

Worksheet 3

Practical Lab Numerical Computing

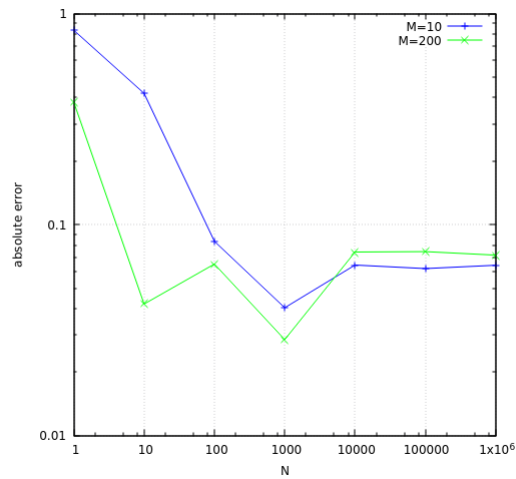
Andrii Lischishin

Lars Schleithoff

Hendrik Kleikamp

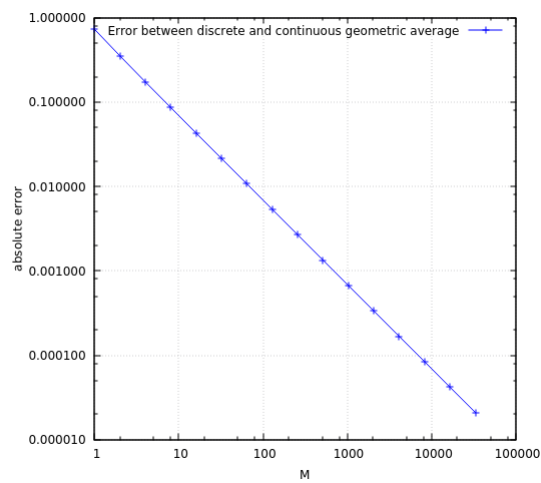
June 20, 2017

Task 3



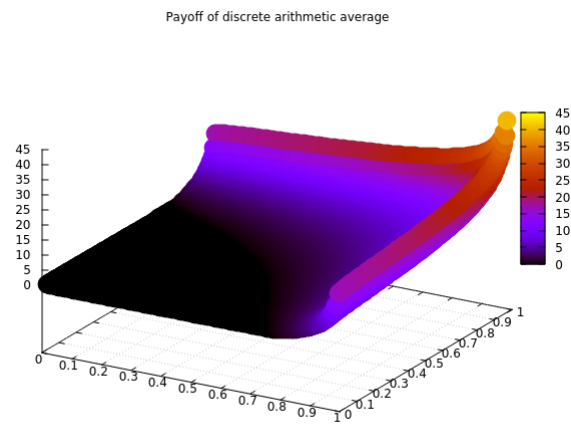
The number of timesteps is not really effecting the convergence.

Task 4



The discrete geometric average converges linearly to the continuous geometric average.

