

A Framework for Solving Hybrid Influence Diagrams containing Deterministic Conditional Distributions*

Yijing Li
yjl@ku.edu

Prakash P. Shenoy
pshenoy@ku.edu

University of Kansas School of Business
1300 Sunnyside Ave., Summerfield Hall
Lawrence, KS 66045-7601 USA

February 11, 2012

*Accepted for publication in *Decision Analysis*, 9(1), March 2012.

Contents

1	Introduction	4
2	Previous Work on Solving Hybrid IDs	5
2.1	Discretization	5
2.2	Monte Carlo Methods	6
2.3	Gaussian IDs	6
2.4	Mixture of Gaussian IDs	6
2.5	Mixture of Truncated Exponentials	7
2.6	Mixture of Polynomials	7
2.7	Contributions	7
2.8	Limitations	8
3	A Framework for Solving Hybrid IDs	9
3.1	Variables and States	9
3.2	Projection of States	10
3.3	Discrete Potentials	10
3.4	Continuous Potentials	10
3.5	Dirac Delta Functions	11
3.6	Constraint Potentials	11
3.7	Utility Potentials	12
3.8	Summary	12
3.9	An Example	12
3.10	Combination of Potentials	14
3.11	Marginalization of Potentials	14
3.12	Division of Potentials.	15
3.13	An Algorithm for Solving Hybrid Influence Diagrams.	16
3.14	Marginalizing a Chance Variable-Case 1	16
3.15	Marginalizing a Decision Variable-Case 1	16
3.16	Marginalizing a Chance/Decision Variable-Case 2	17
4	Mixture of Polynomials Functions	17
4.1	MOP Functions	18
4.1.1	Advantages of Hyper-rhombus Regions	19
4.2	Finding MOP Approximations of Univariate PDFs	20
4.2.1	Lagrange Interpolating Polynomials	20
4.2.2	Chebyshev Points	21
4.2.3	Finding a MOP Approximation for a PDF	21
4.3	Finding Piecewise-Linear Approximations of Deterministic Conditionals	22
5	Two Examples	23
5.1	Entrepreneur’s Problem	23
5.1.1	Marginalizing C_a	24
5.1.2	Marginalizing Z_2	24

5.1.3	Marginalizing C_n	25
5.1.4	Marginalizing Q_a	25
5.1.5	Marginalizing Z_1	26
5.1.6	Marginalizing Q_n	26
5.1.7	Marginalizing P	26
5.2	An American Put Option Problem	26
5.2.1	Marginalizing D_3 and S_3	28
6	Summary and Conclusions	29
6.1	Limitations	30
6.2	Future Work	30

A Framework for Solving Hybrid Influence Diagrams containing Deterministic Conditional Distributions

Yijing Li and Prakash P. Shenoy

Abstract

We describe a framework and an algorithm for approximately solving a class of hybrid influence diagrams (IDs) containing discrete and continuous chance variables, discrete and continuous decision variables, and deterministic conditional distributions for chance variables. A conditional distribution for a chance variable is said to be deterministic if its variances, for each state of its parents, are all zeroes. The solution algorithm is an extension of Shenoy’s fusion algorithm for discrete influence diagrams. To mitigate the integration and optimization problems associated with solving hybrid IDs, we propose using mixture of polynomials approximations of conditional probability density and utility functions, and piecewise linear approximations of nonlinear deterministic conditional distributions for continuous chance variables. The class of hybrid IDs that can be solved by our framework are those that do not involve divisions. The framework and algorithm are illustrated by solving two small examples of hybrid IDs.

Key Words: solving hybrid influence diagrams, deterministic conditional distributions, mixture of polynomials

1 Introduction

An influence diagram (ID) is a formal compact representation of a Bayesian decision making under uncertainty problem. It consists of four parts: a sequence of decisions, a set of chance variables with a joint distribution represented by a Bayesian network (BN), the decision maker’s preferences for the uncertain outcomes represented by a joint utility function, and information constraints that indicate what uncertainties are known and unknown when a decision has to be made. IDs were initially defined by Howard and Matheson [18, 19]. Howard and Matheson’s definition of an ID allowed a single (unfactored) utility node. Tatman and Shachter [42] subsequently generalized IDs to include multiple utility nodes that combine additively or multiplicatively or some combination of the two. In this paper, we assume that the utility factors combine additively.

Hybrid IDs are IDs containing a mix of discrete and continuous chance variables, and discrete and continuous decision variables. A conditional distribution (or conditional, in short) for a chance variable in an ID is said to be *deterministic* if the variances, for each state of the variable’s parents, are all zeroes. Deterministic conditionals for discrete chance variables pose no computational problems. Deterministic conditionals for continuous chance variables pose a computational challenge as the joint density function for all continuous variables does not exist, and this non-existence can pose problems when solving such IDs. Therefore, here onwards, when we speak of variables with deterministic conditionals we are referring to continuous variables.

In practice, one encounters decision problems in which some chance and decision variables (such as demand, cost, stock price, profit, etc.) are continuous in nature. If we maintain the continuous nature of these variables (i.e., we do not discretize such variables), the result is a hybrid ID. However, solving a hybrid ID involves two main computational challenges. First, marginalizing a continuous chance variable involves integration of a product of density and utility functions. In some cases, such as the Gaussian density function, there may not exist a closed-form representation of the integral. We will refer to this problem as the *integration* problem.

Second, marginalizing a decision variable involves maximizing a utility function. If a decision variable is continuous and has relevant continuous information predecessors, then we may be faced with the problem of finding a closed-form solution to the maximization problem. Not only do we have to find an optimal value of the decision variable as a function of the states of its relevant information predecessors, we also have to find a closed-form expression of the maximum utility as a function of the states of its relevant information predecessors. We will refer to this problem as the *optimization* problem.

In this paper, we describe a framework and an algorithm for solving a class of hybrid IDs approximately. The framework is an extension of the Shenoy-Shafer [38] architecture for making inferences in hybrid BNs described in Shenoy and West [39], and includes decision variables and utility functions. The algorithm consists of using mixtures of polynomials (MOPs) for approximating PDFs of continuous variables, approximating nonlinear deterministic conditionals by piecewise linear ones, using Dirac delta functions to represent deterministic conditionals for continuous chance variables, and approximating utility functions by MOPs. The class of hybrid IDs that can be solved by our framework are those IDs that can be solved using local computation without the use of the division operation. We illustrate our method by solving two small examples.

An outline of the remainder of this paper is as follows. In section 2, we review the literature on solving hybrid IDs, we list the contributions of our paper, and sketch the limitations of our method. In section 3, we describe a framework and an algorithm to solve hybrid IDs with deterministic variables. In section 4, we define MOP functions, and a process for approximating conditional PDFs and utility functions by MOP functions, and a process for finding piecewise linear approximations of nonlinear deterministic conditionals. In section 5, we solve two decision problems to illustrate our framework and algorithm. Finally, in section 6, we conclude with a summary and a discussion on the limitations of using MOP functions for solving hybrid IDs and some related topics for future work.

2 Previous Work on Solving Hybrid IDs

In this section, we review previous work on solving hybrid IDs, and discuss the main contributions and limitations of our method.

2.1 Discretization

A traditional method for solving a hybrid ID is to approximate the hybrid ID with a discrete ID by discretizing the continuous chance and decision variables (see, e.g., Miller and Rice

[28], Keefer and Bodily [21], and Smith [41]). If we discretize a continuous variable using too few bins, we may have an unacceptable approximation of the problem. On the other hand, if we use many bins, we increase the computational effort of solving the resulting discrete ID. In the BN literature, Kozlov and Koller [23] described a dynamic non-uniform discretization technique for chance variables depending on the region where the posterior density lies. This technique needs to be adapted for solving hybrid IDs.

2.2 Monte Carlo Methods

Another method for solving hybrid IDs is to use Monte Carlo (MC) methods. One of the earliest to suggest MC methods for solving decision trees was Hertz [15], where the entire joint distribution of all chance variables is sampled. Charnes and Shenoy [5] proposes a MC method that samples from a small set of chance variables at a time for each decision variable. Ortiz and Kaelbling [31] proposes several MC methods and provides bounds on the number of samples required given some error bounds. Bielza et al. [1] explores the use of Markov chain MC methods to solve a single-stage decision problem with continuous decision and chance nodes to solve the maximization problem. Cano et al. [4] describes a forward-backward Monte Carlo method for approximate solutions of IDs. While Monte Carlo methods can handle continuous chance variables, there is one limitation. If we have a decision variable with continuous chance variables as relevant predecessors, then finding an optimal decision function for the decision variable requires discretization of the continuous chance variables that are in the relevant domain.

2.3 Gaussian IDs

Among exact methods, Shachter and Kenley[34] provides a theory to solve IDs where all chance and decision variables are continuous. The continuous chance variables are required to have the conditional linear Gaussian (CLG) distributions, and the utility function is required to be quadratic. Such IDs are called Gaussian IDs. These requirements ensure that the joint distribution of all chance variables is multivariate Gaussian, whose marginals can be easily found without the need for integration. Also, the quadratic nature of the utility function ensures that there is a unique maximum that can be computed in closed form without the need for searching for an optimal solution.

2.4 Mixture of Gaussian IDs

Poland [32] and Poland and Shachter [33] extend Gaussian IDs to include discrete chance variables that do not have continuous parents. If a continuous chance variable does not have a CLG distribution, then it can be approximated by a mixture of Gaussians represented by a discrete variable with mixture weights and a continuous variable with the discrete variable as its parent and with CLG distributions. Like Gaussian IDs, mixture of Gaussian IDs are required to have quadratic utility functions.

2.5 Mixture of Truncated Exponentials

To find posterior marginals in hybrid BNs, Moral et al. [29] proposes approximating PDFs by mixtures of truncated exponentials (MTEs) as a solution for the integration problem. The family of MTE functions is easy to integrate, and is closed under combination and marginalization, and can be propagated using the Shenoy-Shafer architecture ([38]). Cobb et al. [11] describes MTE approximations for several commonly-used uni-variate PDFs such as normal, log-normal, Gamma, etc. Cobb and Shenoy [8] extends the MTE BN framework to include one-dimensional deterministic conditionals described by linear functions. For one-dimensional nonlinear functions, Cobb and Shenoy[9] proposes approximating them by piecewise linear functions.

For solving IDs, Cobb and Shenoy [10] describes MTE IDs, where the PDFs of continuous chance variables and the utility functions are described using MTE functions, and decision nodes are all discrete. Thus any PDF can be used as long as they can be approximated by MTEs, and discrete variables can have continuous parents. Cobb [7] describes continuous decision MTE IDs, where in addition to using MTE potentials to represent PDFs and utility functions, continuous decisions are allowed.

The MTE methods surveyed here for BNs and IDs cannot cope with multi-dimensional linear deterministic conditionals. For example, if X and Y are independent exponential random variables with Poisson rate parameter $\lambda = 1$ (whose PDFs are MTE functions), then $Z = X + Y$ has a Gamma distribution (with parameters $r = 2$, and $\lambda = 1$), whose PDF ($f_Z(z) = ze^{-z}$ if $z > 0$) is not an MTE function (because of the presence of the ze^{-z} term in the PDF).

2.6 Mixture of Polynomials

Similar to MTEs, Shenoy and West[40], and Shenoy [37] propose approximating PDFs by piecewise polynomial functions called mixtures of polynomials (MOPs). Like MTEs, MOPs are closed under multiplication, addition, and integration. Thus, they can be used to find marginals in hybrid BNs using the Shenoy-Shafer [38] architecture. MOP functions have some advantages over MTE functions. MOP approximations can be found (more easily than MTE) using Lagrange interpolating polynomials with Chebyshev points ([37]), even for multidimensional ones. Also, they are closed for a larger class of deterministic functions than MTE functions, which are closed only for one-dimensional linear functions (e.g., $W = aX + b$). MOP functions are closed under transformations required for multi-dimensional linear (e.g. $W = X + Y$), and for multi-dimensional quotient (e.g., $W = X/Y$, $W = (X/Y)/Z$), etc.) deterministic functions.

2.7 Contributions

The major contributions of this paper are as follows. First, we further extend the extended Shenoy-Shafer architecture, described in [39] for inference in hybrid BNs, to enable the solution of hybrid IDs with deterministic conditionals. We extend the architecture to include discrete and continuous decision variables, and utility functions. The algorithm for solving

hybrid IDs is essentially the same as the fusion algorithm proposed in Shenoy [36] for discrete IDs.

Second, to address the integration and optimization problems, we propose using MOP approximations of PDFs and utility functions. The family of MOP functions is closed under multiplication, addition, integration, and transformations needed for multi-dimensional linear deterministic functions. It is not closed under divisions, or transformations needed for nonlinear deterministic functions (such as $Z = X \cdot Y$, $Y = X^2$, etc.) For hybrid IDs that contain nonlinear deterministic conditionals, we propose approximating these by piecewise linear functions as suggested in Cobb and Shenoy [9].

Regarding the optimization problem, because MOP functions are easily differentiable, finding the maximum of a utility function that is in MOP form is also easier than for non-MOP utility functions.

