Fast Multi-asset option pricing using a parallel Fourier-based technique

C.C.W. Leentvaar and C.W. Oosterlee

Delft University of Technology

Contents

- Problem description
- Fourier-based technique
- Parallelization of the technique
- Results
- Larger problems: Sparse grids
- Conclusions

Definition of the problem

Examples of multi-asset options

- $\left(\prod_{j=1}^{d} S_j^{\frac{1}{d}} - K\right)^+$ (call on the geometric average of the assets).
- $\left(\sum_{j=1}^{d} c_j S_j - K\right)^+$, with c_j the basket weights (basket call).
- Options on the minimum or maximum of the underlying assets, for example: $(K - \max_j S_j)^+$, put on the maximum of the underlying assets.

The CONV-method

General formulation

The method presented falls in the category of transform methods. $\Rightarrow$ So use methods that are based on the risk-neutral valuation formula, for options on a single asset:

$$V(t, S(t)) = e^{-r(T-t)}\mathbb{E}\left[V(T, S(T))|S(t)\right],$$

- V is the value of the option
- r is the risk-free interest rate and is assumed to be deterministic here.
- t is the current time and T is the maturity date
- S represents the price of the underlying

Single asset European option

Solution technique

Option price can be determined directly with numerical integration if probability density is known.

$$V(t, x(t)) = e^{-r(T-t)} \int_{\ln K}^{\infty} \left(e^{x(T)} - K \right) f(x) dx,$$

with $x(T) = \ln S(T)$ and $f(x)$ the probability density function.

Use a damping function to make the integral square integrable and take the Fourier transform:

$$\begin{aligned} \psi(\omega) &= e^{-r(T-t)} \int_{-\infty}^{\infty} e^{i\omega K^*} \int_{K^*}^{\infty} e^{\alpha K^*} \left(e^{x} - e^{K^*} \right) f(x) dx dK^* \\ &= \frac{e^{-r(T-t)} \varphi(\omega - (\alpha+1)i)}{\alpha^2 + \alpha - v^2 + i(2\alpha+1)v}, \end{aligned}$$

where the characteristic function of the underlying is defined by

$$\varphi(u) = \mathbb{E}\left[e^{iu \ln S(T)} \right].$$

Single asset European option

Solution technique

To compute the call price, the inverse Fourier transform has to be computed:

$$V(t, x(t)) = \frac{e^{-\alpha K^*}}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega K^*} \psi(\omega) d\omega.$$

The same procedure cannot be done for multi-asset options. The characteristic function of a log-basket must be known. For standard basket options, quite accurate approximations can be obtained, only if it can be assumed that the basket self has the same probability density function as the the underlying assets.

The CONV-method does not rely on such an approximation.

Multi-dimensional CONV-method

General formulation

In the multi-dimensional version we need to compute:

$$V(t, \mathbf{x}(t)) = e^{-r(T-t)} \int_{\mathbb{R}^d} V(T, \mathbf{y}) f(\mathbf{y}|\mathbf{x}) d\mathbf{y},$$

where $\mathbf{x} = \ln \mathbf{S}(t)$ is a vector of the log-asset prices, $\mathbf{y} = \ln \mathbf{S}(T)$ and $f(\mathbf{y}|\mathbf{x})$ is the probability density function of the transition of $\mathbf{x}$ at time t to $\mathbf{y}$ at time T.
Main premise of the CONV method:

$$f(\mathbf{y}|\mathbf{x}) = f(\mathbf{y} - \mathbf{x}).$$

This holds for several models, such as geometric Brownian motion and, more generally, Lévy processes, which have independent increments.