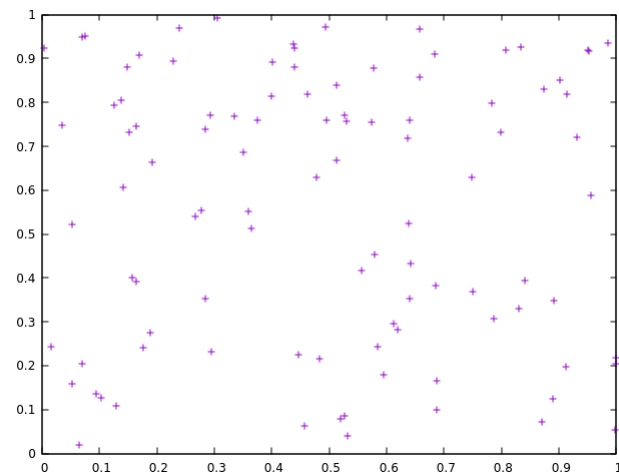
Task 5



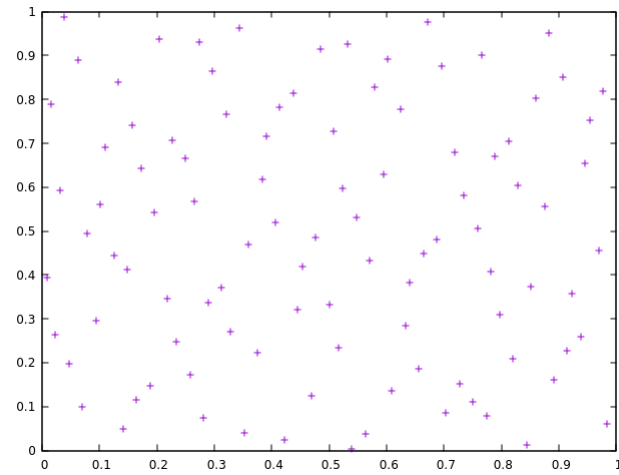
The discrete geometric payoff function is evaluated on 10.000 points in $[0, 1]^2$.

Task 7

Uniform random numbers in $(0, 1)^2$:



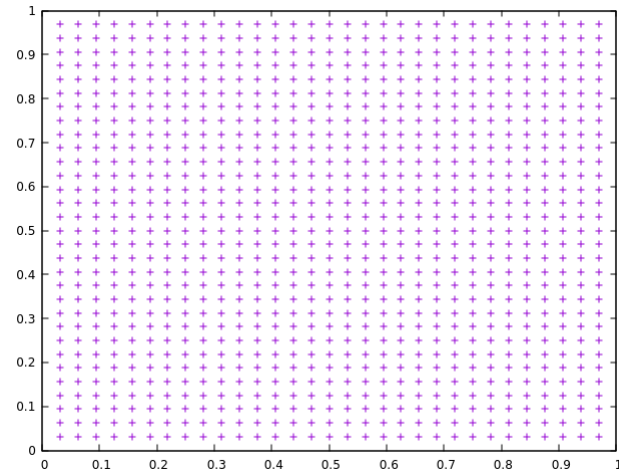
Halton sequence:



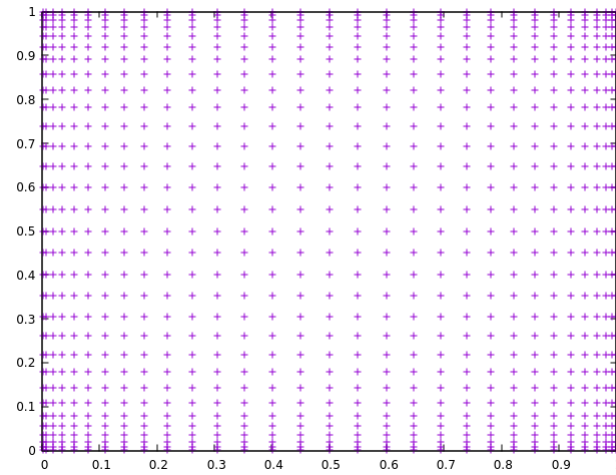
One can see that there are some regions in $(0, 1)^2$ where no uniform random numbers are set. This is not the case in the point set calculated by the Halton sequence. The Halton sequence gives a very uniform spread set without any holes.

Task 9

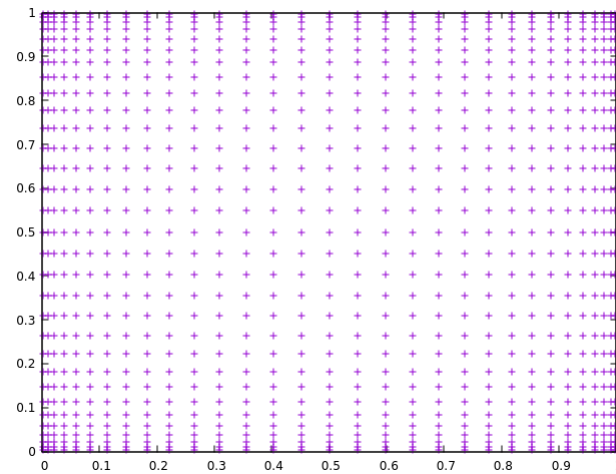
Quadrature nodes of the two-dimensional product rule for trapezoidal rule:



Quadrature nodes of the two-dimensional product rule for Gauss-Legendre:



