Numerical Integration Methods in Local Positioning

Niilo SIROLAa, Robert PICHÉ, Henri PESONEN,
Tampere University of Technology, Tampere, Finland

a corresponding author, e-mail: niilo.sirola@tut.fi

Abstract - A Bayesian approach to position estimation requires the evaluation of multidimensional integrals which, except for special cases, must be evaluated numerically. In this paper different numerical integration methods are compared: nodal quadrature with uniform grid, Monte Carlo with random and quasi-random nodes, and adaptive cubature. The methods are used to compute a number of two- and three-dimensional integrals having features typical of “local” positioning. It is found that adaptive cubature methods outperform the other approaches both in terms of accuracy and computation time, at least in low dimensions and when the probability densities are smooth.

1 Local Positioning

One of the challenges of positioning and tracking is that the number and locations of the reference signal sources can be, at best, barely sufficient. Under these circumstances, the estimation algorithm should use all the available information as effectively and efficiently as possible. In local positioning and tracking, which is based on reference signals from nearby sources such as Bluetooth, WLAN, acoustic sensors, or cellular networks, the measurement geometry is strongly non-linear and the measurement errors are strongly non-normal, with multiple modes and curved ridges. These characteristics can seriously degrade the accuracy and reliability of conventional least-squares methods, which are optimal for linear geometry and normal error distributions.

The Bayesian approach[2] offers a general framework for positioning and tracking for arbitrary geometries and probability distributions. Many conventional methods can be formulated as special cases of Bayesian positioning and tracking. A Bayesian position estimate is based on the posterior probability distribution of the position. By Bayes’ theorem, the posterior distribution is proportional to the product of the measurement likelihood function and the prior distribution. A Bayesian model of a typical positioning problem is presented in section 2.

The position estimate can be derived from the posterior probability distribution in different ways. The maximum likelihood estimate is the point at which the posterior distribution is maximum. When the measurement errors have normal distributions, the computation of the maximum likelihood estimate is equivalent to the conventional iterative least squares method. For general multimodal distributions, however, numerically finding the global maximum can be challenging.

An alternative estimate is the mean value of the posterior distribution. In the special case of a unimodal and symmetric posterior distribution, the maximum likelihood estimate coincides with the mean. Apart from some special cases, computing the mean value requires a numerical method to approximate integrals in several dimensions. The challenge is then to find a numerical method to compute these integrals with reasonable speed and accuracy, especially taking into account the limited computing power available in mobile positioning settings.

In section 3, we describe three standard numerical integration methods:
Monte Carlo integration is a popular choice for multidimensional integrals. The samples can be from a random number generator or from a quasi-random sequence (lattice). The error of the approximation can be evaluated using standard variance estimation formulas.

Nodal quadrature with a uniform rectangular grid is a generalization of the trapezoid rule; error can be estimated by combining solutions from different grid refinements.

Adaptive quadrature methods combine sophisticated subregion grid refinement and integration formulas of different orders. Schürer [4] reports that such methods can be competitive with Monte Carlo methods even for problems of up to 100 dimensions.

In section 4, we report on a series of numerical tests of these methods applied to two- and three-dimensional integrals arising in a typical position estimation problem.

The paper closes with a summary of our conclusions and an outline of further investigations.

2 Model

In this work we consider only positioning with range measurements, but any types of measurements with a known error model, such as range difference measurements, can be modeled similarly and incorporated to the model.

The range measurement to the $i$th fixed station can be modeled as

$$ r_i = \| s_i - x \| + \varepsilon_i $$

where $s_i$ is the known station position, $x$ is the unknown user position, and $\varepsilon_i$ is a random variable that represents the measurement error. Denoting the measurement error probability density as $\phi_i$, the measurement likelihood function can be defined as

$$ p(r_i \mid x) = \phi_i(\| s_i - x \| - r_i). $$

This formula can be interpreted as the probability density of the measurement given a particular user position. For example, if the measurement error has normal distribution with zero mean and variance $\sigma_i^2$, then the measurement likelihood would be

$$ p(r_i \mid x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-[(\| s_i - x \| - r_i)/2\sigma_i^2]}. $$

In general, however, the measurement error distribution need not be normal, or even symmetric or zero-mean. In the case of several independent measurements $r_1, \ldots, r_n$, stacked into a vector $r$, we have

$$ p(r \mid x) = \prod_{i=1}^{n} \phi_i(\| s_i - x \| - r_i). $$

Bayes’ formula gives the posterior distribution, modulo a normalising factor, as

$$ p(x \mid r) \propto p(x) p(r \mid x), $$

where $p(x)$ is the prior density for $x$. A position estimate is then given by the mean value of the posterior density,

$$ \hat{x} = \int x p(x) p(r \mid x) \, dx \quad / \quad \int p(x) p(r \mid x) \, dx. \quad (1) $$
Adaptive cubature

Grid

Quasi-Monte Carlo

Monte Carlo

Figure 1: Sample points chosen by different numerical integration methods for $N = 500$.

