Product-Grid Quadrature versus Monte Carlo Sampling on a Three-Line Network

Expected curtailment and variance attribution under uncertain line ratings

Ricardo Marquez, Ph.D.
rmarquez123@gmail.com

September 2026

Abstract

A three-line network with a 160 MW load and independent, uniformly distributed line ratings is used to compare two ways of propagating input uncertainty through a TERRA model: product-grid quadrature, which marginalizes the discretized joint distribution exactly, and Monte Carlo sampling. Both methods call the same engine solve and are costed by the number of solves. Two quantities are computed — the expected curtailment $\mathbb{E}[G]$ and the Shapley attribution of $\mathrm{Var}[G]$ to the three ratings — and both are checked against a closed-form solution. For $\mathbb{E}[G]$, the grid reaches a smaller error with 729 solves than Monte Carlo reaches with 10,000, a factor of about 14 that reflects the faster convergence of quadrature in low dimension. For the attribution the difference is structural: every conditional expectation required is a partial sum over the solved grid, so the attribution costs no additional solves, whereas Monte Carlo must construct six further sample sets by pick-freeze, at $7N$ solves in total. The $n = 17$ grid places line B's share within 0.29 percentage points of the reference for 4,913 solves; Monte Carlo with $N = 3,000$ is 2.64 points off after 21,000. The grid's cost of n^d solves confines it to a few uncertain inputs; the appendices show that the compiled model exposes cuts of its dependency graph that reduce the solve count without changing any result, and discuss the extension beyond three inputs.

1 Introduction

A dynamic line-rating program must decide which conductors to instrument. The decision depends on two quantities: the expected load curtailment when the ratings fall below their nominal values, and the fraction of the curtailment's variance attributable to each line's rating. Both are functionals of the distribution of a network's response to uncertain inputs, and there are two classical ways to compute such functionals: enumerate the input space on a grid and weight each point by its probability, or sample the input space at random. This note compares the two on the smallest network for which the comparison is informative.

Section 2 states the network and the two quantities of interest. Section 3 presents the two methods — product-grid quadrature, which is how the TERRA model is marginalized exactly, and Monte Carlo sampling — for each quantity in turn. Section 4 reports the cost of each method to reach a given accuracy against the closed-form solution, and Section 5 states the conclusions. Both methods use the engine solve as their only primitive and are costed by the number of solves.

2 Problem statement

Figure 1 shows the network. Two lines in parallel supply a third, which serves a demand of 160 MW. The rating of each line is uncertain: lines A and C are rated between 70 and 120 MW, line B between 140 and 180 MW, each uniformly distributed over its range and independent of the others.

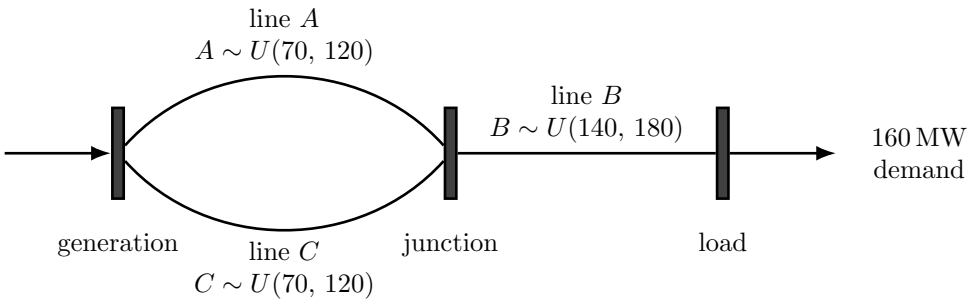


Figure 1: The three-line network. Lines A and C connect the generation bus to the junction in parallel; line B connects the junction to the load. Each line is labeled with the distribution of its rating in MW; the three ratings are independent.