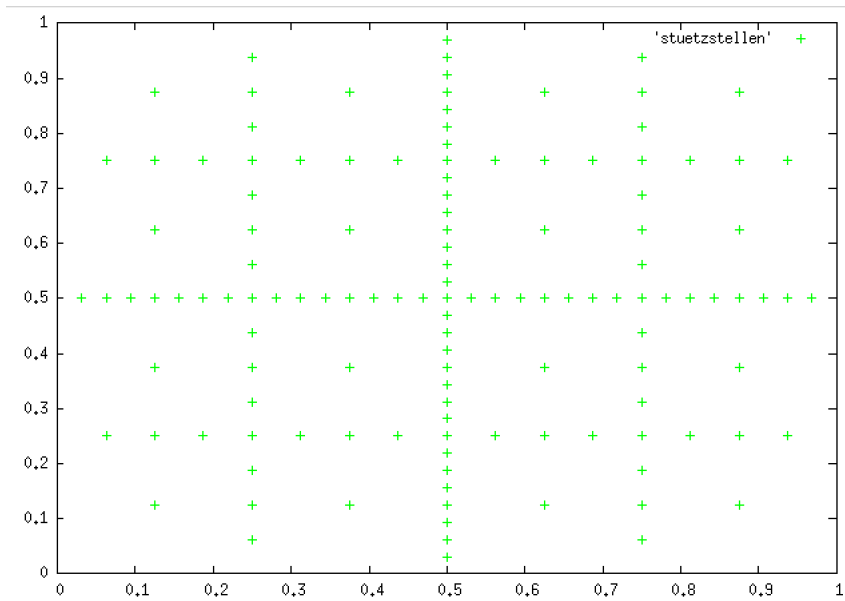
Quadrature nodes of the two-dimensional product rule for Clenshaw Curtis:



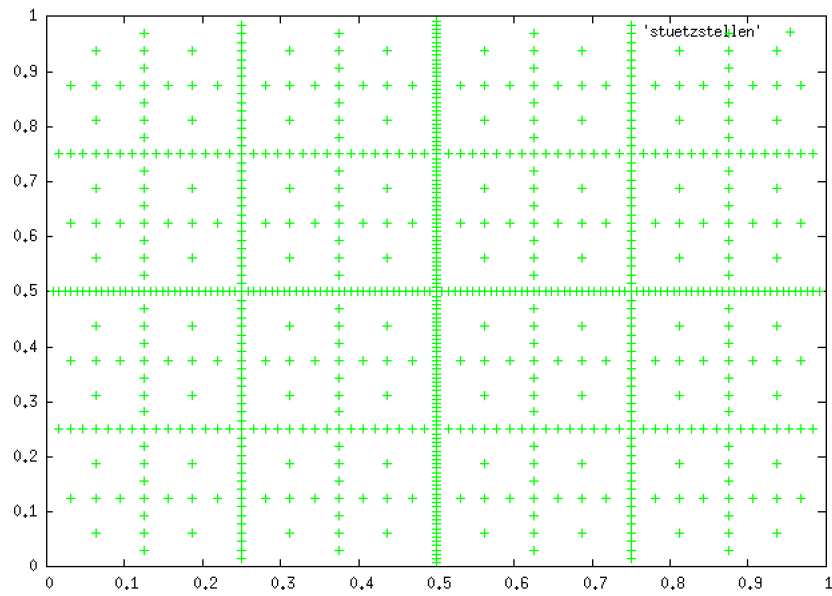
Both Gauß-Quadrature and Clenshaw-Curtis-Quadrature have an increased amount of nodes at the edges of the integration region. The trapeziodal rule distributes the nodes equidistantly over the integration region.