Previous methods for solving IDs containing continuous chance variables assume either CLG conditionals, in which case one can allow deterministic conditionals described by linear functions ([34, 33]), or non-CLG conditionals that are approximated by MTEs ([10]), which are closed only for one-dimensional linear deterministic conditionals ([35]). The framework described here extends the class of IDs which can be solved—the chance variables can have any distributions as long as they can be approximated by MOPs, the utility functions can be of any form as long as they can be approximated by MOPs, there are no topological restrictions such as discrete variables with no continuous parents, and we can have any deterministic conditionals as long as they can be approximated by piecewise linear functions.

2.8 Limitations

Some limitations of our method are as follows. First, the family of MOP functions is not closed under the division operation. Solving an ID with an additive factorization of the utility function using local computation may require divisions. Such problems will not be amenable to our method. The *Pigs* problem, discussed in Lauritzen and Nilsson [26], is an example of a problem of this type (requires divisions for solution using local computation).

Second, for IDs containing deterministic conditionals, MOPs are closed only for multi-dimensional linear and quotient functions. For multi-dimensional deterministic conditionals that are described by functions that are neither linear nor quotient, the family of MOPs is not closed under transformations required for such functions. However, if such deterministic functions can be approximated by piecewise linear ones, then one can still solve such problems using our method.

Third, since our method uses MOPs to approximate PDFs and utility functions, it inherits all the limitations of MOP-based methods. For example, finding a MOP approximation of a high-dimensional conditional PDF can be difficult. Thus, if we have a continuous chance variable with many continuous chance parents, this will pose a problem for finding a MOP approximation. Shenoy [37] describes a MOP approximation of a three-dimensional CLG PDF. In this paper, we describe a procedure for finding a MOP approximation of a PDF using Lagrange interpolating polynomials with Chebyshev points. Using this procedure, we can find MOP approximations of the two-dimensional conditional log-normal PDFs needed to solve the *American Put Option* problem described in Section 5.2. In any case, we are not at a stage where one can fully automate the procedure of finding MOP approximations of

conditional PDFs and utility functions.

3 A Framework for Solving Hybrid IDs

In this section, we describe a framework and an algorithm for solving hybrid IDs with deterministic conditionals. The framework described here is a further extension of the extended Shenoy-Shafer architecture described in Shenoy and West [39] for inference in hybrid BNs with deterministic conditionals. Here, we include decision variables, and utility potentials, and we keep track of the nature of potentials (discrete, continuous, or utility) by keeping track of their units during the combination and the marginalization operations. The algorithm described is adapted from Shenoy [36] for the case of discrete IDs.

3.1 Variables and States

We are concerned with a finite set $\mathcal{V} = \mathcal{D} \cup \mathcal{C}$ of *variables*. Variables in \mathcal{D} are called *decision* variables, and variables in \mathcal{C} are called *chance* variables. Each variable $X \in \mathcal{V}$ is associated with a set Ω_X of possible *states*. If Ω_X is finite or countable, we say X is *discrete*, otherwise X is *continuous*. We will assume that the state space of continuous variables is the set of real numbers (or some measurable subset of it), and that the state space of discrete variables is a set of symbols (not necessarily real numbers). If $r \subseteq \mathcal{V}$, $r \neq \emptyset$, then $\Omega_r = \times \{\Omega_X | X \in r\}$. If $r = \emptyset$, we will adopt the convention that $\Omega_\emptyset = \{\diamond\}$.

We will distinguish between discrete and continuous chance variables. Let \mathcal{C}_d and \mathcal{C}_c denote the set of all discrete and continuous chance variables, respectively. Then, $\mathcal{C} = \mathcal{C}_d \cup \mathcal{C}_c$. We do not distinguish between discrete and continuous decision variables.

In an ID, each chance variable has a conditional distribution function for each state of its parents. A conditional distribution function associated with a chance variable is said to be *deterministic* if its variances (for each state of its parents) are all zeros. For example, suppose P (profit), R (revenue), and C (cost) are three continuous chance variables, and suppose R and C are parents of P . Furthermore, suppose the conditional of P is as follows: $P | (r, c) = r - c$ with probability 1. In this example, the conditional for P is deterministic, and we will denote it by the equation: $P = R - C$.

In an ID, we will depict decision variables by rectangular nodes, discrete chance variables by single-bordered elliptical nodes, continuous chance variables with non-deterministic conditionals by double-bordered elliptical nodes, continuous chance variables with deterministic conditionals by triple-bordered elliptical chance nodes, and additive factors of the joint utility function by diamond-shaped nodes. We do not distinguish between discrete and continuous decision variables.

An example of a hybrid ID is shown in Figure 1. This ID is a representation of the entrepreneur's problem, which will be described later in this section. In this ID, Z_1 and Z_2 are continuous chance nodes with non-deterministic conditionals, Q_n , Q_a , C_n , and C_a are continuous chance nodes with deterministic conditionals, P is a continuous decision node, and π is a utility node.

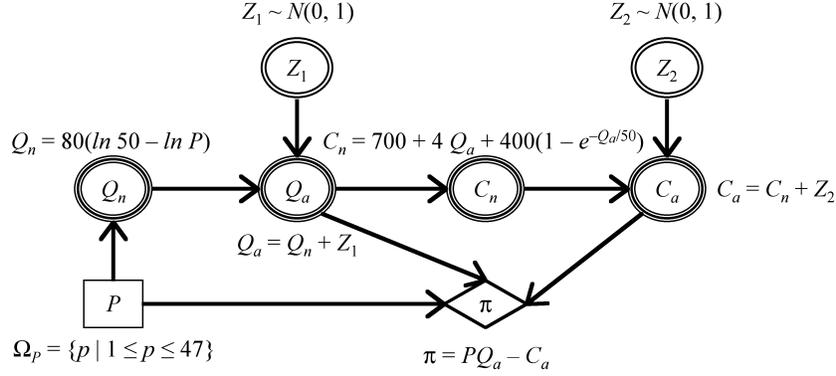


Figure 1: An Influence Diagram Representation of the Entrepreneur's Problem

3.2 Projection of States

If $\mathbf{x} \in \Omega_r$, $\mathbf{y} \in \Omega_s$, and $r \cap s = \emptyset$, then $(\mathbf{x}, \mathbf{y}) \in \Omega_{r \cup s}$. Thus, $(\mathbf{x}, \diamond) = \mathbf{x}$. Suppose $\mathbf{x} \in \Omega_r$, and $s \subseteq r$. Then, the *projection* of \mathbf{x} to s , denoted by $\mathbf{x}^{\downarrow s}$, is the state of s obtained from \mathbf{x} by dropping states of $r \setminus s$. Thus, e.g., $(w, x, y, z)^{\downarrow \{W, X\}} = (w, x)$, where $w \in \Omega_W$, and $x \in \Omega_X$. If $s = r$, then $\mathbf{x}^{\downarrow s} = \mathbf{x}$. If $s = \emptyset$, then $\mathbf{x}^{\downarrow s} = \diamond$.

3.3 Discrete Potentials

In an ID, the conditional probability functions associated with chance variables are represented by functions called potentials. If A is discrete, it is associated with a conditional probability mass function. The conditional probability mass functions are represented by functions called discrete potentials. Formally, suppose $r \subseteq \mathcal{V}$. A *discrete potential* for r is a function $\alpha: \Omega_r \rightarrow [0, 1]$ such that the values (in the interval $[0, 1]$) are in units of probability, which are dimensionless numbers without any physical units (such as feet, pounds, seconds, etc.).

Although the domain of the potential α is Ω_r , we will refer to r as the *domain* of α . Thus, the domain of a potential representing the conditional probability function associated with some chance variable X in an ID is always the set $\{X\} \cup pa(X)$, where $pa(X)$ denotes the set of parents of X in the ID graph.

The values of discrete potentials are always in units of probability. For example, suppose B is a discrete chance variable with states b (buyer) and nb (no buyer), suppose P (price in \$/bushel) is a continuous variable, and suppose β is a discrete potential for $\{B, P\}$, representing the conditional for B given P , such that $\beta(b, p) = 1/(1 + e^{-6.5+p})$, and $\beta(nb, p) = e^{-6.5+p}/(1 + e^{-6.5+p})$. The values of β are in units of probability.

3.4 Continuous Potentials

Continuous chance variables with non-deterministic conditionals are associated with conditional probability density functions (PDFs). Conditional PDFs are represented by functions called continuous potentials. Formally, suppose $r \subseteq \mathcal{V}$. A *continuous potential* ζ for r is

a function $\zeta: \Omega_r \rightarrow \mathbb{R}^+$, where \mathbb{R}^+ is the set of non-negative real numbers with units of (probability) density.

The values of continuous potentials are always in units of density. For example, suppose Y is a continuous variable whose states are in units of, say, $\text{unit}Y$, with continuous chance variable X as a parent. Suppose that the conditional associated with $Y|x$ is $N(x, 1)$. Then, the values of the continuous potential ψ for $\{X, Y\}$ such that $\psi(x, y) = (1/\sqrt{2\pi})e^{-(y-x)^2/2}$ are in units of probability per unit of Y , which is denoted simply by $(\text{unit}Y)^{-1}$.

Continuous variables with deterministic conditionals have conditionals described by equations. We will represent such conditionals by continuous potentials that use Dirac delta functions δ defined in Dirac [12].

3.5 Dirac Delta Functions

A function $\delta: \mathbb{R} \rightarrow \mathbb{R}^+$ is called a *Dirac delta function* if $\delta(x) = 0$ if $x \neq 0$, and $\int_{-\infty}^{\infty} \delta(x) dx = 1$. The values of δ are in units of density.

The Dirac delta function δ is not a proper function since the value of the function at 0 doesn't exist (i.e., is not finite). It can be regarded as a limit of a certain sequence of functions (such as, e.g., the Gaussian density function with mean 0 and variance σ^2 in the limit as $\sigma \rightarrow 0$). However, it can be used as if it were a proper function for practically all our purposes without getting incorrect results.

Although the value $\delta(0)$ (in units of density) is undefined, i.e., ∞ , we argue that we can *interpret* the value $\delta(0)$ as probability 1 at the location $x = 0$. Consider the Gaussian PDF with mean 0 and variance σ^2 . Its moment generating function (MGF) is $M(t) = e^{\sigma^2 t^2/2}$. In the limit as $\sigma \rightarrow 0$, $M(t) = 1$. Now, $M(t) = 1$ is the MGF of the degenerate probability distribution $X = 0$ with probability 1. Thus, we can interpret the value $\delta(0)$ as probability 1 at the location $x = 0$.

Some basic properties of the Dirac delta function are given in the Appendix. An example of a deterministic conditional is as follows. Suppose R (revenue in $m\$$) is a continuous chance variable with continuous chance parents P (price in $\$/bushel$), and C (crop size in $mbushels$), and discrete chance parent B with states b (buyer) and nb (no buyer). Suppose R is associated with a deterministic conditional as follows: $R = P \cdot C$ if $B = b$, and $R = 0$ if $B = nb$. Then this conditional is represented by a continuous potential ρ for $\{P, C, B, R\}$ such that $\rho(p, c, b, r) = \delta(r - p \cdot c)$, and $\rho(p, c, nb, r) = \delta(r)$. The values of ρ are in units of $(m\$)^{-1}$.

In general, if Y is a continuous variable with continuous parents $\{X_1, \dots, X_n\}$, and discrete parents $\{A_1, \dots, A_m\}$, and has a deterministic conditional $Y = g_i(X_1, \dots, X_n)$ if $(A_1, \dots, A_m) = \mathbf{a}_i$, for $i = 1, \dots, |\Omega_{\{A_1, \dots, A_m\}}|$, then such a deterministic conditional is represented by the continuous potential $\psi(\mathbf{x}, \mathbf{a}_i, y) = \delta(y - g_i(\mathbf{x}))$ for all $\mathbf{x} \in \Omega_{\{X_1, \dots, X_n\}}$, $\mathbf{a}_i \in \Omega_{\{A_1, \dots, A_m\}}$, $i = 1, \dots, |\Omega_{\{A_1, \dots, A_m\}}|$, and $y \in \Omega_Y$. The units of values of ψ are $(\text{unit}Y)^{-1}$.

3.6 Constraint Potentials

In some problems, there may be constraints on the possible states of decision variables based on states of other preceding variables. Such constraints are represented by potentials called constraint potentials. Suppose s is a set of variables such that it includes a decision variable,

say X . A *constraint potential* χ for s associated with $X \in s$ is a function $\chi: \Omega_s \rightarrow \{0, 1\}$ such that $\chi(x, \mathbf{y}) = 1$ if $x \in \Omega_X$ is a possible alternative given $\mathbf{y} \in \Omega_{s \setminus \{X\}}$, and $\chi(x, \mathbf{y}) = 0$ if not. We assume that the constraint potential is formally specified for all states of s . In practice, it is sufficient to just specify the states of s that are possible (with the rest assumed to be not possible). The values (0 or 1) of constraint potentials are in dimensionless units. Constraint potentials are used during the process of marginalizing a decision variable.