The CONV-method

Fourier-transform

- Reformulate the risk-neutral valuation formula

$$V(t,\mathbf{x}) = e^{-r(T-t)} \int_{\mathbb{R}^d} V(T,\mathbf{x}+\mathbf{z}) f(\mathbf{z}) d\mathbf{z}$$

- Truncate the domain and take Fourier transform:

$$\begin{aligned} e^{r(T-t)} \mathcal{F}\{V(t,\mathbf{x})\} &= \int_{\Omega^d} e^{i\boldsymbol{\omega}\mathbf{x}} \left(\int_{\Omega^d} V(T,\mathbf{x}+\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \right) d\mathbf{x} \\ &= \int_{\Omega^d} \int_{\Omega^d} e^{i\boldsymbol{\omega}(\mathbf{y}-\mathbf{z})} V(T,\mathbf{y}) d\mathbf{y} f(\mathbf{z}) d\mathbf{z} \\ &= \int_{\Omega^d} e^{i\boldsymbol{\omega}\mathbf{y}} V(T,\mathbf{y}) d\mathbf{y} \int_{\Omega^d} e^{-i\boldsymbol{\omega}\mathbf{z}} f(\mathbf{z}) d\mathbf{z} \end{aligned}$$

The truncation of the domain is needed for two reasons. A multi-dimensional damping cannot be constructed to make the integral square integrable. Secondly, a truncated domain is mandatory for the discretization of the integral. Ω_d is chosen, such that the error is negligible.

The CONV-method

Expression

$$V(t,\mathbf{x}) = e^{-r(T-t)}\mathcal{F}^{-1}\left\{\mathcal{F}\left\{V(T,\mathbf{y})\right\}\cdot\varphi(-\boldsymbol{\omega})\right\}$$

Numerical solution:

- Truncation in asset-space and frequency space
- Use the repeated Trapezoidal rule to approximate the remaining finite integral
- Rewrite the discrete expression in terms of the discrete Fourier transforms

Discrete equation:

$$V(t,\mathbf{x}_m) \approx e^{-r(T-t)}\sum_{\mathbf{n}=0}^{\mathbf{N}-1}\left\{\sum_{\mathbf{n}=0}^{\mathbf{N}-1}\left\{\widehat{V}(T,\mathbf{y}_{\mathbf{k}})W^{-\mathbf{n}\cdot\mathbf{k}}\right\}\varphi(-\boldsymbol{\omega}_{\mathbf{n}})W_{\mathbf{N}}^{\mathbf{m}\cdot\mathbf{n}}\right\}$$

with $W_{\mathbf{N}} = e^{-\frac{2\pi i}{\mathbf{N}}}$

The CONV-method

Algorithm

Numerical algorithm:

- Define the payoff
- Take the multi-dimensional Fourier transform of the payoff
- Multiply it with the characteristic function
- Take the multi-dimensional Fourier transform again
- Discount it with the interest rate.

The CONV-method

Advantages

- We can use the FFT algorithm which is very efficient
- There is no time integration for European options
- The method is flexible and other characteristic functions can be implemented
- The sparse grid method is applicable to this method as well
- FFT can be easily parallelized $\Rightarrow$ Sparse grid may not necessary.
- Dividing one of the coordinates in two section dampens the effect of the curse of dimensionality

The CONV-method

Splitting outer transform

We split the double transforms into parts by use of the techniques on which the FFT is based. Consider the one-dimensional problem. Let $M = N/2$ and we divide the problem into a part with the odd frequency points and the even frequency points:

$$H(m) = \sum_{n=0}^{N-1} \varphi(n) W_N^{mn} \left(\sum_{k=0}^{N-1} f(k) W_N^{-nk} \right) = \sum_{n=0}^{N-1} \varphi(n) W_N^{mn} F(n) =$$

$$= \sum_{n=0}^{M-1} \varphi(2n) W_N^{m \cdot 2n} F(2n) + \sum_{n=0}^{M-1} \varphi(2n+1) W_N^{m \cdot (2n+1)} F(2n+1)$$

$$= \sum_{n=0}^{M-1} \varphi(2n) W_M^{mn} F(2n) + W_N^m \sum_{n=0}^{M-1} \varphi(2n+1) W_M^{mn} F(2n+1)$$

where $W_N^{2mn} = e^{-\frac{2\pi i m 2n}{N}} = e^{-\frac{2\pi i m n}{M}} = W_M^{mn}$