Task 11

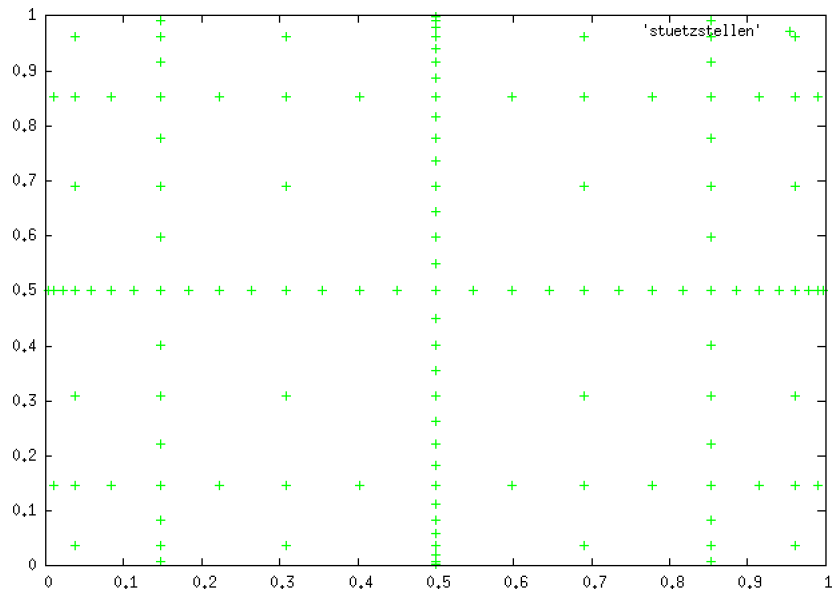
Two-dimensional Sparse Grid using Trapeziodal Rule for $l = 5$:



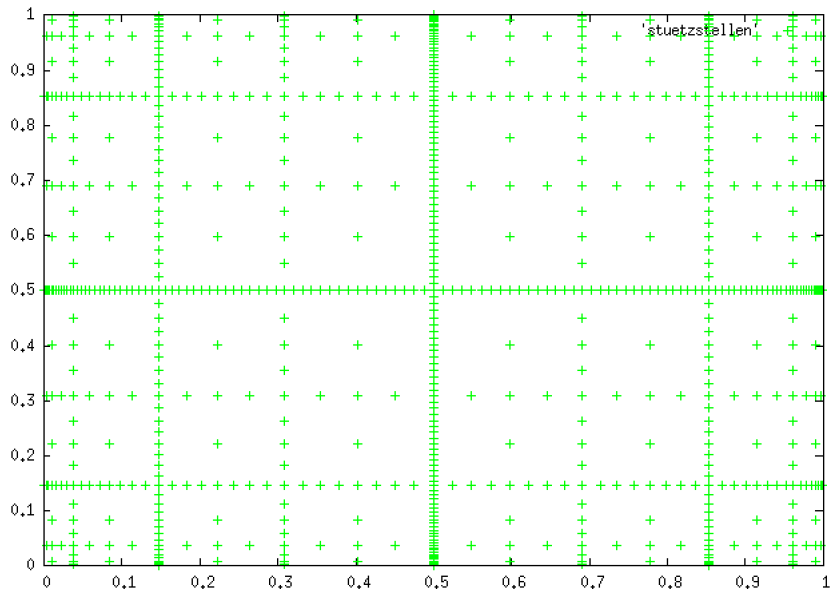
Two-dimensional Sparse Grid using Trapezoidal Rule for $l = 7$:



Two-dimensional Sparse Grid using Clenshaw-Curtis Quadrature Rule for $l = 5$:



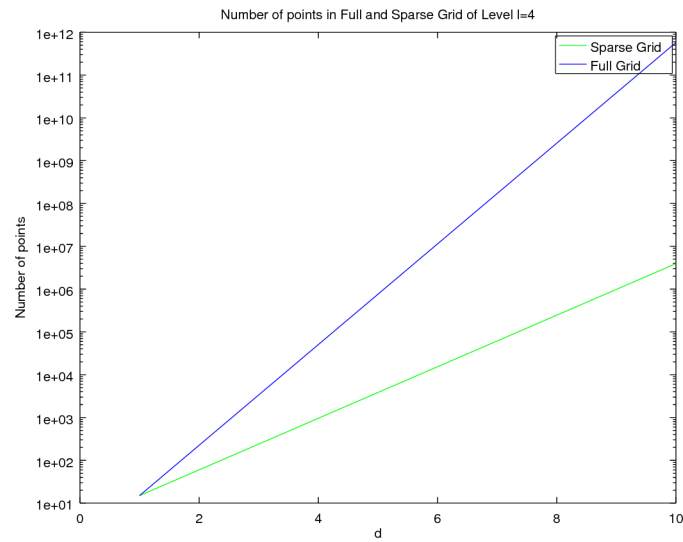
Two-dimensional Sparse Grid using Clenshaw-Curtis Quadrature Rule for $l = 7$:



For Clenshaw-Curtis Quadrature, with increasing level, more nodes are situated at the edges of the integration region. For Trapezoidal Rule, this is not the case.