The entrepreneur's problem discussed in Section 5 has a constraint on the price variable $1 \leq p \leq 47$. This constraint is handled implicitly since we expect from the nature of the problem to find an optimal price that lies in this interval (at the two extreme prices, we expect the profits to be small or negative). In the American put option problem (also discussed in Section 5), we have constraints that are handled explicitly. These are described in Section 5.

Constraint potentials share the same units as discrete potentials, and so it is important to not confuse the two. In an ID representation, discrete potentials are associated with discrete chance variables, and constraint potentials are associated with decision variables. We will sometimes refer to the set of discrete and continuous potentials as *probability potentials*, which does not include constraint potentials.

3.7 Utility Potentials

An ID representation includes utility functions, that represent the preferences of the decision maker for the various outcomes. If an ID has more than one utility node, we assume an additive factorization of the joint utility function. Each additive factor of the utility function is represented by a utility potential. Formally, a *utility potential* v for $t \subseteq \mathcal{V}$ is a function $v: \Omega_t \rightarrow \mathbb{R}$ such that the values (in \mathbb{R}) are in units of utiles. An example of an utility potential is found in an example described below.

3.8 Summary

In summary, we can have four different kinds of potentials in IDs. The values of discrete potentials are in units of probabilities, which are dimensionless numbers (in the interval $[0, 1]$) with no physical units. The values of continuous potentials are in units of density, such as $(\text{unit}X)^{-1}$, $(\text{unit}X)^{-1} \cdot (\text{unit}Y)^{-1}$, etc. The values of utility potentials are in units of utiles. The values of constraint potentials are either 0 or 1 in dimensionless units. In the process of solving an ID, we may create potentials that have hybrid units such as utiles $\cdot (\text{unit}X)^{-1}$, etc. However, after we marginalize all chance and decision variables, we will ultimately end with a utility potential for the empty set. Details are provided in the section on solving hybrid IDs.

3.9 An Example

We will illustrate the concepts described so far using the entrepreneur's problem adapted from Howard [17]. An entrepreneur has to decide on a price for her new product. When the entrepreneur selects a price P (in \$/widget), the quantity Q_n (in m widgets) that she will sell is determined from the demand curve $Q_n(P)$. This quantity Q_n will have a total cost

of manufacturing $C_n(Q_n)$ (in $m\$$) given by the total cost curve. The entrepreneur's profit π_n (in $m\$$) will then be the difference between her revenue $P \cdot Q_n$ and her cost C_n , i.e., $\pi_n = P \cdot Q_n - C_n$. We assume that the entrepreneur is risk neutral, i.e., her utility is linear in million dollars, $u(x \text{ } m\$) = x$ utiles. The entrepreneur needs to decide on a price p that will maximize her utility.

This problem would be simple if the demand curve and total cost curve were known with certainty, but this is seldom the case. We shall assume that the quantity Q_n determined from the demand curve is only a nominal value and that the actual quantity sold will be $Q_a = Q_n + Z_1$, where Z_1 (in m widgets) is a standard normal random variable. Furthermore, producing this quantity Q_a will cost $C_a = C_n(Q_a) + Z_2$, where Z_2 (in $m\$$) is another independent standard normal random variable. Note that the profit π (in $m\$$) is now $\pi = P \cdot Q_a - C_a$.

For the demand curve, the functional form is $Q_n(p) = (\ln \alpha - \ln p)/\beta$, where $p \leq \alpha$, and the constants are given by $\alpha = 50$, $\beta = 1/80$. This is a decreasing function—at a price of $\$1/\text{widget}$, she would sell $80 \cdot \ln 50 \approx 313$ m widgets, and at a price of $\$50/\text{widget}$, she would sell none. For the total cost function we assume the form $C_n(q_a) = k_0 + k_1 q_a + k_2(1 - e^{-k_3 q_a})$ with constants $k_0 = 700$, $k_1 = 4$, $k_2 = 400$, $k_3 = 1/50$. The total cost function is an increasing function, but at a decreasing rate. We restrict the range of P to make sure that Q_a is nonnegative. An ID representation of the problem is depicted in Figure 1.

The potentials in this example are as follows. We start with the name of the potential, its domain, details of the potential, and its units.

1. χ_a for $\{C_n, Z_2, C_a\}$ such that $\chi_a(c_n, z_2, c_a) = \delta(c_a - (c_n + z_2))$, $(m\$)^{-1}$.
2. φ_2 for Z_2 such that $\varphi_2(z_2) = (1/\sqrt{2\pi})e^{-z_2^2/2}$, $(m\$)^{-1}$.
3. χ_n for $\{Q_a, C_n\}$ such that $\chi_n(q_a, c_n) = \delta(c_n - (700 + 4q_a + 400(1 - e^{-q_a/50})))$, $(m\$)^{-1}$.
4. θ_a for $\{Q_n, Z_1, Q_a\}$ such that $\theta_a(q_n, z_1, q_a) = \delta(q_a - (q_n + z_1))$, $(m\text{widgets})^{-1}$.
5. φ_1 for Z_1 such that $\varphi_1(z_1) = (1/\sqrt{2\pi})e^{-z_1^2/2}$, $(m\text{widgets})^{-1}$.
6. θ_n for $\{P, Q_n\}$ such that $\theta_n(p, q_n) = \delta(q_n - 80(\ln 50 - \ln p))$, $(m\text{widgets})^{-1}$.
7. π for $\{P, Q_a, C_a\}$ such that $\pi(p, q_a, c_a) = p \cdot q_a - c_a$, utiles

It is evident from the units that the first six potentials are continuous potentials and the seventh is a utility potential. There is no potential associated with decision variable P . A valuation network (VN) representation ([36]) (also called a factor graph in Kschischang et al. [24]) of the entrepreneur's problem is shown in Figure 2. A VN is a bi-partite graph with variables and potentials as nodes. Variable nodes are depicted just as in IDs. Potential nodes are depicted by diamond-shaped nodes. Probability and utility potentials are depicted by single-bordered diamond-shaped nodes. Constraint potentials are depicted by double-bordered diamond-shaped nodes. Each potential has an edge between it and the variables in its domain. During the solution phase, we switch from the ID to the VN representation as there are no guarantees that the ID representation will be maintained at each step of the solution algorithm ([36]). In Section 5, we describe a solution to this problem.

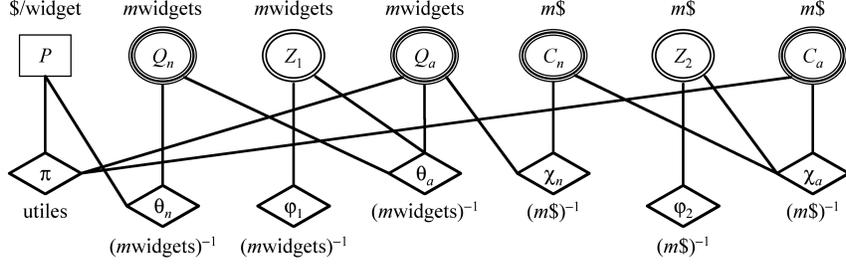


Figure 2: A VN Representation of the Entrepreneur's Problem

3.10 Combination of Potentials

The definition of combination of potentials depends on the units of the potentials being combined. Although there are many possible combinations of units, we have only two distinct definitions. Utility functions are additive factors of the joint utility function. Thus, combination of two utility potentials (both in units of utiles) involves pointwise addition. In all other cases, combination of potentials involves pointwise multiplication. Thus, in problems where we have a single utility node, combination is always pointwise multiplication.

Suppose v_1 and v_2 are utility potentials for t_1 and t_2 , respectively. Then, the combination of v_1 and v_2 , denoted by $v_1 \otimes v_2$, is a utility potential for $t_1 \cup t_2$ defined as follows:

$$(v_1 \otimes v_2)(\mathbf{x}) = v_1(\mathbf{x}^{\downarrow t_1}) + v_2(\mathbf{x}^{\downarrow t_2}) \quad \text{for all } \mathbf{x} \in \Omega_{t_1 \cup t_2}. \quad (3.1)$$

The units of $(v_1 \otimes v_2)$ are utiles.

Suppose α_1 and α_2 are potentials for t_1 and t_2 , respectively, such that α_1 and α_2 are not both utility. Then, the combination of α_1 and α_2 , denoted by $\alpha_1 \otimes \alpha_2$, is a potential for $t_1 \cup t_2$ defined as follows:

$$(\alpha_1 \otimes \alpha_2)(\mathbf{x}) = \alpha_1(\mathbf{x}^{\downarrow t_1}) \cdot \alpha_2(\mathbf{x}^{\downarrow t_2}) \quad \text{for all } \mathbf{x} \in \Omega_{t_1 \cup t_2}. \quad (3.2)$$

The units of $(\alpha_1 \otimes \alpha_2)$ are the product of the units of α_1 and α_2 . Thus, e.g., if α_1 is discrete and α_2 is utility (or vice versa), then $\alpha_1 \otimes \alpha_2$ is utility; and if α_1 is continuous and α_2 is utility (or vice versa), then $\alpha_1 \otimes \alpha_2$ will have hybrid units such as $\text{utiles} \cdot (\text{unit } X)^{-1}$, etc.

Observe that combination of potentials is non-associative. Thus, if σ is a discrete or continuous potential, and v_1 and v_2 are utility potentials, then $\sigma \otimes (v_1 \otimes v_2) \neq (\sigma \otimes v_1) \otimes v_2$. This non-associativity of combination will necessitate divisions if we wish to use local computation ([36]). This will be discussed further in the section on solving hybrid IDs.

3.11 Marginalization of Potentials

In the process of solving an ID, we marginalize chance and decision variables in some sequence that is dictated by the information constraints. Before we marginalize a variable, we may have to do some combination and division operations prior to marginalization. In this subsection, we define just the marginalization operation without describing the details of how the potential being marginalized is obtained. The details of the solution algorithm are described after we have completed all requisite definitions.

The definition of marginalization of potentials depends on the nature of the variable being marginalized. We marginalize discrete chance variables by addition over its state space, continuous chance variables by integration over its state space, and decision variables (discrete or continuous) by maximization over its state space, which may be further constrained by constraint potentials.

Suppose α is a potential for a , and suppose $X \in a$ is a discrete variable. Then, the marginal of α by deleting X , denoted by α^{-X} , is a potential for $a \setminus \{X\}$ given as follows:

$$\alpha^{-X}(\mathbf{y}) = \sum_{x \in \Omega_X} \alpha(x, \mathbf{y}) \quad \text{for all } \mathbf{y} \in \Omega_{a \setminus \{X\}}. \quad (3.3)$$

In this case, the units of α^{-X} are exactly the same as the units of α .

If $X \in a$ is a continuous variable, then α^{-X} is defined as follows:

$$\alpha^{-X}(\mathbf{y}) = \int_{-\infty}^{\infty} \alpha(x, \mathbf{y}) dx \quad \text{for all } \mathbf{y} \in \Omega_{a \setminus \{X\}}. \quad (3.4)$$

In this case, the units of α^{-X} are the units of α multiplied by the units of X .

And if $X \in a$ is a decision variable, then α^{-X} is defined as follows:

$$\alpha^{-X}(\mathbf{y}) = \max_{x \in \Omega_X} \alpha(x, \mathbf{y}) \quad \text{for all } \mathbf{y} \in \Omega_{a \setminus \{X\}}. \quad (3.5)$$

In this case, the units of α^{-X} are exactly the same as the units of α . If we have a constraint potential χ for s associated with $X \in s$, then we assume that χ is already included in α (so that $s \subseteq a$), and the maximization in Equation (3.5) is over $x \in \Omega_X$ such that $\chi(x, \mathbf{y}^{\downarrow s \setminus \{X\}}) = 1$.

3.12 Division of Potentials.

The process of solving an ID may involve division of discrete or continuous potentials by discrete or continuous potentials. Also, the potential in the divisor is always a marginal of the potential being divided.

Suppose α is a discrete or continuous potential for a , and suppose $X \in a$ is a discrete or continuous chance variable. Then the division of α by α^{-X} , denoted by $\alpha \oslash \alpha^{-X}$, is a potential for a defined as follows:

$$(\alpha \oslash \alpha^{-X})(x, \mathbf{y}) = \alpha(x, \mathbf{y}) / \alpha^{-X}(\mathbf{y}) \quad \text{for all } x \in \Omega_X, \text{ and } \mathbf{y} \in \Omega_{a \setminus \{X\}} \quad (3.6)$$

In Equation (3.6), if the denominator is zero, then the numerator is also 0, and in this case we define 0/0 as 0. The units of the potential $\alpha \oslash \alpha^{-X}$ are the units of α divided by the units of α^{-X} . For the division operations that are done in the process of solving an ID (described in the next subsection) it can be shown that $\alpha \oslash \alpha^{-X}$ represents the conditional for X given variables in $a \setminus \{X\}$. Thus, if X is discrete, then $\alpha \oslash \alpha^{-X}$ is discrete, and if X is continuous, $\alpha \oslash \alpha^{-X}$ is continuous in units of $(\text{unit } X)^{-1}$ ([6]).