When the lines cannot carry the full demand, the shortfall is the curtailment G. The parallel pair delivers at most $A + C$ to the junction and line B at most B to the load, so the power served is $\min(160,\ A + C,\ B)$ and

$$G(A, B, C) \ = \ \big(160 - \min(A + C,\ B)\big)^{+}, \qquad x^{+} = \max(x, 0).$$

The engine does not use this expression: it solves the network as a graph of conservation rows and closures (the flow split between A and C, the capacity of the pair, the requirement on B, the power served) and returns G as the difference between demand and power served. The closed form serves only to verify the engine — the two agree to 4.0×10^{-9} MW on every spot check in `results.json` — and to derive the reference values of Appendix A. Since the ratings are uncertain, G is a random variable, and two quantities are required:

1. the expected curtailment $\mathbb{E}[G]$, together with its variance;
2. the attribution of $\mathrm{Var}[G]$ to the three ratings — each line's share of the variance, which determines the order of instrumentation.

The example is small enough to solve in closed form (Appendix A): $\mathbb{E}[G] = 5.33$ MW, $\mathrm{Var}[G] = 42.0\,\mathrm{MW}^2$, and $P(G = 0) = 0.46$. Both methods are scored against these values, and the cost of each is measured in engine solves: one solution of the network for one triple of ratings.

3 Methods

3.1 Expected curtailment

Product-grid quadrature. Discretize each rating on n points, solve the network at every combination, and weight each solution by the probability of the combination:

$$\mathbb{E}[G] \ = \ \sum_{\text{grid}} P(A)\, P(B)\, P(C)\ G(A, B, C).$$

This requires n^3 solves and is deterministic. The weights are the discretized rating distributions, and because the ratings are independent the weight of a grid point is a product of one-dimensional weights.

Relation to the TERRA model. The quadrature itself is ordinary numerical integration. What is specific to TERRA is the object being integrated: the network as a graph of conservation rows and closures, with the ratings as uncertain inputs. Written out, that model is a product of factors, $P(A)\,P(B)\,P(C)$ times the physics, and the sum above is its exact marginalization — the sum-product rule of a factor graph applied to a discretized joint. The sum leaves behind a table of G over the product grid together with the product weights: a discrete representation of the model's joint distribution. Monte Carlo sampling is the other standard marginalization of the same model.

Monte Carlo sampling. Draw N independent rating triples from the joint distribution, solve each, and average:

$$\mathbb{E}[G] \;\approx\; \frac{1}{N}\sum_{\text{draws}} G(A,B,C).$$

This requires N solves; the estimate is random, with standard error proportional to $N^{-1/2}$. What remains after the computation is a single sample from the joint distribution.

3.2 Variance attribution

Shapley attribution. Each rating is assigned a share of $\mathrm{Var}[G]$; the shares sum to unity, and a rating's share is its average marginal contribution to the variance over all orderings in which the ratings are introduced (the Shapley value). The computation requires, for every subset u of the three ratings, the variance of the conditional expectation

$$v(u) \;=\; \mathrm{Var}\big(\,\mathbb{E}[G \mid X_u]\,\big),$$

where X_u denotes the ratings in u. Of the eight subsets, two are immediate — $v(\emptyset) = 0$ and $v(ABC) = \mathrm{Var}[G]$ — and six must be computed. For three inputs the shares are fixed linear combinations of the eight values; for line A,

$$\mathrm{share}_A \propto \tfrac{1}{3}\,v(A) + \tfrac{1}{6}\,[v(AB) - v(B)] + \tfrac{1}{6}\,[v(AC) - v(C)] + \tfrac{1}{3}\,[v(ABC) - v(BC)],$$

and correspondingly for B and C.