Splitting inner transform 1

The vectors $F(2n)$ and $F(2n+1)$ are taken from the vector of size N, but it can also be split:

$$\begin{aligned}
F(2n) = \sum_{k=0}^{N-1} f(k) W_N^{-2nk} &= \sum_{k=0}^{M-1} f(k) W_N^{-2nk} + \sum_{k=M}^{N-1} f(k) W_N^{-2nk} \\
&= \sum_{k=0}^{M-1} f(k) W_M^{-nk} + \sum_{k=M}^{N-1} f(k) W_M^{-nk} \\
&= \sum_{k=0}^{M-1} f(k) W_M^{-nk} + \sum_{k=0}^{M-1} f(k+M) W_M^{-n(k+M)} \\
&= \sum_{k=0}^{M-1} f(k) W_M^{-nk} + \sum_{k=0}^{M-1} f(k+M) W_M^{-nk}
\end{aligned}$$

where we used: $W_M^{-nM} = 1$

The CONV-method

Splitting inner transform 2

The vectors $F(2n)$ and $F(2n+1)$ are taken from the vector of size N, but it can also be split:

$$
\begin{aligned}
F(2n+1) = \sum_{k=0}^{N-1} f(k) W_N^{-(2n+1)k} &= \sum_{k=0}^{M-1} f(k) W_N^{-(2n+1)k} + \sum_{k=M}^{N-1} f(k) W_N^{-(2n+1)k} \\
&= \sum_{k=0}^{M-1} f(k) W_N^{-k} W_M^{-nk} + \sum_{k=0}^{M-1} f(k+M) W_N^{-(k+M)} W_M^{-n(k+M)} \\
&= \sum_{k=0}^{M-1} f(k) W_N^{-k} W_M^{-nk} - \sum_{k=0}^{M-1} f(k+M) W_N^{-k} W_M^{-nk}
\end{aligned}
$$

where we used: $W_N^{-M} = -1$

The CONV-method

Splitting results

Combination of this splitting:

$$\begin{aligned} H(m) = & \sum_{n=0}^{M-1} \varphi(2n) W_M^{mn} \left(\sum_{k=0}^{M-1} f(k) W_M^{-nk} \right) \\ & + \sum_{n=0}^{M-1} \varphi(2n) W_M^{mn} \left(\sum_{k=0}^{M-1} f(k+M) W_M^{-nk} \right) \\ & + W_N^m \sum_{n=0}^{M-1} \varphi(2n+1) W_M^{mn} \left(\sum_{k=0}^{M-1} f(k) W_N^{-k} W_M^{-nk} \right) \\ & - W_N^m \sum_{n=0}^{M-1} \varphi(2n+1) W_M^{mn} \left(\sum_{k=0}^{M-1} f(k+M) W_N^{-k} W_M^{-nk} \right) \end{aligned}$$

The CONV-method

Splitting results

Rearranging:

$$H(m) = \sum_{n=0}^{M-1} \varphi(2n) W_M^{mn} \left(\sum_{k=0}^{M-1} \left(f(k) + f(k+M) \right) W_M^{-nk} \right)$$
$$+ W_N^m \sum_{n=0}^{M-1} \varphi(2n+1) W_M^{mn} \left(\sum_{k=0}^{M-1} \left(f(k) + f(k+M) \right) W_N^{-k} W_M^{-nk} \right)$$

The CONV-method

General splitting

If we divide each asset in the multi-d setting into 2^{β_i} parts, we have:

$$V(t, \mathbf{x_m}) = \frac{e^{-r(T-t)}}{(2\pi)^d} \sum_{\mathbf{q}=\mathbf{0}}^{\boldsymbol{\beta}-\mathbf{1}} W_{\boldsymbol{\beta}}^{\mathbf{pq}} W_{\mathbf{N}}^{\mathbf{mq}} \mathcal{D}_d^{inv} \left[\varphi_{\boldsymbol{\beta}\mathbf{n}+\mathbf{q}} \mathcal{D}_d \left\{ \sum_{\mathbf{p}=0}^{\boldsymbol{\beta}-\mathbf{1}} \widehat{V}_{\mathbf{k}+\mathbf{pM}}^{T} W_{\mathbf{N}}^{-\mathbf{qk}} \right\} \right]$$