3.13 An Algorithm for Solving Hybrid Influence Diagrams.

We have all the definitions needed to solve hybrid IDs with deterministic conditionals. The solution algorithm is basically the same as described in Shenoy [36] and Lauritzen and Nilsson [26] for discrete IDs. The details of the solution algorithm are as follows.

First, all variables need to be marginalized in a sequence that respects the information constraints in the sense that if X precedes Y in the information sequence, then Y must be marginalized before X . In a well-defined ID, the information constraints form a partial order such that if C is a chance variable, and D is a decision variable, exactly one of the following information constraints must hold: either C precedes D , or D precedes C . In the former case, the true value of C is known by the decision-maker prior to choosing a state of D , and in the latter case, the true value of C is not known at the time the decision-maker has to choose a state of D .

First, we describe the general case where we have an additive factorization of the joint utility function. In this case, divisions may be required. Next, we describe some special cases where divisions can be avoided.

We start with a set of potentials included in an ID representation. These potentials get modified in the process of marginalization.

3.14 Marginalizing a Chance Variable-Case 1

Suppose we have to marginalize a chance variable C . First, we combine all probability potentials whose domains include C , resulting in the potential, say χ . Next, we compute the marginal χ^{-C} . Then, we compute the quotient $(\chi \oslash \chi^{-C})$. The set of all probability potentials whose domains include C are replaced by the potentials χ^{-C} and $(\chi \oslash \chi^{-C})$. The units of $(\chi \oslash \chi^{-C})$ are units of probability if C is discrete, and $(\text{unit}C)^{-1}$ if C is continuous. The operations described so far is equivalent to the operations involved in arc-reversal ([30]). Next, we combine all utility potentials that include C in their domains, resulting in utility potential, say v . The set of all utility potentials that include C in their domains is now replaced by the potential v . Next, we replace v and $(\chi \oslash \chi^{-C})$ by the potential $(v \otimes (\chi \oslash \chi^{-C}))^{-C}$, which must be a utility potential. This concludes the end of the process of marginalizing C . After marginalizing chance variable C , there will not be any potentials that include C in their domains.

3.15 Marginalizing a Decision Variable-Case 1

Suppose we have to marginalize decision variable D . First we combine all utility potentials that include D in their domains, and then we combine the resulting utility potential with constraint potentials for D if any, resulting in utility potential, say v . Next, we marginalize D from v . All utility and constraint potentials that include D in their domains are now replaced by v^{-D} . In the process of marginalizing D from v , we keep track of where the maximum is attained (as a function of the remaining variables in the domain of v). This yields a *decision function* for the decision variable. The collection of all decision functions constitutes an *optimal strategy* for the ID.

After all variables have been marginalized, we end up with a single utility potential for the empty set, whose value represents the optimal utility associated with an optimal strategy.

This general algorithm described above involves divisions in the process of marginalizing a chance variable. This step may be simplified in the case where we have a single utility potential as follows.

3.16 Marginalizing a Chance/Decision Variable-Case 2

Suppose we have to marginalize a (chance or decision) variable X . First, we combine all potentials that include X in their domains, resulting in potential, say, v , and then marginalize X from v . The set of all potentials that include X in their domains is replaced by v^{-X} . In this case, we cannot predict the nature of v^{-X} , i.e., it may have hybrid units.

Notice that there are no divisions involved in this process. When we have a single utility factor, combination always involves multiplication, which is associative, and it follows from the axiomatic approach of Shenoy and Shafer [38] that we can find marginals without doing any divisions. The process of solving an ID can be described as finding the marginal for the empty set by sequentially marginalizing all variables in a sequence that respects the information constraints. The first example (entrepreneur's problem) solved in Section 5 has a single utility factor, and thus, no divisions are required.

Another special case where no divisions are necessary is as follows. In the process of marginalizing a chance variable C , suppose that there is only one probability (discrete or continuous) potential, say χ that includes C in its domain. In this case χ must be the conditional for C given its parents, $pa(C)$. Thus, χ^{-C} is an identity potential for $pa(C)$ (whose values are all 1's). In this case also, we can skip the divisions. If this is true for all chance variables C (this will happen if the arcs into each chance variable are consistent with the partial order representing the information constraints, and we pick a deletion sequence consistent with all arcs in the ID), then we can use the rules described in Case 2 above. The second example (American put option problem) solved in Section 5 is an example of this type and no divisions are required.

Finally, we remark that one can always avoid divisions by combining all utility potentials and replacing the set of all utility potentials by the combination. This may, however, increase the computational effort of solving an ID as the domain of the single joint utility potential will have all decision variables in its domain and potentially could be large. Shenoy [36] has described a small example where divisions are inescapable assuming that we wish to use local computation and avoid computing on the domain of all variables.

The algorithm described in this subsection is illustrated in Section 5 by solving two small hybrid IDs in complete detail.

4 Mixture of Polynomials Functions

In this section, we define MOP functions, describe some methods for finding MOP approximations of univariate and two-dimensional conditional PDFs, and piecewise-linear approximations of nonlinear deterministic functions. We illustrate our method for the log-normal distribution. Shenoy and West [40] describes MOP approximations of the PDFs of the normal

and chi-square univariate distributions, and CLG distributions in two dimensions. Shenoy [37] describes MOP approximations of the CLG PDFs in one, two and three dimensions.

4.1 MOP Functions

The definitions of one-dimensional, and multi-dimensional MOP functions is taken from Shenoy [37].

A one-dimensional function $f : \mathbb{R} \rightarrow \mathbb{R}$ is said to be a *mixture of polynomials* (MOP) function if it is a piecewise function of the form:

$$f(x) = \begin{cases} a_{0i} + a_{1i}x + \cdots + a_{ni}x^n & \text{for } x \in A_i, i = 1, \dots, k, \\ 0 & \text{otherwise.} \end{cases} \quad (4.1)$$

where A_1, \dots, A_k are disjoint intervals in \mathbb{R} that do not depend on x , and a_{0i}, \dots, a_{ni} are constants for all i . We will say that f is a k -piece (ignoring the 0 piece), and n -degree (assuming $a_{ni} \neq 0$ for some i) MOP function.

An example of a 2-piece, 3-degree MOP function $g_1(\cdot)$ in one-dimension is as follows:

$$g_1(x) = \begin{cases} 0.41035 + 0.09499x - 0.09786x^2 - 0.02850x^3 & \text{if } -3 < x < 0, \\ 0.41035 - 0.09499x - 0.09786x^2 + 0.02850x^3 & \text{if } 0 \leq x < 3 \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

$g_1(\cdot)$ is a MOP approximation of the PDF of the standard normal distribution on the domain $(-3, 3)$, and was found using Lagrange interpolating polynomials with Chebyshev points, which will be discussed in the next subsection.

The main motivation for defining MOP functions is that such functions are easy to integrate in closed form, and the family of MOP functions is closed under multiplication, addition, integration, the main operations in solving hybrid IDs. Also, since MOP functions are easily differentiable, it is easy to maximize MOP functions in closed form.

A multivariate polynomial is a polynomial in several variables. For example, a polynomial in two variables is as follows:

$$P(x_1, x_2) = a_{00} + a_{10}x_1 + a_{01}x_2 + a_{11}x_1x_2 + a_{20}x_1^2 + a_{02}x_2^2 + a_{21}x_1^2x_2 + a_{12}x_1x_2^2 + a_{22}x_1^2x_2^2 \quad (4.3)$$

The degree of the polynomial in Equation (4.3) is 4 assuming a_{22} is a non-zero constant. In general, the degree of a multivariate polynomial is the largest sum of the exponents of the variables in the terms of the polynomial.

An m -dimensional function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is said to be a MOP function if

$$f(x_1, x_2, \dots, x_m) = \begin{cases} P_i(x_1, x_2, \dots, x_m) & \text{for } (x_1, x_2, \dots, x_m) \in A_i, i = 1, \dots, k, \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

where $P_i(x_1, x_2, \dots, x_m)$ are multivariate polynomials in m variables for all i , and the disjoint regions A_i are as follows. Suppose π is a permutation of $\{1, \dots, m\}$. Then each A_i is of the

form:

$$\begin{aligned}
l_{1i} &\leq x_{\pi(1)} \leq u_{1i}, \\
l_{2i}(x_{\pi(1)}) &\leq x_{\pi(2)} \leq u_{2i}(x_{\pi(1)}), \\
&\vdots \\
l_{mi}(x_{\pi(1)}, \dots, x_{\pi(m-1)}) &\leq x_{\pi(m)} \leq u_{mi}(x_{\pi(1)}, \dots, x_{\pi(m-1)})
\end{aligned} \tag{4.5}$$

where l_{1i} and u_{1i} are constants, and $l_{ji}(x_{\pi(1)}, \dots, x_{\pi(j-1)})$ and $u_{ji}(x_{\pi(1)}, \dots, x_{\pi(j-1)})$ are linear functions of $x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(j-1)}$ for $j = 2, \dots, m$, and $i = 1, \dots, k$. We will refer to the nature of the region described in Equation (4.5) as a *hyper-rhombus*. Although we have defined the hyper-rhombus as a closed region in Equation (4.5), each of the $2m$ inequalities can be either strictly $<$ or \leq . Notice that the definition of the region A_i in the m -dimensional case (in Equation 4.5) is a generalization of the requirement in the 1-dimensional case (Equation 4.1) that the regions A_i are intervals.

A special case of the hyper-rhombus region A_i is a region of the form:

$$l_{1i} \leq x_1 \leq u_{1i}, \quad l_{2i} \leq x_2 \leq u_{2i}, \quad \dots, \quad l_{mi} \leq x_m \leq u_{mi} \tag{4.6}$$

where $l_{1i}, \dots, l_{mi}, u_{1i}, \dots, u_{mi}$ are all constants. We refer to the region defined in Equation 4.6 as a *hypercube* (in m -dimensions).

An example of a 2-piece, 3-degree MOP $g_2(\cdot, \cdot)$ defined on a two-dimensional hyper-rhombus is as follows:

$$g_2(x, y) = \begin{cases} 0.41035 + 0.09499(y - x) - 0.09786(y - x)^2 - 0.02850(y - x)^3 & \text{if } x - 3 < y < x, \\ 0.41035 - 0.09499(y - x) - 0.09786(y - x)^2 + 0.02850(y - x)^3 & \text{if } x \leq y < x + 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

$g_2(x, y)$ is a two-dimensional MOP approximation of the PDF of the CLG distribution of $Y|x \sim N(x, 1^2)$ on the domain $-\infty < x < \infty, x - 3 < y < x + 3$. Notice that $g_2(x, y) = g_1(y - x)$, where $g_1(\cdot)$ is as defined in Equation 4.2.

4.1.1 Advantages of Hyper-rhombus Regions

One advantage of defining multi-dimensional MOP functions on hyper-rhombuses is that MOP functions are closed under transformations needed for multi-dimensional linear deterministic conditionals. For example consider the case where X, Y , and Z are continuous variables, where X has PDF $f_X(x)$, $Y|x$ has conditional PDF $f_{Y|x}(y)$, and Z has a deterministic conditional $Z = X + Y$, which is represented by the function $\delta(z - x - y)$, where δ is the Dirac delta function. Suppose that $f_X(x)$ is a 1-dimensional MOP function, and suppose that $f_{Y|x}(y)$ is a 2-dimensional MOP function (in x and y) defined on hypercubes. Suppose we wish to find the marginal of Z . After we marginalize Y (by computing $\int_{-\infty}^{\infty} f_{Y|x}(y) \delta(z - x - y) dy$), we obtain the function $f_{Y|x}(z - x)$. Notice that even though $f_{Y|x}(y)$ was defined on hypercubes, $f_{Y|x}(z - x)$ is no longer defined on hypercubes since we now have regions such as $l_{1i} \leq z - x \leq u_{1i}$, which is a hyper-rhombus.

Another advantage is that we can obtain MOP approximations of CLG PDFs from a MOP approximation of the univariate standard normal PDF ([37]). For example, suppose

$g_1(x)$ is a MOP approximation of the PDF of $N(0, 1^2)$. Now suppose $Y|x \sim N(ax + b, \sigma^2)$, where a , b , and σ are constants, and $\sigma \neq 0$. We can find a MOP approximation of the PDF of $Y|x$ as follows:

$$h(x, y) = \frac{1}{|\sigma|} g_1\left(\frac{y - ax - b}{\sigma}\right) \quad (4.8)$$

Notice that even though $g_1(x)$ is defined on hypercubes, $h(x, y)$ is no longer defined on hypercubes (since we now have regions such as $l_{1i} \leq \frac{y-ax-b}{\sigma} \leq u_{1i}$). However, $h(x, y)$ is defined on a hyper-rhombus region, and therefore is a MOP. The MOP function $g_2(x, y)$ described in Equation 4.7 is an instance of $h(x, y)$ when $a = 1$, $b = 0$, and $\sigma = 1$.