Attribution from the grid. Let the grid nodes be a_i, b_j, c_k with weights p_i, q_j, r_k (the discretized $P(A)$, $P(B)$, $P(C)$) and let $g_{ijk} = G(a_i, b_j, c_k)$ be the solved table, so that $\mathbb{E}[G] = \sum_{i,j,k} p_i q_j r_k\, g_{ijk}$. Every conditional expectation required by the attribution is the same table with some indices summed out,

$$\mathbb{E}[G \mid A = a_i] = \sum_{j,k} q_j\, r_k\, g_{ijk}, \qquad \mathbb{E}[G \mid A = a_i, B = b_j] = \sum_{k} r_k\, g_{ijk},$$

and each $v(u)$ is the weighted variance of the corresponding partial sums:

$$v(A) = \sum_i p_i \Big(\sum_{j,k} q_j r_k\, g_{ijk}\Big)^2 - \mathbb{E}[G]^2, \qquad v(AB) = \sum_{i,j} p_i q_j \Big(\sum_{k} r_k\, g_{ijk}\Big)^2 - \mathbb{E}[G]^2.$$

Two properties make these quantities free. The joint distribution is a product, $P(A)\,P(B)\,P(C)$, and g is tabulated on the product grid; by Fubini's theorem any subset of coordinates may therefore be summed out, in any order, as a sequence of one-dimensional sums over a table already computed. This is the sum-product rule of a factor graph: marginalizing a factored joint is local arithmetic. All six $v(u)$ cost $O(n^3)$ additions and no solves. The attribution is trivial for the grid because the grid is a representation of the joint distribution and the attribution consists of its marginals.

Attribution by pick-freeze sampling. A random sample cannot be conditioned on a fixed value of A after the fact: each draw has its own A, and with probability one no two draws share a value, so a plain sample contains nothing to condition on. The conditional expectations must instead be built into the design of a second sample:

1. Draw two independent sample sets of N rating triples, $X^{(1)}$ and $X^{(2)}$. Solve the network on $X^{(1)}$ (N solves); this base set also yields the mean and $v(ABC)$.

2. For each of the six subsets u, construct a hybrid set whose columns for the ratings in u are taken from $X^{(1)}$ and whose remaining columns are taken from $X^{(2)}$. For $u = \{A\}$, each row retains its A from the first set and receives B and C from the second. Solve the network on the hybrid set (N solves).

3. Row i of $X^{(1)}$ and row i of the hybrid set share exactly the ratings in u and are independent in the remainder. Conditioning on the shared ratings factorizes the expectation of the product,

$$\mathbb{E}\big[G(A,B,C)\,G(A,B',C')\big] = \mathbb{E}_A\big[\,\mathbb{E}[G \mid A]^2\,\big] = v(A) + \mathbb{E}[G]^2,$$

   so that $v(A) \approx \frac{1}{N}\sum_i G(X_i^{(1)})\,G(\text{hybrid}_i) - \bar{G}^2$. This is the pick-freeze identity; it holds only because the hybrid set retains the u-columns and redraws the others.

4. Combine the eight $v(u)$ with the Shapley weights.

The cost is one base set and six hybrid sets, $7N$ solves — seven times the cost of the mean alone; with K uncertain inputs it would be $(2^K - 1)N$. The contrast with the grid is exact. On the grid the conditioning structure resides in the object (a product grid with product weights, so that every marginal is a Fubini sum); in sampling it must be placed in the design of each sample, one design per conditional. Table 1 summarizes the cost of each quantity.

Table 1: Cost of each conditional variance required by the Shapley attribution.

subset u	product grid	Monte Carlo
$\emptyset$	0 (identically zero)	0 (identically zero)
ABC	0 (variance of the solved table)	0 (variance of the base set)
A, B, C	0 each (partial sums)	N solves each (one hybrid set)
AB, AC, BC	0 each (partial sums)	N solves each (one hybrid set)
total beyond the mean	**0**	$6N$

4 Results and discussion

4.1 Expected curtailment

Table 2: Expected curtailment and its variance by both methods. The error is the absolute deviation of the estimated mean from the closed form; for Monte Carlo it is the root-mean-square error over 8 seeds, and the mean and variance shown are those of the first seed.