Task 12

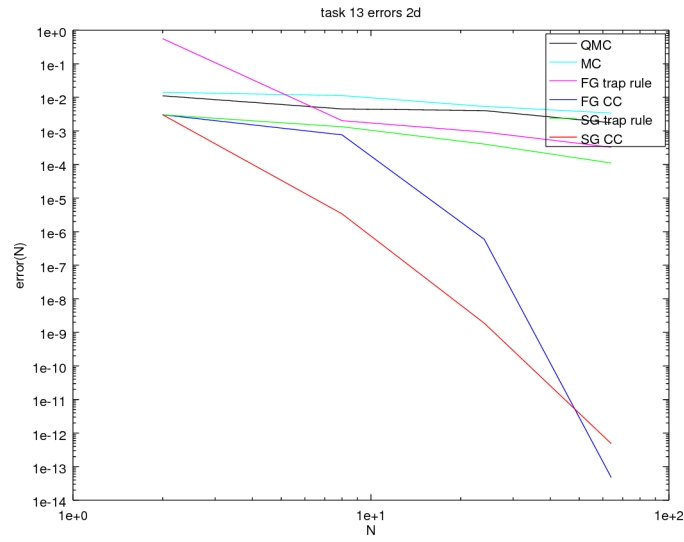
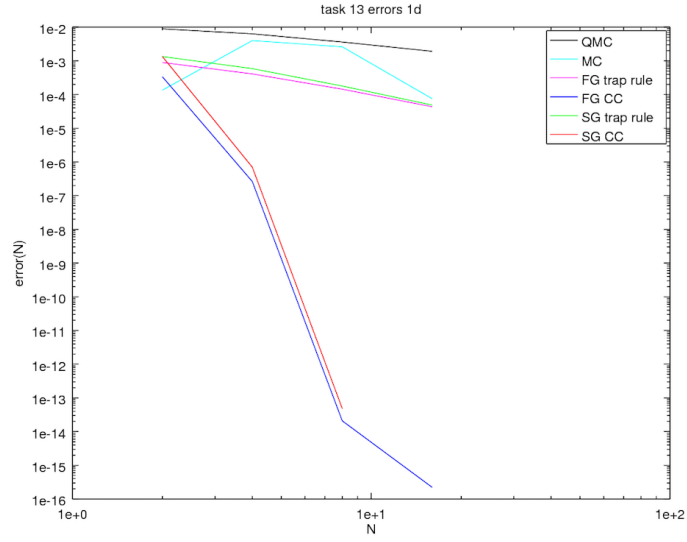
Number of points in Sparse Grid and Full Grid in Level $l = 4$:

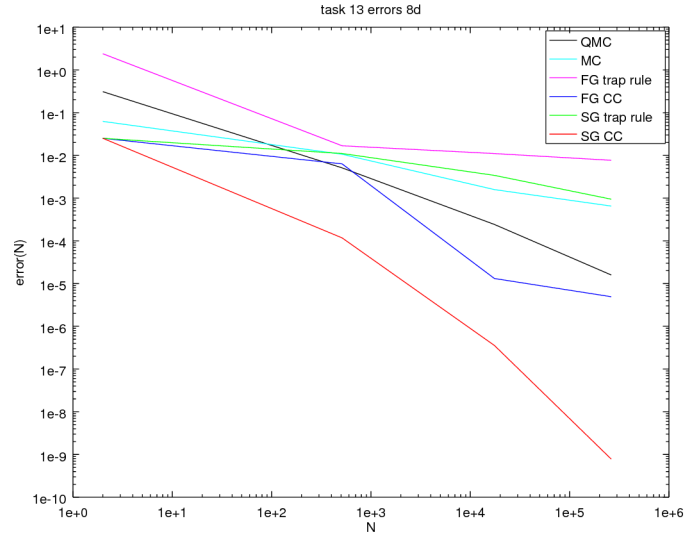
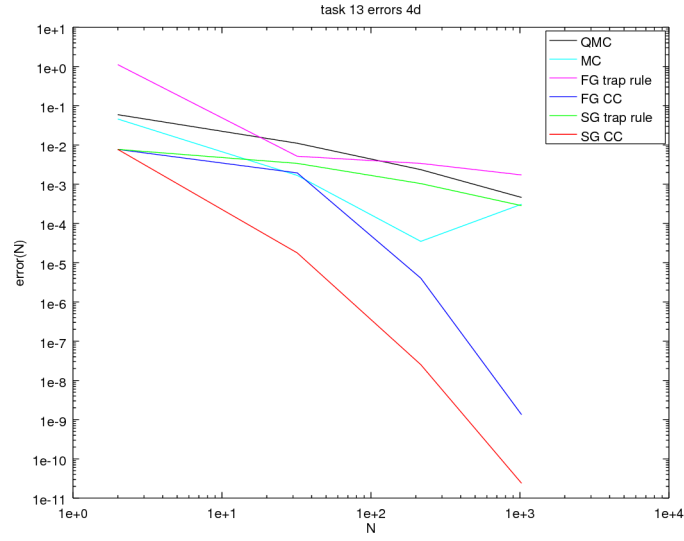