Finally, the hyper-rhombus region allows us to find MOP approximations of conditional PDFs using fewer pieces and lower degrees. Using hypercubes, we were able to find a 16-piece, 18-degree MOP approximation of a conditional log-normal PDF. Using hyper-rhombuses, we found an 8-piece, 5-degree MOP approximation for the same conditional log-normal PDF. This is because in the hyper-rhombus case, we can truncate the region where the PDF has very small values, and thus avoid the high degree necessitated by the non-negativity condition of PDFs.

There are some disadvantages associated with hyper-rhombus regions compared to hypercubes. MOPs defined on hyper-rhombuses take longer to integrate. After integration, MOPs defined on hyper-rhombuses may have higher degrees. Some comparisons of hyper-rhombuses versus hypercubes appear in Shenoy [37], and Shenoy et al. [35].

4.2 Finding MOP Approximations of Univariate PDFs

In this subsection, we will describe a process for finding a MOP approximation of a univariate PDF using Lagrange interpolating polynomials with Chebyshev points. In the next section, we will work with log-normal PDFs. Therefore, we will use the log-normal distribution for illustration purposes.

4.2.1 Lagrange Interpolating Polynomials

Suppose we need to fit a polynomial for a one-dimensional function $f(x)$ in some interval (a, b) . Given a set of n points $\{(x_1, f(x_1)), \dots, (x_n, f(x_n))\}$, the Lagrange interpolating polynomial (LIP) $P(x)$ is given by:

$$P(x) = \sum_{j=1}^n \left[f(x_j) \prod_{k=1, k \neq j}^n \frac{x - x_k}{x_j - x_k} \right] \quad (4.9)$$

The polynomial $P(x)$ has the following properties ([3]). It is a polynomial of degree $\leq (n - 1)$ that passes through the n points $\{(x_1, f(x_1)), \dots, (x_n, f(x_n))\}$, i.e., $P(x_j) = f(x_j)$ for $j = 1, \dots, n$. If $f(x)$ is continuous and $(n + 1)$ -times differentiable in an interval (a, b) , and x_1, \dots, x_n are distinct points in (a, b) such that $x_1 < \dots < x_n$, then for each $x \in (a, b)$, there exists a number $\xi(x)$ (generally unknown) between x_1 and x_n such that

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{n!} (x - x_1)(x - x_2) \cdots (x - x_n) \quad (4.10)$$

4.2.2 Chebyshev Points

One question in the use of LIP is the choice of the points x_1, \dots, x_n . For an interval (a, b) where $b > a$, the n Chebyshev points are given by

$$x_i = \frac{1}{2}(a + b) + \frac{1}{2}(b - a) \cos\left(\frac{2i - 1}{2n}\pi\right), \text{ for } i = 1, \dots, n \quad (4.11)$$

The Chebyshev points are often used with LIP because the resulting polynomial approximation $P(x)$ minimizes the quantity $|(x - x_1) \cdots (x - x_n)|$ for all $x \in (a, b)$, which is proportional to the absolute error between the function $f(x)$ and the LIP $P(x)$ (see Equation (4.10)). The minimum value of $|(x - x_1) \cdots (x - x_n)|$ is $\frac{1}{2^{n-1}}$. Thus, as n increases, the maximum absolute deviation decreases.

4.2.3 Finding a MOP Approximation for a PDF

The construction of a k -piece, n -degree MOP $g(x)$ that approximates a PDF $f(x)$ on some domain (l, u) proceeds as follows. First, we compute the LIP polynomial, say $g_u(x)$, for $f(x)$ using $n = 3$ Chebyshev points for the domain (l, u) . Second, we check to see if $g_u(x)$ is non-negative over the entire domain (by computing the minimum of $g_u(x)$ over the entire domain and making sure it is positive). If not, we increase n until we obtain non-negativity. Since we are using Chebyshev points, we are guaranteed to obtain non-negativity for some n assuming $f(x) > 0$ for $x \in (l, u)$. If the smallest degree n for which we obtain non-negativity is too high (> 5 , e.g. for a 1-dimensional MOP), then we partition the domain into more pieces and restart. Third, we normalize the fitted polynomial $g_u(x)$ so that it integrates to 1.

The procedure described in the previous paragraph can be applied to any PDF, including, e.g., the class of quantile-parameterized distributions described in Keelin and Powley [22]. We will apply the above procedure for a log-normal distribution. Suppose $X \sim N(\mu, \sigma^2)$ and $Y = e^X$. Then, we say Y has the log-normal distribution with parameters μ and σ^2 , written as $Y \sim LN(\mu, \sigma^2)$. First, we need to decide on the precision of the MOP approximation. The exact domain of the PDF of Y is $(0, \infty)$. For the standard normal distribution, the domain $(-3, 3)$ covers 99.7% of the total probability. Thus, we can approximate the PDF of Y on the domain $(e^{\mu-3\sigma}, e^{\mu+3\sigma})$. If we need greater precision, we can approximate the PDF of Y on a larger domain, e.g., on $(e^{\mu-4\sigma}, e^{\mu+4\sigma})$, which captures more than 99.99% of the total probability.

Suppose $S_1 \sim LN(\mu, \sigma^2)$ where $\mu = \ln(40) + 0.00074$ and $\sigma^2 = 0.13229^2$ (these constants are obtained from the American put option example described in Section 5.2). The 0.15 percentile of the PDF of S_1 is 27.03, and the 99.85 percentile is 59.28. If we try to fit a 1-piece MOP approximation of the PDF of S_1 using the above procedure, the result is a 8-degree MOP on the domain $(27.03, 59.28)$. So we partition the domain into two pieces $(27.03, 39.34)$ and $[39.34, 59.28)$, where 39.34 is the mode of S_1 ($= e^{\mu-\sigma^2}$). In this case, we find a 2-piece, 5-degree MOP $\phi_{p1}(s_1)$. A graph of the MOP approximation $\phi_{p1}(s_1)$ overlaid on the actual PDF $\phi_1(s_1)$ truncated to $(27.03, 59.28)$ is shown in Figure 3 and it shows that there are not much differences between the two functions. The mean of $\phi_{p1}(s_1)$ is ≈ 40.370 , and the mean of $\phi_1(s_1) \approx 40.371$. The variance of $\phi_{p1}(s_1)$ is ≈ 27.875 and the variance of $\phi_1(s_1) \approx 27.879$.

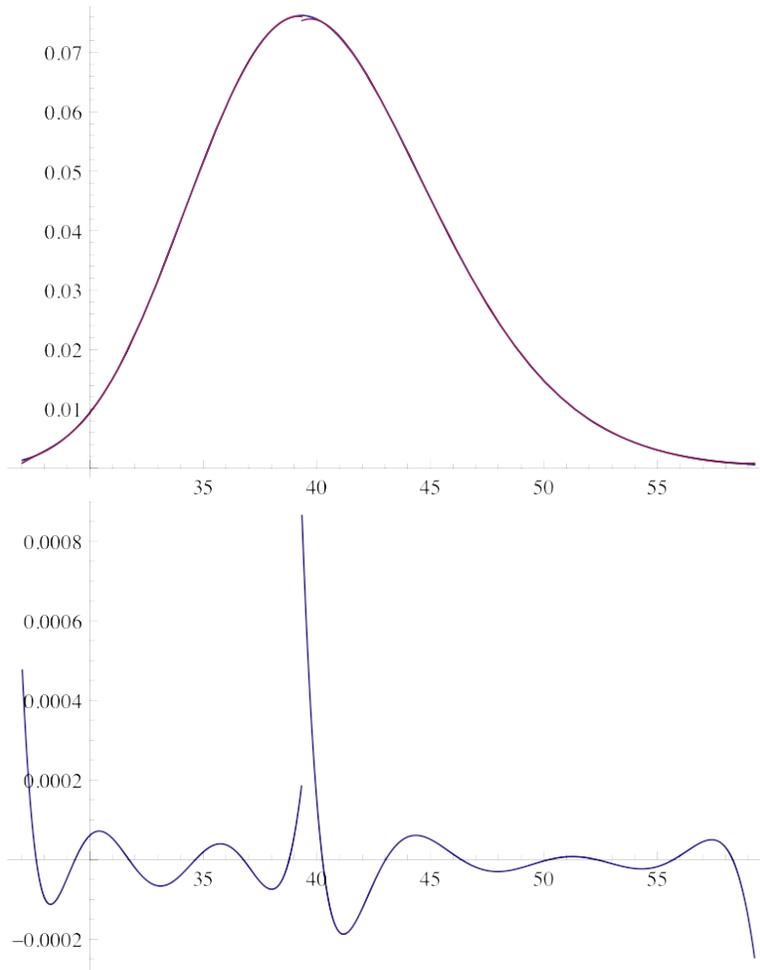


Figure 3: A Graph of ϕ_{p1} (in red) Overlaid on ϕ_1 (in blue); the Bottom Graph Shows the Difference $\phi_1 - \phi_{p1}$ in the Curves

The LIP method described for univariate PDFs can also be applied for 2-dimensional conditional PDFs. A generalization of Lagrange interpolating polynomials exists for two and higher dimensions and its implementations exist in commercial software such as Mathematica and Maple. Chebyshev points are also defined in closed form for two-dimensional regions in Xu [43].

4.3 Finding Piecewise-Linear Approximations of Deterministic Conditionals

In this subsection, we describe finding piecewise-linear approximations of deterministic conditionals. MOP functions are closed under transformation required for multi-dimensional linear and quotient functions. Thus, if we have a nonlinear deterministic conditional (as we do in the entrepreneur's problem), then we need to approximate such conditionals by piecewise linear functions.

Consider the nominal demand function f_{Q_n} as a function of price p in the range $1 \leq p \leq 47$

given as follows:

$$f_{Q_n}(p) = 80(\ln 50 - \ln p) \quad \text{if } 1 \leq p \leq 47 \quad (4.12)$$

Since $f_{Q_n}(p)$ is not a linear function, the deterministic conditional $Q_n = f_{Q_n}(P)$ (associated with Q_n) is not linear. Thus, we need to approximate $f_{Q_n}(p)$ by a piecewise linear function. We select a set of intermediate points in the interval $(1, 47)$ and find a piecewise linear approximation $f_{pQ_n}(p)$ of this one-dimensional function as follows:

$$f_{pQ_n}(p) = \begin{cases} 225.073 - 43.9445(p - 3) & \text{if } 1 \leq p < 3 \\ 157.289 - 16.946(p - 7) & \text{if } 3 \leq p < 7 \\ 107.766 - 8.25386(p - 13) & \text{if } 7 \leq p < 13 \\ 69.4 - 4.79573(p - 21) & \text{if } 13 \leq p < 21 \\ 28.534 - 2.919(p - 35) & \text{if } 21 \leq p < 35 \\ 4.95003 - 1.96533(p - 47) & \text{if } 35 \leq p \leq 47 \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

The number of intermediate points and the location of the points was selected by trial and error. A graph of $f_{pQ_n}(p)$ vs. p overlaid on the graph of $f_{Q_n}(p)$ vs. p is shown in Figure 4. The maximum absolute percentage deviation between $f_{Q_n}(p)$ and $f_{pQ_n}(p)$ is 5.3% at $p = 43.5$.

5 Two Examples

In this section, we illustrate our framework and algorithm for solving hybrid IDs with deterministic conditionals by solving two problems. The first one is the entrepreneur's problem described in section 3, and has continuous chance and deterministic conditionals, a continuous decision variable, and one (unfactored) utility function. The second problem is an American put option described in Charnes and Shenoy [5]. This problem has continuous chance variables, discrete decision variables with continuous chance predecessors, and an additive factorization of the utility function. For some of the complicated marginalization operations, we report the approximate time it takes Mathematica to do the operation (using the *Timing* command in Mathematica). We used Mathematica version 7.0.1 on a MacBook Pro laptop computer to do the computations.

5.1 Entrepreneur's Problem

We will solve the entrepreneur's problem by marginalizing variables in the following sequence: $C_a, Z_2, C_n, Q_a, Z_1, Q_n, P$. To avoid the integration and optimization problems, we will approximate the continuous potentials associated with Z_1 and Z_2 by MOP potentials φ_{p1} and φ_{p2} , respectively, and the nonlinear deterministic functions associated with Q_n and C_n by piecewise-linear functions f_{pQ_n} and f_{pC_n} , respectively. Since we have a single utility potential, no divisions are necessary during the solution process.