method	engine solves	$\mathbb{E}[G]$ (MW)	$\mathrm{Var}[G]$ (MW^2)	error (MW)	time
grid, $n = 9$	729	5.3784	44.42	0.0451	1 s
grid, $n = 17$	4,913	5.3457	42.56	0.0124	5 s
grid, $n = 25$	15,625	5.3387	42.23	0.0054	17 s
grid, $n = 33$	35,937	5.3360	42.11	0.0026	38 s
grid, $n = 41$	68,921	5.3345	42.06	0.0012	74 s
Monte Carlo, $N = 300$	300	4.7962	40.47	0.5023	0 s
Monte Carlo, $N = 1,000$	1,000	5.5884	44.19	0.1278	1 s
Monte Carlo, $N = 3,000$	3,000	5.3483	42.02	0.1131	4 s
Monte Carlo, $N = 10,000$	10,000	5.4523	42.95	0.0589	12 s
closed form	—	5.3333	41.96	—	—

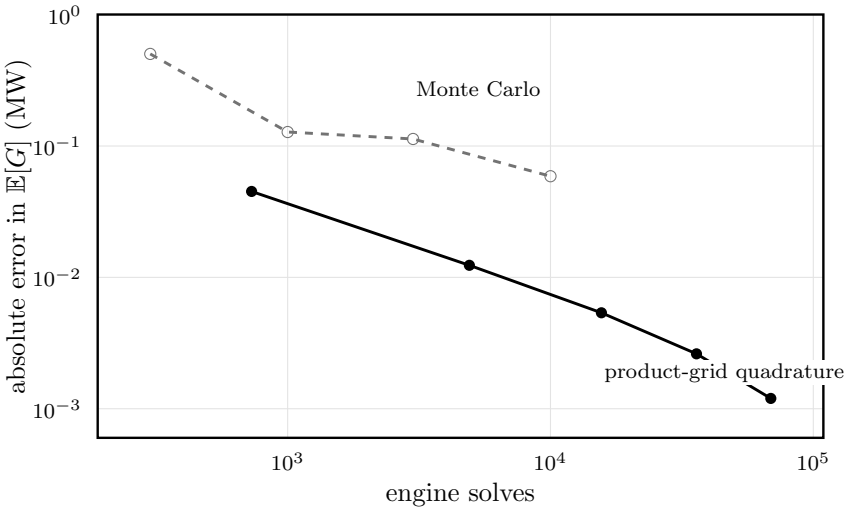


Figure 2: Absolute error in $\mathbb{E}[G]$ against the number of engine solves. Grid points are single deterministic runs; Monte Carlo points are the root-mean-square error over 8 seeds. Both axes are logarithmic.

Both methods converge to the closed-form values (Table 2, Figure 2); they differ in cost. The largest Monte Carlo run reaches a root-mean-square error of 0.06 MW with 10,000 solves; the grid reaches a smaller error with 729 solves, a factor of about 14. The gap widens with the accuracy demanded, because the two errors decay at different rates in the solve count: as its inverse square root for Monte Carlo and approximately as its inverse for the grid.

4.2 Variance attribution

Table 3: Shapley attribution of Var[G] to the three ratings. "Additional solves" are those required beyond the estimate of the mean: none for the grid, six further sample sets for Monte Carlo. The error is the absolute deviation of line B's share from the reference, in percentage points; for Monte Carlo it is the root-mean-square over 5 seeds.

method	engine solves	additional solves	A (%)	B (%)	C (%)	error (pp)
grid, $n = 9$	729	0	3.76	92.47	3.76	1.12
grid, $n = 17$	4,913	0	3.35	93.30	3.35	0.29
grid, $n = 25$	15,625	0	3.27	93.47	3.27	0.13
grid, $n = 33$	35,937	0	3.24	93.52	3.24	0.08
grid, $n = 41$	68,921	0	3.23	93.55	3.23	0.05
Monte Carlo, $N = 300$	2,100	1,800	4.54	91.85	3.61	7.05
Monte Carlo, $N = 1,000$	7,000	6,000	1.46	96.16	2.38	3.11
Monte Carlo, $N = 3,000$	21,000	18,000	5.56	87.93	6.52	2.64
reference	—	—	3.20	93.60	3.20	—

Line B accounts for approximately 94% of the variance, and both methods identify it (Table 3). The $n = 17$ grid places B's share within 0.29 percentage points of the reference for 4,913 solves in total, whereas Monte Carlo with $N = 3,000$ requires 21,000 solves, of which 18,000 serve the attribution alone, and is 2.64 percentage points from the reference.