One can see that the number of points increases much faster in the Full Grid than in Sparse Grid.

Task 13

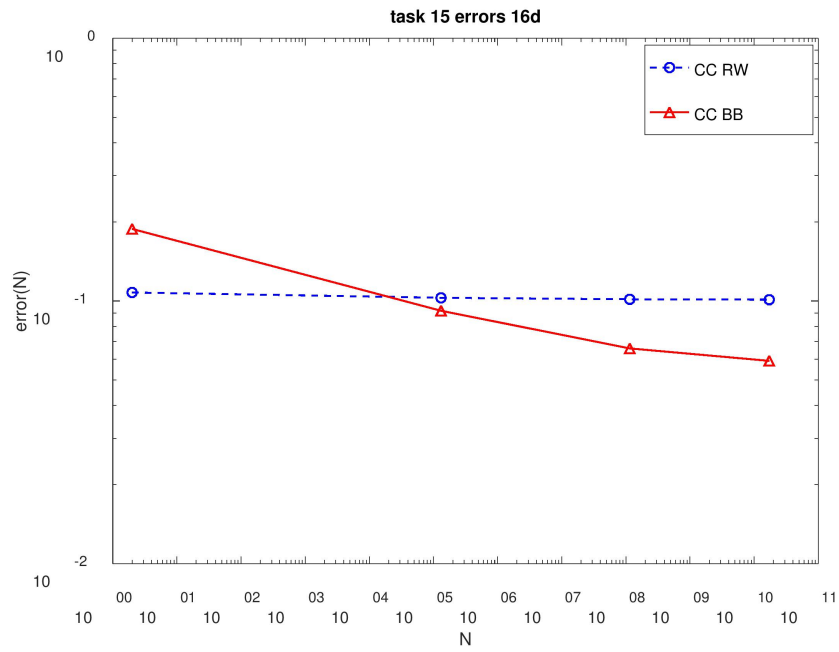
Convergence plots for the testfunction f_γ in different dimensions:





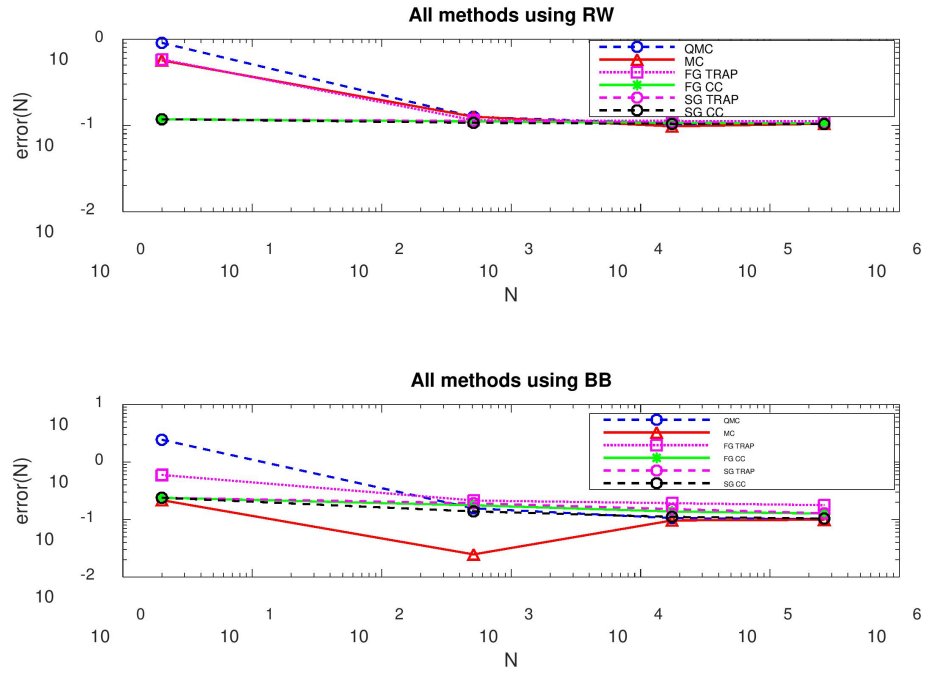
Obviously, the Sparse Grid Method has the best convergence and the Full Grid method has the second best convergence, while MC and QMC can't match these convergence rates. Since the quadrature nodes are better distributed on the integration region, QMC has better convergence than MC.