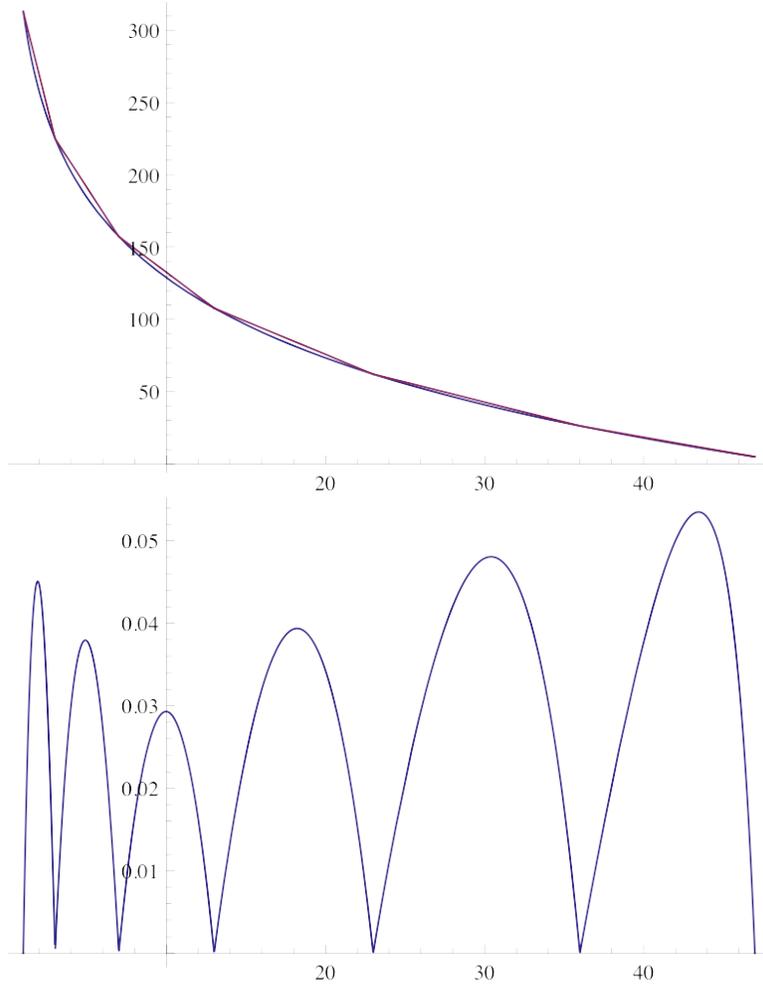


Figure 4: A Graph of $f_{pQ_n}(p)$ (in red) Overlaid on $f_{Q_n}(p)$ (in blue); the Bottom Graph Shows the Difference $f_{pQ_n}(p) - f_{Q_n}(p)$ in the Curves

5.1.1 Marginalizing C_a

First, we marginalize C_a . C_a is in the domain of potentials χ_a and π . Let potential π_1 denote $(\chi_a \otimes \pi)^{-C_a}$. Then,

$$\begin{aligned}
 \pi_1(p, q_a, c_n, z_2) &= (\chi_a \otimes \pi)^{-C_a}(p, q_a, c_n, z_2) \\
 &= \int_{-\infty}^{\infty} \delta(c_a - (c_n + z_2)) \cdot (p \cdot q_a - c_a) dc_a \\
 &= p \cdot q_a - (c_n + z_2), \quad (\text{utiles})
 \end{aligned} \tag{5.1}$$

The result in Equation (5.1) follows from property 1 of Dirac delta functions.

5.1.2 Marginalizing Z_2

Next, we marginalize Z_2 . Z_2 is in the domain of potentials φ_{p2} and π_1 . Let $\varphi_{p2}(z)$ denote the 2-piece, 3-degree MOP approximation of $\varphi_2(z)$, the PDF associated with the standard

normal distribution, as described in Equation 4.2.

Let π_2 denote $(\varphi_{p2} \otimes \pi_1)^{-Z_2}$. Details of π_2 are as follows:

$$\begin{aligned}
\pi_2(p, q_a, c_n) &= (\varphi_{p2} \otimes \pi_1)^{-Z_2}(p, q_a, c_n) \\
&= \int_{-\infty}^{\infty} \varphi_{p2}(z_2) \cdot (p \cdot q_a - (c_n + z_2)) \, dz_2 \\
&= p \cdot q_a - c_n \quad (\text{utils})
\end{aligned} \tag{5.2}$$

5.1.3 Marginalizing C_n

Next we marginalize C_n . C_n is in the domain of potentials χ_n and π_2 . Let f_{pC_n} denote a 3-piece piecewise-linear approximation of the cost function f_{C_n} as follows:

$$f_{pC_n}(q_a) = \begin{cases} 705.1 + 9.29q_a & \text{if } 2 \leq q_a < 42 \\ 844.19 + 5.98q_a & \text{if } 42 \leq q_a < 104 \\ 1025.87 + 4.23q_a & \text{if } 104 \leq q_a \leq 316 \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

The maximum absolute percentage deviation between f_{pC_n} and f_{C_n} is 2.3% at $q_a = 18.64$. The Dirac potential associated with C_n is $\chi_n(c_n, q_a) = \delta(c_n - f_{pC_n}(q_a))$. Let π_3 denote $(\chi_n \otimes \pi_2)^{-C_n}$. Details of π_3 are as follows.

$$\begin{aligned}
\pi_3(p, q_a) &= (\chi_n \otimes \pi_2)^{-C_n}(p, q_a) \\
&= \int_{-\infty}^{\infty} (p q_a - c_n) \cdot \delta(c_n - f_{pC_n}(q_a)) \, dc_n \\
&= p q_a - f_{pC_n}(q_a) \quad (\text{utils})
\end{aligned} \tag{5.4}$$

5.1.4 Marginalizing Q_a

Next, we marginalize Q_a . Q_a is in the domain of potentials θ_a and π_3 . Let π_4 denote $(\theta_a \otimes \pi_3)^{-Q_a}$. Details of π_4 are as follows.

$$\begin{aligned}
\pi_4(p, q_n, z_1) &= (\theta_a \otimes \pi_3)^{-Q_a}(p, q_n, z_1) \\
&= \int_{-\infty}^{\infty} \delta(q_a - (q_n + z_1)) \cdot (p \cdot q_a - f_{pC_n}(q_a)) \, dq_a \\
&= p \cdot (q_n + z_1) - f_{pC_n}(q_n + z_1) \quad (\text{utils})
\end{aligned} \tag{5.5}$$

Notice that π_4 is a MOP function.

5.1.5 Marginalizing Z_1

Next, we marginalize Z_1 , which is in the domain of potentials φ_{p1} and π_4 . Let π_5 denote $(\varphi_{p1} \otimes \pi_4)^{-Z_1}$. π_5 is computed as follows:

$$\begin{aligned}\pi_5(p, q_n) &= (\varphi_{p1} \otimes \pi_4)^{-Z_1}(p, q_n) \\ &= \int_{-\infty}^{\infty} \varphi_{p1}(z_1) \cdot (p(q_n + z_1) - f_{pC_n}(q_n + z_1)) dz_1 \\ &= p \cdot q_n - \int_{-\infty}^{\infty} f_{pC_n}(q_n + z_1) \cdot \varphi_{p1}(z_1) dz_1\end{aligned}\tag{5.6}$$

Notice that since φ_{p1} and f_{pC_n} are MOP functions, π_5 is also a MOP function (15 pieces, 5 degree). It takes Mathematica about 3.7 seconds to do the integration in Equation (5.6).

5.1.6 Marginalizing Q_n

Next, we marginalize Q_n . Q_n is in the domain of potentials θ_n and π_5 . Let π_6 denote $(\theta_n \otimes \pi_5)^{-Q_n}$. The details of π_6 are as follows.

$$\begin{aligned}\pi_6(p) &= (\theta_n \otimes \pi_5)^{-Q_n}(p) \\ &= \int_{-\infty}^{\infty} \delta(q_n - f_{pQ_n}(p)) \cdot \pi_5(p, q_n) dq_n \\ &= \pi_5(p, f_{pQ_n}(p))\end{aligned}\tag{5.7}$$

Since $f_{pQ_n}(p)$ is a piecewise-linear function, and π_5 is a MOP function, $\pi_6(p)$ is a MOP function. π_6 is computed as a 15-piece, 5-degree MOP function. It takes Mathematica about 14.8 seconds to do the integration in Equation (5.7).

5.1.7 Marginalizing P

Figure 5 shows a graph of $\pi_6(p)$ vs. p . Finally, we marginalize P . The maximum utility is 234.12 utiles at $p = \$25.76/\text{widget}$. It takes Mathematica 0.15 seconds to marginalize P from π_6 . For comparison, when demand and supply are known with certainty, the problem reduces to a nonlinear optimization problem and the maximum utility 198 utiles is obtained when price is $\$24.10/\text{widget}$.

5.2 An American Put Option Problem

This problem is adapted from Charnes and Shenoy [5]. An option trader has to decide whether or not to exercise a 7-month put option with initial stock price $S_0 = \$40$ and exercise price $X = \$35$. A put option on a stock provides the owner of the option the right to sell one share of the stock at the exercise price during the period of the option. For example, if the price of the stock dips to, say $\$30$, during the option period, then the owner of the option described above can buy one share at $\$30$, and sell it for $\$35$, with a realized profit of $\$5$. In reality, the option can be exercised at any time before the expiration of the option. For modeling purposes, we assume that the option is available for exercise at three

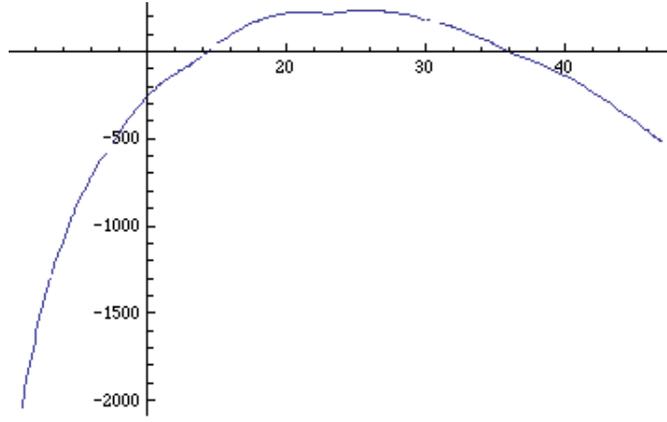


Figure 5: A Graph of $\pi_6(p)$ vs. p

equally-spaced decision points over a 7-month period. Following standard practice in the financial literature, the stock prices, S_1, S_2, \dots, S_k evolve according to the discrete stochastic process: $S_j = S_{j-1} \cdot Y$, where $Y \sim LN((r - \frac{\sigma^2}{2})\Delta t, \sigma^2\Delta t)$, for $j = 1, 2, \dots, k$, S_j is the stock price (in \$) at time $j\Delta t$, r is the risk-less interest rate (per year), σ is the stock's volatility (per year), T denotes the length of the option (in years), and $\Delta t = \frac{T}{k}$. We assume $r = 0.0488$ per year, $T = 0.5833$ years, $\Delta t = 0.1944$ years, $k = 3$ stages and $\sigma = 0.3$ per year (these constants are borrowed from Geske and Johnson [14], which provides an analytic value of the option for comparison purposes). Thus, $S_1 \sim LN(\ln 40 + 0.00074, 0.13229^2)$, $S_2|s_1 \sim LN(\ln s_1 + 0.00074, 0.13229^2)$, $S_3|s_2 \sim LN(\ln s_2 + 0.00074, 0.13229^2)$. An ID representation of the problem is shown in Figure 6.

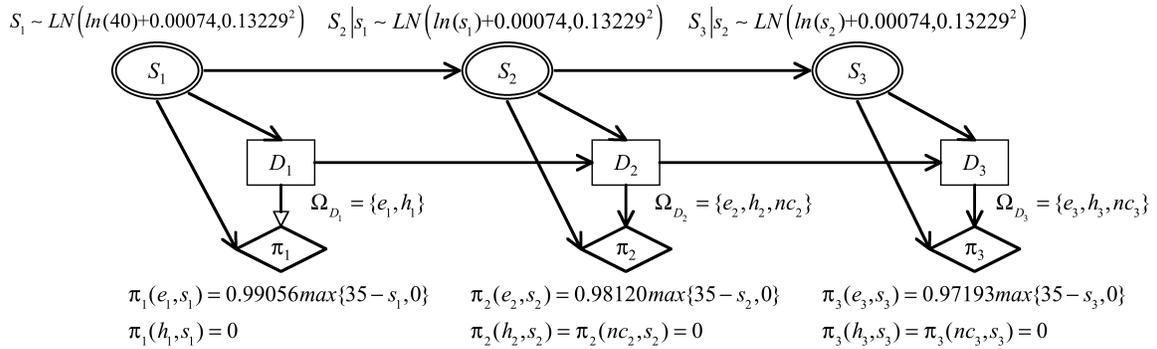


Figure 6: An ID Representation of the American Put Option

The state space of D_1 is $\{e_1, h_1\}$, i.e., exercise or hold. The constraints for the decision nodes D_2 and D_3 in the problem are shown in Figure 7, where e_i, h_i, nc_i denote the alternatives: exercise, hold, or no choice, respectively, for decision D_i , $i = 2, 3$. The only possible decision for stage i is no choice if the stock was exercised at a prior time. The additive factors of the utility function are: $\pi_j(d_j, s_j) = e^{-rj\Delta t} \max\{35 - s_j, 0\}$, if $d_j = e_j$; $\pi_j = 0$, otherwise. The $e^{-rj\Delta t}$ is a discount factor to translate future profits back to the present ($j = 0$). As in the entrepreneur's problem, we assume that the decision maker's utilities for profits are linear in \$.