4.3 Discussion

Source of the advantage. Two distinct effects should be kept apart. For the mean, the advantage is only that a deterministic quadrature converges faster than random sampling in three dimensions. For the attribution, the advantage is structural: the grid holds the joint distribution on a product set, so every conditional expectation is a Fubini sum over it, whereas a sample must be redesigned for each conditional. The second advantage does not depend on the grid resolution.

Limits. The cost of the grid is n^3 here and n^d for d uncertain inputs. Beyond four or five inputs no grid is affordable at a useful resolution, whereas the cost of Monte Carlo is independent of d. A low-discrepancy point set (quasi-Monte Carlo) converges faster than random sampling and recovers much of the grid's advantage for the mean, but it must still be resampled for the attribution, so the second advantage persists. Appendix B examines whether the grid's cost can be reduced by reading the model's structure rather than by coarsening the grid, and Appendix C discusses the extension beyond three inputs.

5 Conclusions

- For three uncertain ratings, product-grid quadrature reaches a given accuracy in $\mathbb{E}[G]$ with roughly 14× fewer engine solves than Monte Carlo sampling — the usual advantage of quadrature over sampling in low dimension.
- Variance attribution costs nothing beyond the mean on the grid and seven times the mean with Monte Carlo, because the grid is a representation of the joint distribution and a sample

is not. This is the advantage that belongs to the model-based formulation, and it survives improved sampling.

- Neither advantage is a general claim of speed: the grid becomes impractical beyond a small number of uncertain inputs. Any faster method must be assessed against sampling on the same network at the same solve count.

A Closed-form reference values

The expression for G in Section 2 splits into two independent terms,

$$G = \big(160 - \min(A + C,\, B)\big)^+ = \max(d, V), \qquad d = (160 - A - C)^+, \quad V = (160 - B)^+.$$

V is zero when $B \geq 160$ (probability $\frac{1}{2}$) and otherwise uniform on $(0, 20)$; d depends only on the sum $S = A + C$, whose density is triangular on $[140, 240]$ with mode 190.

Integration over B. For fixed $d \in [0, 20]$, $G = \max(d, V)$ equals d when $V \leq d$ and V otherwise, so

$$\mathbb{E}[G \mid d] = d\Big(\tfrac{1}{2} + \tfrac{d}{40}\Big) + \int_d^{20} \frac{v}{40}\, dv = 5 + \frac{d}{2} + \frac{d^2}{80}, \qquad \mathbb{E}[G^2 \mid d] = \frac{200}{3} + \frac{d^2}{2} + \frac{d^3}{60}.$$

Integration over S. Write $S = 140 + 50T$ with $T = U_1 + U_2$ for independent uniforms on $[0, 1]$, so that T has triangular density t on $[0, 1]$. Curtailment on the parallel pair requires $S < 160$, i.e. $T < 0.4$, in which case $d = 20 - 50T$; otherwise $d = 0$. Hence

$$\mathbb{E}[d^m] = \int_0^{0.4} (20 - 50t)^m\, t\, dt : \qquad \mathbb{E}[d] = \tfrac{8}{15}, \quad \mathbb{E}[d^2] = \tfrac{16}{3}, \quad \mathbb{E}[d^3] = 64.$$

Moments.

$$\mathbb{E}[G] = 5 + \tfrac{1}{2} \cdot \tfrac{8}{15} + \tfrac{1}{80} \cdot \tfrac{16}{3} = \frac{16}{3} = 5.3333\,\text{MW},$$

$$\mathbb{E}[G^2] = \frac{200}{3} + \frac{1}{2} \cdot \frac{16}{3} + \frac{64}{60} = 70.4, \qquad \mathrm{Var}[G] = 70.4 - \Big(\frac{16}{3}\Big)^2 = 41.9556\,\text{MW}^2.$$

