu, \mu \right\} \]

- Expresses 10J as a six-dimensional integral and one-dimensional sum over 6J symbols.

- Thus, dimension of integral is reduced (from 9 to 6) and experimentation seems to indicate the integrand is in general less oscillatory. When triangle inequalities are violated it decays exponentially.
Numerical Implementation

- This has been implemented, but at present is not as fast as the existing direct integration
  - Need to improve asymptotics in $6J \xrightarrow{\text{faster}} 6J$
  - Use importance sampling in integrals to take advantage of exponential decay.
  - Other methods of evaluating $6J$?
- Hope to test these improvements in next couple of months.
- Thanks to: NSF grant OISE-0401966; Dan Christensen, Wade Cherrington, and Igor Khavkine for discussions.