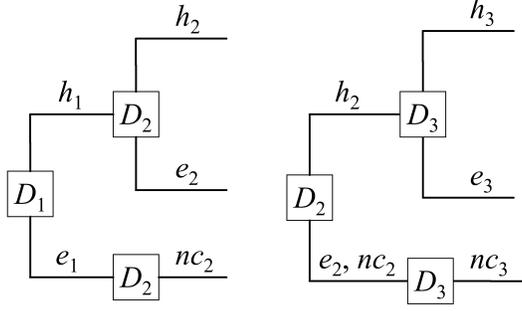


Figure 7: Constraints on D_2 and D_3

We approximate the marginal PDF of S_1 by a MOP function $\phi_{p1}(s_1)$. Also the MOP approximations of the conditional PDFs for $S_2|s_1$, and $S_3|s_2$ are denoted by $\psi_{p2}(s_1, s_2)$, and $\psi_{p3}(s_2, s_3)$, respectively. Also, we model the constraints on the choices at D_2 and D_3 by constraint potentials χ_2 for D_2 in $\{D_1, D_2\}$ and χ_3 for D_3 in $\{D_2, D_3\}$. The values of χ_2 and χ_3 are 1's for the possible states (as shown in Figure 7), and 0's for the rest.

The potentials in the problem are as follows (name, domain, and units):

π_3 ,	for $\{S_3, D_3\}$,	utils
χ_3 ,	for $D_3 \in \{D_2, D_3\}$,	no units
ψ_{p3} ,	for $\{S_2, S_3\}$,	$(\$)^{-1}$
χ_2 ,	for $D_2 \in \{D_1, D_2\}$,	no units
π_2 ,	for $\{S_2, D_2\}$,	utils
ψ_{p2} ,	for $\{S_1, S_2\}$,	$(\$)^{-1}$
π_1 ,	for $\{S_1, D_1\}$,	utils
ϕ_{p1} ,	for $\{S_1\}$,	$(\$)^{-1}$

The information constraints in the ID constrain us to marginalize the variables in the following sequence: $D_3, S_3, D_2, S_2, D_1, S_1$. Since the conditional arcs for S_1, S_2 and S_3 are consistent with the partial order determined by the information constraints, no divisions are required.

5.2.1 Marginalizing D_3 and S_3

First we marginalize D_3 . Since D_3 is in the domains of potentials π_3 and χ_3 , we first combine these and then marginalize D_3 from the combination. Let π'_3 denote $(\chi_3 \otimes \pi_3)^{-D_3}$. The units of values of π'_3 are utils. Because $\pi_3(e_3, s_3) \geq \pi_3(h_3, s_3)$ and $\pi_3(e_3, s_3) \geq \pi_3(nc_3, s_3)$ for all values of $s_3 < 35$, the details of π'_3 are as follows:

$$\pi'_3(d_2, s_3) = \begin{cases} 0.97 \max \{35 - s_3, 0\} & \text{if } d_2 = h_2, \\ 0 & \text{otherwise.} \end{cases} \quad (5.8)$$

Thus, the optimal strategy of Stage 3 would be to exercise the option if the observed value of $S_3 < 35$, assuming this alternative is available (i.e., the option has not been exercised earlier), and to abandon the option if the observed value of $S_3 \geq 35$.

Next, we marginalize S_3 , which is in the domains of ψ_{p3} and π'_3 . Let π''_3 denote $(\psi_{p3} \otimes \pi'_3)^{-S_3}$. The units of values of π''_3 are in utiles. Details of π''_3 are as follows.

$$\pi''_3(d_2, s_2) = \begin{cases} \int_0^{35} \pi'_3(d_2, s_3) \cdot \psi_{p3}(s_2, s_3) ds_3 & \text{if } d_2 = h_2, \\ 0 & \text{otherwise.} \end{cases} \quad (5.9)$$

Since π'_3 and ψ_{p3} are MOP functions, π''_3 is a MOP function (4 piece, 9 degree).

Similarly, we marginalize the remaining variables. The optimal decision function at stage 2 is to exercise the option if the observed stock price is less than \$24.75 (assuming this option is available); otherwise hold it for the next stage. The optimal decision function in stage 1 is to exercise the option when the stock price is less than \$28.15; otherwise hold it for the next stage. The optimal value of the option is computed as \$1.219.

Our result is comparable to the financial analytic result \$1.219 (using Black and Scholes [2] option pricing theory computed analytically in Geske and Johnson [14]), and the result \$1.224 computed by Monte Carlo method using 30 stages ([5]).

One practical benefit of solving this ID exactly is that not only do we get the value of the option (which is the focus of option pricing theory), we also get a strategy for exercising the option. The financial analytic result provides only the value of the option. The Monte Carlo method proposed in Charnes and Shenoy [5] provides an approximate strategy by providing bounds on when to exercise the option. Our technique provides an exact strategy for the ID in which the conditional PDFs are approximated by MOP functions.

6 Summary and Conclusions

The main contribution of this paper is a framework and an algorithm for solving hybrid IDs with discrete and continuous chance variables, discrete and continuous decision variables, and deterministic conditionals for continuous chance variables.

First, the extended Shenoy-Shafer architecture for making inferences in hybrid BNs proposed in Shenoy and West [39] has been further extended to include decision variables and utility functions. Second, we propose approximating conditional PDFs and utility functions by MOPs, and approximating nonlinear deterministic functions for continuous chance variables by piecewise linear functions. We have illustrated our framework and algorithm by solving two small hybrid IDs.

Two main problems in solving hybrid IDs are marginalization of continuous chance variables and marginalization of continuous decision variables. For decision problems that can be solved without divisions, one solution is to approximate conditional PDFs and utility functions by MOP functions, and nonlinear deterministic conditionals by piecewise linear functions. MOP functions are closed under multiplication, addition, integration, and under transformations needed for linear deterministic conditionals. However, they are not closed under divisions. Thus, MOP approximations could be used to mitigate the problems associated with marginalization of continuous chance and continuous decision variables when no divisions are needed. Also, it is relatively easier to maximize a utility function that is expressed in MOP form with a low degree. By solving for all real roots of a low-degree polynomial in closed form, we can compute a global maximum of the utility function as a function of other continuous variables in closed form.

There are two classes of decision problems that can be solved using local computation without doing any divisions. First, if we have a single utility function (with no additive factors), then combination is always multiplication, which is associative, and the axioms for local computations (see [38]) are satisfied by the combination and marginalization operations without needing any divisions. The entrepreneur’s problem is an example of this genre. Second, if we have a decision problem where the arcs pointing to chance variables are consistent with the partial order determined by information constraints, then again no divisions are necessary. The American put option is an example of this genre. Also, Markov reward processes, where the arcs always point forward in time (from state $S(t)$ to state $S(t + 1)$), are a class of problems where divisions are not required.

6.1 Limitations

For more general decision problems where divisions are needed to solve a problem using local computation, the method described in this paper will not work. The family of MOP functions is not closed under the division operation. Thus, if we divide a MOP function by another MOP function, the resulting function may not be a MOP, in which case there are no guarantees that we can integrate such functions in closed form. The *Pigs* problem, discussed in Lauritzen and Nilsson [26], is an example of a problem of this type (requiring divisions for solution using local computation). In general, if we have an additive factorization of the joint utility function, and arc reversals are necessary for solution, then such problems cannot be solved using local computation by using MOPs.

Our method based on MOPs inherits all the problems and issues that are inherent with the MOP method. First, we need to find MOP approximations of PDFs and utility functions. We can find MOP approximations by using Lagrange interpolating polynomials (LIP) with Chebyshev points, but it needs manual interventions regarding the location of knots that make up the pieces. Currently, we have some heuristics (mode, inflection points, equal width, etc.), but no theory for this decision. Except for this issue, we can automate the process of finding MOP approximations of conditional PDFs, and the process of solving hybrid IDs containing deterministic conditionals. Lagrange interpolating polynomials for one and higher dimensional functions can be easily found using commercial software (such as Mathematica, Maple, Matlab, etc.). The LIP method does not require that the function being approximated be differentiable. The theory of Chebyshev points exists for one and two-dimensional functions. Also, we can use a MOP approximation of the one-dimensional standard normal PDF to construct MOP approximations of higher dimensional CLG PDFs. However, constructing tractable MOP approximations of high dimensional non-CLG PDFs (such as a three-dimensional log-normal PDF) can be a challenge.

6.2 Future Work

How does the MOP method compare with the discretization and Monte Carlo methods? This is an important question that needs to be answered and for which we do not have answers. At this stage, we note that discretization has only been studied for one-dimensional PDFs. While this can be naïvely applied to multi-dimensional conditional PDFs, the quality of the resulting approximation has not been studied. For the conditional PDFs in the American

put option problem, one could, e.g., find a 3-point discrete approximation of the PDF of S_1 , a 3-point discrete approximation of the conditional PDF of $S_2 | s_1$ for each of the three values of S_1 , resulting in up to $3^2 = 9$ distinct values of S_2 , a 3-point discrete approximation of the PDF of $S_3 | s_2$ for each of the 9 distinct values of S_2 resulting in up to $3^3 = 27$ distinct values of S_3 , etc. Clearly, such a strategy is not tractable for many stages. Also, we cannot imagine obtaining a strategy as detailed as the one we obtain in stage 1 (exercise the option in stage 1 if the observed value of S_1 is less than 29.16, and hold otherwise) from a discretized model with only 3 possible values of S_1 . Finally, we note that Markov chain Monte Carlo methods wouldn't converge for a probability model that includes deterministic conditionals.

How close is the approximate solution found by using our method to the true answer? This is another important question for which we do not have answers. We note that the errors in the MOP approximation of conditional PDFs can be quantified using measures such as Kullback-Liebler (KL) divergence ([25]) and maximum absolute deviation between the MOP approximation and the target PDF ([37]). In terms of these measures, the approximations have very small errors. For example, the KL divergence between the standard normal PDF truncated to $(-3, 3)$ and the 2-piece, 3-degree MOP approximation described in Equation 4.2 is 0.009, and the maximum absolute deviation between the two functions is 0.014. However, we do not know how these errors influence the errors in the optimal strategy and the errors in the maximum expected utility. This is a topic that needs further research.

What is the size of decision problems that can be solved by our method? This is yet another important question for which we do not have answers. We plan to solve the American Put Option problem by gradually increasing the number of stages and observe where the method breaks down, if at all. The main problem here is computing a MOP approximation of the conditional for $S_j | s_{j-1}$. As we change the number of stages, we need to recompute all the MOP approximations of the conditional PDFs. This is another topic for further research.

Acknowledgments

We are grateful for comments from three anonymous reviewers from the *Uncertainty in Artificial Intelligence 2010* conference, and comments from two anonymous reviewers and an associate editor of *Decision Analysis*. A short version of this paper appears in the proceedings of *Uncertainty in Artificial Intelligence 2010* conference as Li and Shenoy [27].

APPENDIX: Properties of Dirac Delta Functions

Two basic properties of Dirac delta functions are as follows (see e.g., [12], [13], [16], [20]).

1. (*Sampling*) If $f(x)$ is any function that is continuous in the neighborhood of a , then

$$\int_{-\infty}^{\infty} f(x) \delta(x - a) dx = f(a). \quad (6.1)$$

2. (*Rescaling*) If $g(x)$ has real (non-complex) zeros at a_1, \dots, a_n , and is differentiable at

these points, and $g'(a_i) \neq 0$ for $i = 1, \dots, n$, then

$$\delta(g(x)) = \sum_{i=1}^n \frac{\delta(x - a_i)}{|g'(a_i)|}.$$

A more extensive list of properties of the Dirac delta function that are relevant for uncertain reasoning can be found in Cinicioglu and Shenoy [6].

References

- [1] C. Bielza, P. Müller, and D. Ríos. Decision analysis by augmented probability simulation. *Management Science*, 45(7):995–1007, 1999.
- [2] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):1323–1352, 1973.
- [3] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks Cole, Boston, MA, 9th edition, 2010.
- [4] A. Cano, M. Gómez, and S. Moral. A forward-backward Monte Carlo method for solving influence diagrams. *International Journal of Approximate Reasoning*, 42(1–2):119–135, 2006.
- [5] J. M. Charnes and P. P. Shenoy. Multi-stage Monte Carlo method for solving influence diagrams using local computation. *Management Science*, 50(3):405–418, 2004.
- [6] E. N. Cinicioglu and P. P. Shenoy. Arc reversals in hybrid Bayesian networks with deterministic variables. *International Journal of Approximate Reasoning*, 50(5):763–777, 2009.
- [7] B. R. Cobb. Influence diagrams with continuous decision variables and non-Gaussian uncertainties. *Decision Analysis*, 4(3):136–155, 2007.
- [8] B. R. Cobb and P. P. Shenoy. Hybrid Bayesian networks with linear deterministic variables. In F. Bacchus and T. Jaakkola, editors, *Uncertainty in Artificial Intelligence: Proceedings of the 21st Conference*, pages 136–144. AUAI Press, Corvallis, OR, 2005.
- [9] B. R. Cobb and P. P. Shenoy. Nonlinear deterministic relationships in Bayesian networks. In L. Godo, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 8th European Conference, ECSQARU 2005*, Lecture Notes in Artificial Intelligence 3571, pages 27–38. Springer, Berlin, 2005.
- [10] B. R. Cobb and P. P. Shenoy. Decision making with hybrid influence diagrams using mixtures of truncated exponentials. *European Journal of Operational Research*, 186(1):261–275, 2008.