Curtailment is zero exactly when $S \geq 160$ and $B \geq 160$, so

$$P(G = 0) = P(T \geq 0.4) \cdot P(B \geq 160) = (1 - 0.08) \cdot \tfrac{1}{2} = 0.46.$$

Reference shares. The Shapley shares of Table 3 have no comparably short closed form, since the conditional variances involve piecewise polynomials over the triangular density. The reference row is therefore the partial-sum computation of Section 3 at a resolution far beyond the grids of Section 4 (241 points per rating, 1.4×10^7 combinations), evaluated with the closed-form G in place of the engine, which the spot checks of Section 2 justify. At that resolution the shares are 3.20 %, 93.60 % and 3.20 %, and the same computation returns $\mathbb{E}[G]$ and $\mathrm{Var}[G]$ within 4×10^{-5} and 3×10^{-3} of the closed forms above.

B Cuts of the dependency graph

Sections 3 and 4 use no knowledge of how the three ratings enter G. This appendix records the structure that the compiled model exposes without annotation, and quantifies what it would have saved.

The dependency graph. A compiled TERRA model consists of a set of variables $\mathcal{V}$ and a set of relations. Each closure declares an output w and its inputs, $w = \varphi_w(\mathrm{in}(w))$; a conservation node with exactly one closure-free incident flow determines that flow from the others. Reading the relations as a directed graph D on $\mathcal{V}$, with an edge $v \to w$ whenever $v \in \mathrm{in}(w)$, the present model gives

$$\begin{aligned} \texttt{G} &\leftarrow \texttt{demand_mw}, \texttt{served} \\ \texttt{cap_pair} &\leftarrow \texttt{rate_A}, \texttt{rate_C} \\ \texttt{f_A} &\leftarrow \texttt{f_B}, \texttt{rate_A}, \texttt{rate_C} \\ \texttt{f_B} &\leftarrow \texttt{req_mid}, \texttt{cap_pair} \\ \texttt{f_C} &\leftarrow \texttt{f_A}, \texttt{f_B} \\ \texttt{req_mid} &\leftarrow \texttt{demand_mw}, \texttt{rate_B} \\ \texttt{served} &\leftarrow \texttt{f_B} \end{aligned}$$

where $\leftarrow$ denotes "is determined by". Let $U = \{\texttt{rate_A}, \texttt{rate_B}, \texttt{rate_C}\}$ be the uncertain inputs, $K = \{\texttt{demand_mw}\}$ the constants, and $\mathcal{E}$ the set of *explicit* variables: the smallest set closed under the rule that a variable belongs to $\mathcal{E}$ whenever all of its inputs lie in $U \cup K \cup \mathcal{E}$. An explicit variable is evaluated from the inputs by composing closures, without an engine solve; a variable determined by a conservation row is not.

Cuts. A set $F \subseteq \mathcal{E} \cup U$ is a *cut* for the target G if every directed path in D from U to G passes through F. Since every variable downstream of F is determined by F, there exists a function h with

$$G(x) \;=\; h\big(F(x)\big) \qquad \text{for every rating triple } x,$$

where $F(x)$ denotes the vector of cut values at x, computable without a solve. The quadrature of Section 3 therefore factors as

$$\mathbb{E}[G] \;=\; \sum_{x \in \mathrm{grid}} P(x)\, G(x) \;=\; \sum_{f \in F(\mathrm{grid})} h(f) \sum_{x:\, F(x) = f} P(x),$$

and the engine is required once per *distinct cut value* $f \in F(\mathrm{grid})$ rather than once per grid point; the inner sums are arithmetic. For the junction cut $F = (A + C,\, B)$ on an equally spaced n-point grid, $A + C$ takes $2n - 1$ distinct values, so $|F(\mathrm{grid})| = (2n - 1)\, n$ against n^3 grid points: 3,321 solves in place of 68,921 at $n = 41$.