In the scope of this work, we take the prior to be constant in a given bounded area of interest $\Omega$ and zero outside this area, although other sensible choices for the prior also exist. With this prior the position estimate (1) reduces to

$$\hat{x} = \frac{\int_{\Omega} x p(r \mid x) \, dx}{\int_{\Omega} p(r \mid x) \, dx}.$$ 

It is interesting to note that the Bayesian approach provides a position estimate even when the number of range measurements is too small to determine a unique geometric position fix or least squares position estimate.

3 Integration methods

All the numerical integration methods discussed here can be written in the form

$$\int_{\Omega} f(x) \, dx \approx V \frac{1}{N} \sum_{k=1}^{N} f(x_k)$$

where the number of integration points $N$ and the integration points $x_k$ are either decided beforehand independent of the integrand, or adaptively during the integration. The integration region $\Omega$ is taken to be a multidimensional cube and $V$ denotes the volume of $\Omega$.

In choosing a suitable integration method for this particular application, there are several considerations. As opposed to asymptotic error analysis, we are interested in the accuracy gained with a minimum amount of work. The measurements usually have errors in the order of tens to hundreds of meters in them, so that it does not make sense to look for more than two or three decimals of accuracy. A more important quality of an integration method is to find a coarse solution with as few function evaluations as possible. In addition, the method should come with a robust error estimate, but not an overly pessimistic one.

Figure 1 illustrates how the three considered integration schemes distribute the integration points in the case of a 2-dimensional example integrand. The distribution produced by the plain Monte Carlo is also given for reference.

3.1 Nodal quadrature

Nodal quadrature (“grid method”) is a generalization of the trapezoid rule to multiple dimensions. The integration nodes $x_k$ are chosen to form an equispaced rectangular grid over the integration domain. For irregular regions a triangular or tetrahedral grid may be used. The integral over each element is then approximated as the element area or volume times the average
of the integrand values at the vertices. The error of the nodal quadrature can be estimated by extrapolation of the solutions for two grids of different densities, using the fact that the error is proportional to the square of the element dimension.

3.2 Monte Carlo and Quasi Monte Carlo Integration

In Monte Carlo approach, the integration points are chosen at random. In this application, we choose the points from a uniform distribution over the integration domain.

The variance of the Monte Carlo estimate is

\[ \frac{V^2}{N} \sum_{k=1}^{N} f^2(x_k) - \left( \frac{V^2}{N^2} \sum_{k=1}^{N} f(x_k) \right)^2, \tag{2} \]

which is easy to compute along with the integral. The integration can be run until desired accuracy is reached. The error estimate is only probabilistic, though. A variant of the method is called quasi-Monte Carlo. Quasi-random numbers are a deterministic sequence that has the properties of a uniform random sequence, but the points are “more uniformly” distributed in space. The rightmost plot in Figure 1 illustrates how the uniformly random points can sometimes miss the important regions even in two dimensions.

3.3 Adaptive Cubature

In an adaptive scheme, the integration nodes are concentrated to where the integrand is changing most rapidly. For testing, we used an experimental CUBPACK package[1] that performs adaptive subregion division and provides error estimates.

Figure 1 illustrates how the adaptive method concentrates its effort around a discontinuous edge in the integrand since that is where the integrand is changing most rapidly. This may not be preferable behavior for this application.

4 Numerical testing and results

In the testing phase, we generated a large amount of test cases where range measurements from two stations are received. In addition, linear constraints may be imposed, simulating the effect of sector boundaries.

The measurement errors are assumed normal, and thus the test problems are of the form

\[ p(r \mid x) \propto I_{A,b}(x) \exp \left( -\frac{1}{2} \sum_{i=1}^{2} \frac{(\|s_i - x\| - r_i)^2}{\sigma_i^2} \right) \tag{3} \]

where the indicator function \( I_{A,b}(x) = 1 \) if \( Ax \geq b \) and 0 otherwise.

The integration methods are used to compute the mean value of the integrand over a 1 km by 1 km domain. The integrands in this case may have either one or two peaks in the domain, and depending on the linear constraints, they may have discontinuous edges.

We chose the parameters \( s_i, r_i, \sigma_i, A, \) and \( b \) for each test case such that we get an equal number of each of the three types of test functions:

Type 1 - unimodal continuous
Type 2 - multimodal continuous
Type 3 - unimodal with a discontinuity
Type 4 - multimodal with one or more discontinuities
The measurement errors have standard deviation $\sigma$ between 50 and 150 meters, which could correspond to either good-quality base station measurements or bad-quality satellite measurements (without multipath).

Figure 2: Examples of the different types of test functions.

Figure 2 shows contour plots of some example functions of each type.

For each test case, a reference solution was sought using Cubpack with a maximum of 150 000 integrand evaluations allowed. In most cases, this took the accuracy of the reference solution to the centimeter-level.

Figure 3: Integration errors (solid lines) and error bounds (dashed lines) with different numbers of integration points.

Obviously, performing this many integrand evaluations is hardly practical, when compared to the iterative point position methods that rarely need to evaluate the cost function more than ten times. Figure 3 illustrates the behavior of the methods with a low number of integration points. In the first plot, the integrand is a well-behaved one and all methods settle to a couple of meters accuracy if more than 1000 integration points are used. Cubpack’s error bound comes down rather quickly, but the slack probabilistic error estimates of the other two methods would require a lot more integration points to reach, say, 10 meters.