Task 15



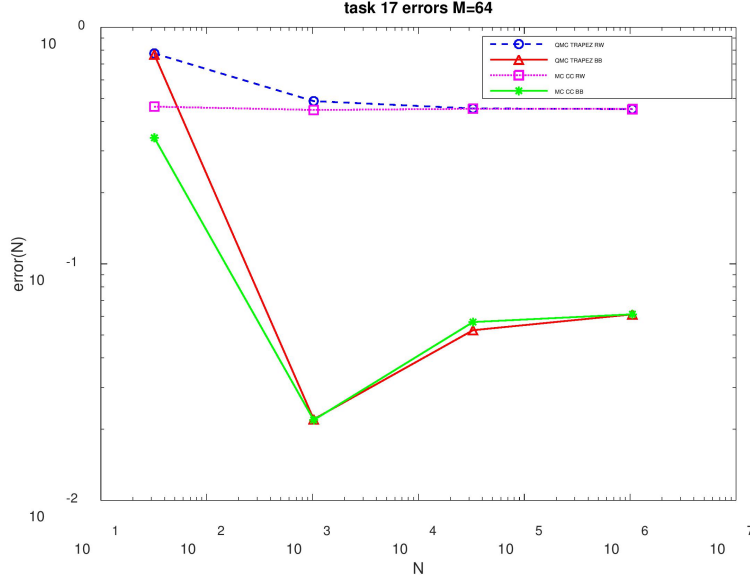
For higher N , Brownian Bridge has better accuracy than Random Walk.

Task 16



Again, Brownian Bridge has better convergence than Random Walk. Interestingly, it doesn't really make a difference whether you use MC, QMC, Full Grid or Sparse Grid.

Task 17



This again illustrates the superiority of Brownian Bridge Interpolation over Random Walk. Furthermore, the simpler MC method has almost the same convergence as QMC.

Task 18

The difference between MC and QMC is, that MC uses actual pseudo random numbers, while QMC uses deterministic sequences. For this reason, QMC Quadrature Nodes are more evenly distributed over the integration region, which enhances the convergence rate of QMC (which is $\mathcal{O}(N)$) compared to the convergence of $\mathcal{O}(N^{1/2})$ of Monte Carlo Method. However, there is a higher computational effort in creating quasi-random-number sequences, which increases the cost of QMC especially in high dimensions compared to Monte Carlo. Both, QMC and MC, can integrate a wide range of functions, regardless of how smooth these functions are.

Sparse Grids and Full Grids make use of nodes and weights which result from quadrature methods like Clenshaw-Curtis or Trapezoidal rule. These methods can only be applied to somewhat smoother functions, but they have much better convergence rates. Full Grids are heavily affected by the dimension, with the number of nodes increasing exponentially with increasing dimension. For sparse grids, dependence on dimension is considerably lower. Therefore, if it is known that the integrand will be relatively smooth, using Sparse Grid methods is the best way to compute integrals in high dimensions. However, if the integrand is unknown or if the integrand is a function that is not differentiable in many points, MC and QMC integration are the preferred methods, especially in high dimensions.