Enumerating the cuts. Starting from $\{G\}$, repeatedly replace one variable by the variables that determine it (one step backwards along D). Every set reached in this way is a cut; those contained in $\mathcal{E} \cup U$ are the cuts evaluable without a solve. For a graph of this size the search is exhaustive. The result is the ladder of Table 4, ordered by $|F|$, with each cut's $|F(\mathrm{grid})|$ on the grids of Section 4. Along the ladder the grid, the weights and hence every quantity reported are unchanged; only the solve count varies.

Table 4: Cuts enumerated from the compiled model, with the distinct engine solves $|F(\mathrm{grid})|$ each requires on an n^3 grid; the last column is the reduction relative to n^3 at $n = 41$.

cut F	$\lvert F\rvert$	$n = 9$	$n = 17$	$n = 25$	$n = 33$	$n = 41$	reduction
`f_B`	1	8	14	20	26	33	2,089×
`cap_pair` + `rate_B`	2	153	561	1,225	2,145	3,321	21×
`cap_pair` + `req_mid`	2	85	297	637	1,105	1,701	41×
`rate_A` + `rate_B` + `rate_C`	3	729	4,913	15,625	35,937	68,921	1×
`rate_A` + `rate_C` + `req_mid`	3	405	2,601	8,125	18,513	35,301	2×

As a check of the factorization, the $n = 17$ quadrature of Section 4 was recomputed through the `cap_pair` + `rate_B` cut, solving its 561 distinct cut values in place of 4,913 grid points; the two estimates of $\mathbb{E}[G]$ agree to 9.7×10^{-14} MW. Figure 3 repeats the convergence plot of Section 4 with the quadrature evaluated through each cut.

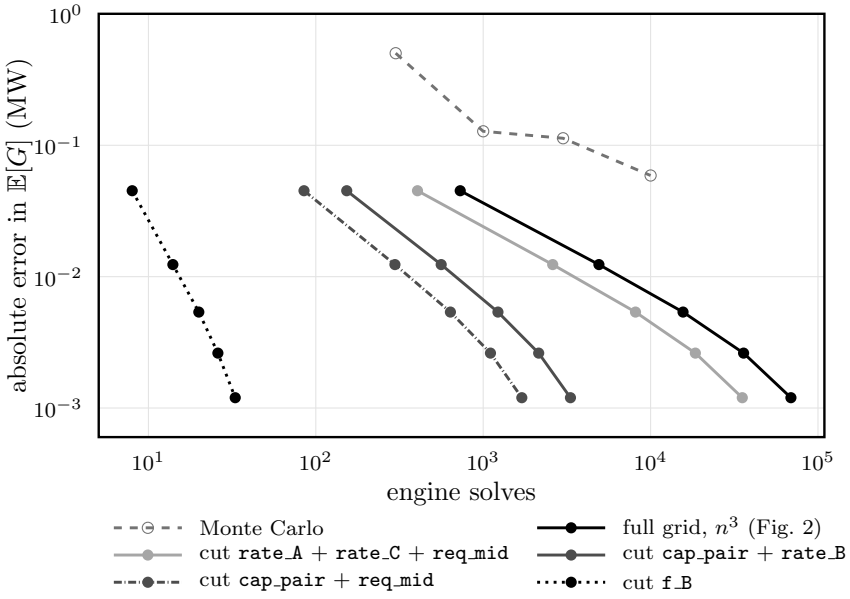


Figure 3: Figure 2 with the grid quadrature evaluated through each cut of the ladder. Every grid curve carries the same five errors, since the quadrature is unchanged; each is translated left by the cut's reduction in distinct solves, $|F(\text{grid})|$ against n^3.