The second plot in Figure 3 shows an example of a more difficult discontinuous integral. In this plot, it is seen how the quasi-Monte Carlo method produces notably different methods with different amount of integration points although the points are assigned deterministically.
Figure 4: N=500.

Figure 5: N=10000.
Next, we ran a larger test set using 500, 1,000, 5,000, and 10,000 integration points and compared the accuracies. Figures 4 and 5 break down the results for the different integrand types. These plots represent the distribution of the integration errors, and thus the higher a curve is the better. From Figure 4, it is interesting to notice that with 500 integration points the grid method performs best in all cases but the Type 2 (multimodal continuous). The quasi-Monte Carlo methods is clearly worse than the two others in the continuous cases (Types 1 and 2), but in the discontinuous cases it comes second between the grid method and Cubpack. It is worth to note that Cubpack, being a general-purpose package, needs more points than this to be effective.

As can be seen in Figure 5, when 10,000 integration points are used, Cubpack dominates the results in the continuous cases and reaches centimeter accuracy almost certainly. In the discontinuous tests, Cubpack and the grid method perform comparably. The quasi-Monte Carlo seems to give almost an order of magnitude worse results in all problem types.

<table>
<thead>
<tr>
<th>$N = 500$</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubpack</td>
<td>41</td>
<td>77</td>
<td>21</td>
<td>8</td>
<td>35</td>
</tr>
<tr>
<td>Grid</td>
<td>73</td>
<td>24</td>
<td>56</td>
<td>89</td>
<td>61</td>
</tr>
<tr>
<td>Quasi-MC</td>
<td>3</td>
<td>4</td>
<td>23</td>
<td>14</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N = 1000$</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubpack</td>
<td>91</td>
<td>96</td>
<td>59</td>
<td>52</td>
<td>74</td>
</tr>
<tr>
<td>Grid</td>
<td>57</td>
<td>7</td>
<td>32</td>
<td>45</td>
<td>35</td>
</tr>
<tr>
<td>Quasi-MC</td>
<td>10</td>
<td>5</td>
<td>12</td>
<td>14</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N = 5000$</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubpack</td>
<td>100</td>
<td>98</td>
<td>69</td>
<td>69</td>
<td>84</td>
</tr>
<tr>
<td>Grid</td>
<td>83</td>
<td>16</td>
<td>32</td>
<td>35</td>
<td>42</td>
</tr>
<tr>
<td>Quasi-MC</td>
<td>36</td>
<td>31</td>
<td>14</td>
<td>27</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$N = 10000$</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubpack</td>
<td>100</td>
<td>98</td>
<td>55</td>
<td>56</td>
<td>77</td>
</tr>
<tr>
<td>Grid</td>
<td>93</td>
<td>28</td>
<td>48</td>
<td>55</td>
<td>56</td>
</tr>
<tr>
<td>Quasi-MC</td>
<td>54</td>
<td>50</td>
<td>26</td>
<td>40</td>
<td>46</td>
</tr>
</tbody>
</table>

Table 1: How many times (%) each method gave the best answer.

The Table 1 gives another perspective to the comparison. The table shows how often a particular method found the best solution to the problem. The columns do not add to 100% because often more than one method produced the same solution (within a tolerance). Again, the table shows that apart from the case of only 500 integration points allowed, the adaptive Cubpack is most often the most accurate of the tested methods.

4.1 A 3D case

For comparison, we did small-scale testing on three-dimensional test functions also. The 3D test case was formed by adding an “altitude measurement” to the 2D case. The test function was formed by multiplying Eq. (3) by

$$
\exp \left( -\frac{1}{2} \frac{x^2}{\sigma_a^2} \right),
$$

simulating a normally distributed zero-altitude measurement with variance $\sigma_a^2$. 


Figure 6 again shows the superiority of the adaptive method in continuous cases 1 and 2. In the discontinuous cases 3 and 4, the quasi-Monte Carlo method is for the first better than the two others in some regions.

Table 2: How many times (%) each method gave the best answer in 3D.

Table 2 also shows how the quasi-Monte Carlo is more feasible in a higher dimension.

5 Conclusion

It is found that adaptive cubature methods outperform the Monte Carlo approach both in terms of accuracy and computation time, at least in low dimensions and when the measurement error densities are smooth. With discontinuous integrands, however, the advantage of the adaptive cubature is not so obvious. This can, however, be improved in the future by using polyhedral integration domains with the discontinuities placed at the edges of the domain.

At this point, we assumed that the computational load of each method depends only on the number of integration nodes, ignoring the overhead in the algorithms themselves. More detailed
performance analysis would call for a more thorough implementation of the different methods, otherwise we are comparing implementations instead of algorithms.

Future research topics include more detailed error analysis, where the analytical properties of the family of integrand functions is taken into account. Furthermore, the adaptive strategy can also be applied to Monte Carlo type methods, yielding very promising results[3].

**Acknowledgements**

This study was funded by Nokia Corporation. Niilo Sirola acknowledges the financial support of the Nokia Foundation.

**References**