where

- $\boldsymbol{\beta} = (\beta_i, \beta_2, \ldots, \beta_d)$, $\mathbf{p} = (p_i, p_2, \ldots, p_d)$ and $\mathbf{q} = (q_i, q_2, \ldots, q_d)$, so in total there are $2d$ sums.
- $\varphi_{\boldsymbol{\beta}\mathbf{n}+\mathbf{q}} = \varphi(-\boldsymbol{\omega}_{\boldsymbol{\beta}\mathbf{n}+\mathbf{q}})$
- $\mathcal{D}_d$ and $\mathcal{D}_d^{inv}$ are the discrete Fourier transform and inverse discrete Fourier transform of size M_i/β_i for each coordinate.

The CONV-method

Computational procedures

1. Compute the payoff on the tensor-product grid;
2. Multiply the payoff by the function $G_{\mathbf{k}}$;
3. Take the fast Fourier transform;
4. Multiply the result by the characteristic function;
5. Take the inverse fast Fourier transform of the product;
6. Multiply it by the discount factor.
7. For Bermudan options: Take the maximum of this value and the payoff function at t_n. Repeat the procedure from step 2 until t_0 is reached.

Complexity analysis

Computation of the payoff and multiplying with some additional (complex) factors is a typical faster procedure than the transform itself and the multiplication with the characteristic function. We distinguish three portions of time consumption during the solution process of European options:

- T_{pay} is the time needed to construct the payoff including the multiplication with the function $Z_{\mathbf{k}}$ and $W_{\mathbf{N}}^{-\mathbf{qk}}$
- T_{four} is the time in steps 3 to 6 in the algorithm,
- T_{add} is the additional time needed for starting the computation, reading and writing files. To be assumed negligible.

So: $T_{total} = T_{pay} + T_{four}$.

Complexity analysis

Furthermore, we assume:

$$T_{four} = AT_{pay}$$

Then the total time is now:

$$T_{total} = T_{pay} + AT_{pay}$$

Now, the problem is divided in B parts (again B is a power of two), then each problem has a computational time of:

$$T_{total,split} = T_{pay} + \frac{1}{B}T_{four} = \frac{A+B}{B}T_{pay}$$

If Q is the available number of equal CPU's, then the total time reads:

$$T_{tot,split} = \left\lceil \frac{B}{Q} \right\rceil \frac{A+B}{B}T_{pay}$$

Hence: in the ideal case, Q is a divisor of B.

Results

Efficiency of the parallelization

$d=4$	Call on the geometric average			CPU times)				
n_f	Price	Error	Ratio	B=1	B=2	B=4	B=16	A
3	1.962	2.0×10^{-1}	5.3	<0.1	<0.1	<0.1	<0.1	
4	2.128	3.8×10^{-2}	5.4	<0.1	<0.1	<0.1	<0.1	
5	2.156	9.3×10^{-3}	4.1	0.5	0.2	0.1	<0.1	4.5
6	2.163	2.3×10^{-3}	4.0	9.4	4.9	3.0	1.6	6.2
7	2.165	5.8×10^{-4}	4.0	164.1	85.1	45.2	25.2	7.1

Maximum grid size: $2^{n_f} \times 2^{n_f} \times 2^{n_f} \times 2^{n_f}$, hence 2^{28} grid points.

Results

Climbing in the dimensions

$d = 5$	Digital put on the geometric average				CPU times	
n_f	Price	Error	Ratio	B=4	B=32	A
2	0.81	3.36×10^{-1}	1.49	<0.1	<0.1	
3	0.32	1.49×10^{-1}	2.26	<0.1	<0.1	
4	0.40	7.43×10^{-2}	2.00	0.2	0.1	4.0
5	0.43	3.71×10^{-2}	2.00	1.8	1.1	4.5
6	0.45	1.86×10^{-2}	2.00	295.6	91.1	8.7

The CONV-method also works on discontinuous options. Convergence ratio will decrease to 2. Five dimensional option with 64 grid points in each direction: hence 2^{30} grid points. The last grid cannot be computed on a computer with only 8 GB of memory and without splitting.