Two qualifications. The junction cut, `cap_pair` with `rate_B`, is the model's own recovery of the fact that A and C reach the load only through their sum — the observation a modeler would otherwise supply by hand, here supplied by D. The one-variable cut, `f_B`, lies one closure upstream of G; it exists because this small model is almost entirely explicit, and its 2,000-fold reduction reflects the toy rather than the method. The enumeration itself is implemented in this study; the engine exposes D but does not yet offer the search as a feature.

C Extension beyond three inputs

The attribution advantage of Section 4 rests on the grid being a representation of the joint distribution, so that every conditional expectation is a partial sum over it. The limitation is not the attribution but the grid: n^d solves for d uncertain inputs, which is impractical beyond four or five. Three distinct limits are contained in that statement, and each admits a different remedy.

The grid need only cover a cut. If the model yields $G = h(F(x))$ for a set of intermediate quantities F (Appendix B), the grid must cover the distinct values of F rather than of x; here $F = (A+C,\ B)$ reduces n^3 to $(2n-1)\,n$. More generally, exact marginalization of a factored model costs exponential in the width of its dependency graph rather than in the number of inputs, and on a radial network, where each rating acts locally, the conditional expectations required for attribution are the same local sums. The limiting case is a meshed network under a dispatch optimization:

every rating is coupled to every other through a single global factor, and no decomposition is available.

Each solve returns derivatives. Every engine solve also returns $\partial G/\partial x_i$ for each rating — the Jacobian, or for a dispatch linear program the dual variable on each rating bound — at no additional cost. From the same N Monte Carlo solves that estimate the mean one therefore obtains, for each input,

$$\nu_i = \mathbb{E}\Big[\Big(\frac{\partial G}{\partial x_i}\Big)^2\Big], \qquad S_i^{\mathrm{tot}} \;\le\; \frac{(b_i - a_i)^2}{\pi^2 \operatorname{Var}[G]}\,\nu_i \quad \text{for } x_i \sim U(a_i, b_i),$$

an upper bound on the input's total Sobol' index (the Sobol'–Kucherenko inequality). This requires no hybrid sample sets, no factor of $7N$, and no 2^d enumeration, in any dimension. It is a bound rather than a share, but for the instrumentation decision a bound that excludes a line is sufficient, and the exact shares can then be recovered on the few lines that remain. This is the route native to the model-based formulation: the attribution is read from the same object that produces the estimate.

The 2^d enumeration belongs to the Shapley question. Shapley attribution requires $v(u)$ for every subset u; the grid makes each free but does not reduce their number. Beyond a small number of inputs, practice reports first-order and total indices, $d + 2$ quantities, and applies the Shapley decomposition only when interaction variance is large. It is large here: A and C enter G only through their sum, which is why the decomposition was informative on three lines. Sampling designs that replicate marginal levels (replicated Latin hypercubes, orthogonal arrays) yield all first-order indices from $2N$ solves in any dimension, by building into the design part of the replication a grid possesses by construction. Surrogates fitted to the sample (polynomial chaos expansions) yield every index from their coefficients, provided G is smooth enough to be fitted; a piecewise-linear G with kinks is not, without a piecewise treatment.

Independence is the substantive assumption. The product weights $P(A)\,P(B)\,P(C)$ are what make every marginal a Fubini sum, and line ratings in practice are not independent: they share weather. Correlation invalidates both the product grid and the standard interpretation of the shares. The remedy is to represent the correlation through a shared driver, $P(A, B, C) = \int P(A \mid w)\,P(B \mid w)\,P(C \mid w)\,P(w)\,dw$: conditional on w the product structure is restored, and w — a small number of weather modes — is low-dimensional. The grid then covers w, the effective dimension is $\dim w$ rather than d, and the attribution question becomes which weather mode is responsible, which is typically the more useful one.

Reproduction. With `PYTHONPATH=src`, run `run.py`, then `run_structure.py` (Appendix B), then `make_report.py` with `--pdf`. Seed 20260906, 391,030 engine solves in the main run; the engine reproduces the closed-form G to 4.0×10^{-9} MW. Every number above is read from `results.json`.