Results

Standard and Bermudan basket options

$d = 4$	European			Bermudan		
n_f	Price	Error	Ratio	Price	Error	Ratio
3	0.38	7.38×10^{-1}		0.61	5.06×10^{-1}	
4	0.87	2.51×10^{-1}	2.94	0.53	9.25×10^{-1}	0.55
5	1.05	6.82×10^{-2}	3.67	1.87	3.36×10^{-1}	2.75
6	1.10	1.76×10^{-2}	3.88	1.85	2.88×10^{-2}	11.67
7	1.11	4.51×10^{-3}	3.90	1.84	5.37×10^{-3}	5.36

Bermudan options can be used to approximate American style options via Richardson extrapolation. The irregular convergence pattern for the Bermudan options, is because of the ten times application of the payoff.

The CONV-method

Combination with sparse grids

- Sparse grids are needed, when the parts of the problem are still to large for the memory.
- Parallelization of the sparse grids is straightforward
- There are no restrictions on the number of processors. One extra processor leads to a lower total computational time, because the number of problems is much larger than the number of processors.

For example, a mimic of the full $4D$ grid with 2^{11} points per dimension.

Details			CPU times		
layer ℓ	complexity of a subproblem	#problems per layer	for a subproblem	total for the layer	paral. perform. Q=12
17	2^{20}	165	0.51	82.5	7.3
16	2^{19}	120	0.24	28.8	2.5
15	2^{18}	84	0.12	10.1	1.0
14	2^{17}	56	0.05	2.88	0.3
Tot		425		124.2	11.1

6D and 7D option

Combination with sparse grids

	6D Put on minimum			6D Put on maximum		
n_s	Price	Error	Ratio	Price	Error	Ratio
7	27.093	1.43×10^{-1}		0.375	2.33×10^{-2}	
8	27.183	9.02×10^{-2}	1.58	0.396	2.13×10^{-2}	1.09
9	27.141	4.21×10^{-2}	2.14	0.412	1.50×10^{-2}	1.42
10	27.158	1.73×10^{-2}	2.43	0.420	8.89×10^{-3}	1.69
	7D Put on minimum			7D Put on maximum		
n_s	Price	Error	Ratio	Price	Error	Ratio
7	26.153	1.22×10^{-1}		0.179	1.45×10^{-2}	
8	26.217	6.31×10^{-2}	1.93	0.194	1.50×10^{-2}	0.96
9	26.189	2.72×10^{-2}	2.32	0.206	1.14×10^{-2}	1.31
10	26.203	1.34×10^{-2}	2.02	0.213	7.21×10^{-3}	1.58

Conclusions

- The CONV method is a powerful and fast method to price European style multi-asset options
- Jumps can be introduced by choosing another characteristic function
- Parallelization of the CONV-method increases the computational efficiency and decreases the computational time.
- For options pricing problems with bounded mixed derivatives, the sparse grid method can also be used to climb in the dimensions.
- For Bermudan options the convergence is irregular in the full grid case and rather slow in the the sparse grid case. Bermudan options generally do not exhibit a smooth pasting between continuation value and early exercise payoff.

References

- C.C.W. Leentvaar Pricing multi-asset options with sparse grids. PhD thesis.
- C.C.W. Leentvaar and C.W. Oosterlee, Multi-asset option pricing using a parallel Fourier-based technique. JCF Volume 12. Nr 1. (2008).
- C.C.W. Leentvaar and C.W. Oosterlee, On coordinate transformation and grid stretching for sparse grid pricing of basket options. To appear in JCAM special issue on Numerical PDE methods in Finance
- H. bin Zubair, C.C.W. Leentvaar and C.W. Oosterlee, d-multigrid preconditioners for sparse-grid subproblems. To appear in *Int. J. Computer Math.* (2007)
- R. Lord, F. Fang, F. Bervoets and C.W. Oosterlee, "A Fast and Accurate Fft-Based Method for Pricing Early-Exercise Options Under Lévy Processes" (February 27, 2007). Available at SSRN: http://ssrn.com/abstract=966046