- [11] B. R. Cobb, P. P. Shenoy, and R. Rumí. Approximating probability density functions in hybrid Bayesian networks with mixtures of truncated exponentials. *Statistics & Computing*, 16(3):293–308, 2006.
- [12] P. A. M. Dirac. The physical interpretation of the quantum dynamics. *Proceedings of the Royal Society of London, Series A*, 113(765):621–641, 1927.
- [13] P. A. M. Dirac. *The Principles of Quantum Mechanics*. Oxford U.P., London, UK, 4th edition, 1958.
- [14] R. Geske and H. E. Johnson. The Americal put option valued analytically. *Journal of Finance*, 39(5):1511–1524, 1984.
- [15] D. B. Hertz. Risk analysis in capital investment. *Harvard Business Review*, 42(1):95–106, 1964.
- [16] R. F. Hoskins. *Generalised Functions*. Ellis Horwood, Chichester, UK, 1979.
- [17] R. A. Howard. Proximal decision analysis. *Management Science*, 17(9):845–879, 1971.
- [18] R. A. Howard and J. E. Matheson. Influence diagrams. In R. A. Howard and J. E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, volume II, pages 719–762. Strategic Decisions Group, Menlo Park, CA, 1984.
- [19] R. A. Howard and J. E. Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, 2005.
- [20] R. P. Kanwal. *Generalized Functions: Theory and Technique*. Birkhäuser, Boston, MA, 2nd edition, 1998.
- [21] D. L. Keefer and S. E. Bodily. Three-point approximations for continuous random variables. *Management Science*, 29(5):595–609, 1983.
- [22] T. W. Keelin and B. W. Powley. Quantile-parameterized distributions. *Decision Analysis*, 8(3):206–219, 2011.
- [23] A. V. Kozlov and D. Koller. Nonuniform dynamic discretization in hybrid networks. In D. Geiger and P. P. Shenoy, editors, *Uncertainty in Artificial Intelligence: Proceedings of the 13th Conference*, pages 302–313. Morgan Kaufmann, San Francisco, CA, 1997.
- [24] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.
- [25] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951.
- [26] S. L. Lauritzen and D. Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251, 2001.

- [27] Y. Li and P. P. Shenoy. Solving hybrid influence diagrams with deterministic variables. In P. Grünwald and P. Spirtes, editors, *Uncertainty in Artificial Intelligence: Proceedings of the 26th Conference*, pages 322–331. AUA Press, Corvallis, OR, 2010.
- [28] A. C. Miller III and T. R. Rice. Discrete approximations of probability distributions. *Management Science*, 29(3):352–362, 1983.
- [29] S. Moral, R. Rumí, and A. Salmerón. Mixtures of truncated exponentials in hybrid Bayesian networks. In S. Benferhat and P. Besnard, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 6th European Conference, ECSQARU-2001*, Lecture Notes in Artificial Intelligence 2143, pages 156–167. Springer, Berlin, 2001.
- [30] S. M. Olmsted. *On representing and solving decision problems*. PhD thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA., 1983.
- [31] L. E. Ortiz and L. P. Kaelbling. Sampling methods for action selection in influence diagrams. In H. Kautz and B. Porter, editors, *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 378–385. AAAI Press, Menlo Park, CA, 2000.
- [32] W. B. Poland III. *Decision analysis with continuous and discrete variables: A mixture distribution approach*. PhD thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA., 1994.
- [33] W. B. Poland III and R. D. Shachter. Mixtures of Gaussians and minimum relative entropy techniques for modeling continuous uncertainties. In D. Heckerman and A. Mamdani, editors, *Uncertainty in Artificial Intelligence: Proceedings of the 9th Conference*, pages 183–190. Morgan Kaufmann, San Francisco, CA, 1993.
- [34] R. D. Shachter and C. R. Kenley. Gaussian influence diagrams. *Management Science*, 35(5):527–550, 1989.
- [35] P. Shenoy, R. Rumí, and A. Salmerón. Some practical issues in inference in hybrid Bayesian networks with deterministic conditionals. In *IEEE Proceedings of the 2011 Eleventh International Conference on Intelligent Systems Design and Applications (ISDA-11)*, pages 605–610. IEEE Research Publishing Services, Piscataway, NJ, 2011.
- [36] P. P. Shenoy. Valuation-based systems for Bayesian decision analysis. *Operations Research*, 40(3):463–484, 1992.
- [37] P. P. Shenoy. Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. *International Journal of Approximate Reasoning*, in press, 2012.
- [38] P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In R. D. Shachter, T. Levitt, J. F. Lemmer, and L. N. Kanal, editors, *Uncertainty in Artificial Intelligence 4*, pages 169–198. North-Holland, Amsterdam, The Netherlands, 1990.

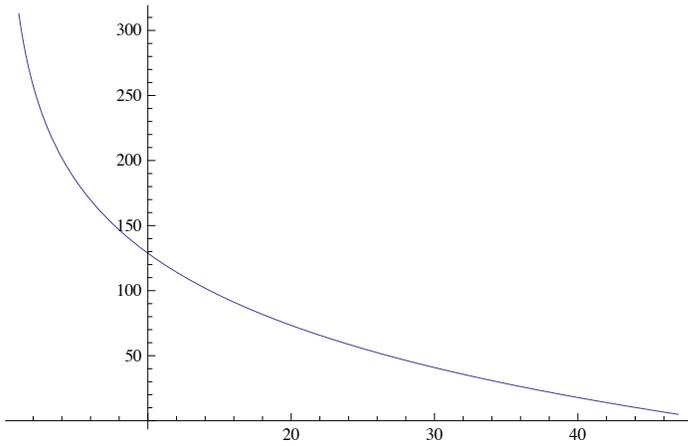
- [39] P. P. Shenoy and J. C. West. Extended Shenoy-Shafer architecture for inference in hybrid Bayesian networks with deterministic conditionals. *International Journal of Approximate Reasoning*, 52(6):805–818, 2011.
- [40] P. P. Shenoy and J. C. West. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657, 2011.
- [41] J. E. Smith. Moment methods for decision analysis. *Management Science*, 39(3):340–358, 1993.
- [42] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):365–379, 1990.
- [43] Y. Xu. Lagrange interpolation on Chebyshev points of two variables. *Journal of Approximation Theory*, 87(2):220–238, November 1996.

Appendix I: Details of the Solution of the Entrepreneur's Problem Using Mathematica®

```

(* This notebook has a solution of the Entrepreneur's Problem *)
(* Computing limits of variables P, Qn, Qa, Cn, Ca *)
Clear[pmin, pmax, qnmin, qnmax, qamin, qamax, cnmin, cnmax, camin, camax];
pmin = 1;
pmax = 47;
Plot[80 (Log[50] - Log[p]), {p, pmin, pmax}]
qnmax = N[80 (Log[50] - Log[pmin])]
qnmin = N[80 (Log[50] - Log[pmax])]
qamin = N[qnmin - 3]
qamax = N[qnmax + 3]
Plot[700 + 4 qa + 400 (1 - Exp[-qa / 50]), {qa, qamin, qamax}]
cnmax = N[700 + 4 qamax + 400 (1 - Exp[-qamax / 50])]
cnmin = N[700 + 4 qamin + 400 (1 - Exp[-qamin / 50])]
camin = N[cnmin - 3]
camax = N[cnmax + 3]

```

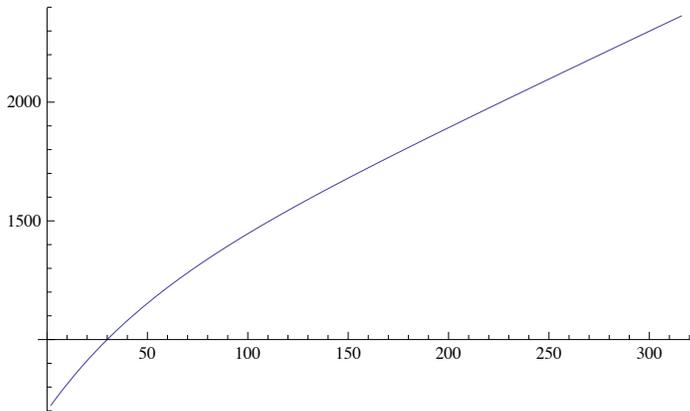


312.962

4.95003

1.95003

315.962



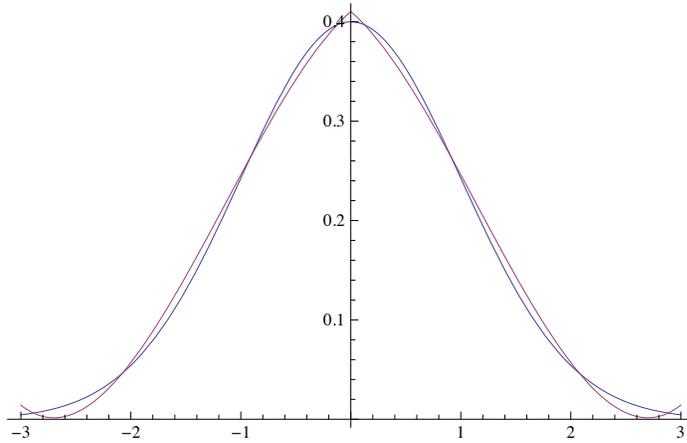
2363.13

723.1

720.1

2366.13

```
(* 2p3d MOP approximation of N(0, 1) from Shenoy[2010]*)
Clear[fZu, kZ, fZ];
fZu[z_] = Piecewise[{{PDF[NormalDistribution[0, 1.], z], -3 ≤ z ≤ 3}}];
kZ = ∫-∞∞ fZu[z] dz;
fZ[z_] = Simplify[fZu[z] / kZ];
Clear[a0, a1, a2, a3, fpZ];
a0 = 0.4103496175477352`;
a1 = 0.09499365944761645`;
a2 = -0.0978625597129868`;
a3 = -0.028502995174713566`;
fpZ[z_] = Piecewise[{{
  {a0 + a1 z + a2 z2 + a3 z3, -3 ≤ z < 0},
  {a0 - a1 z + a2 z2 - a3 z3, 0 ≤ z ≤ 3}
}}];
Plot[{fZ[z], fpZ[z]}, {z, -3, 3}]
Clear[a0, a1, a2, a3, fZu, kZ, fZ];
{
  0.41035 + 0.0949937 z - 0.0978626 z2 - 0.028503 z3  -3 ≤ z < 0
  0.41035 - 0.0949937 z - 0.0978626 z2 + 0.028503 z3  0 ≤ z ≤ 3
  0 True
}
```

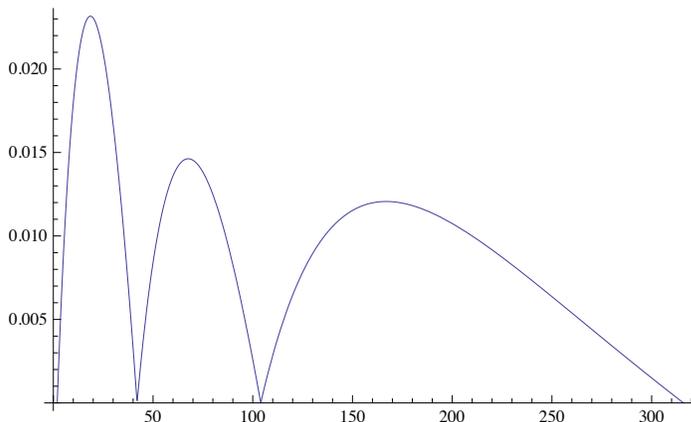
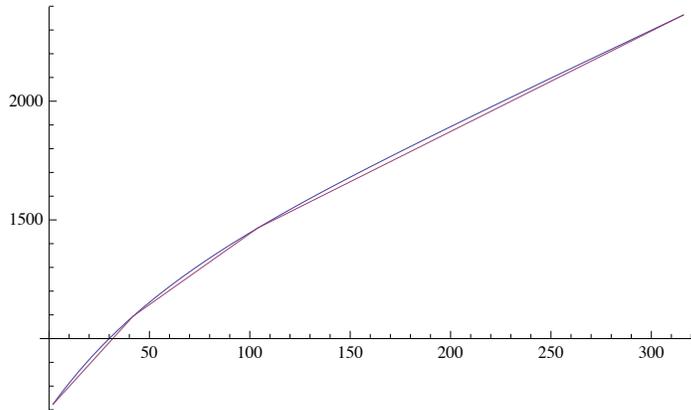


```
(* Define Potentials *)
Clear[p1, p2, gcn, gpcn, p3, p4, p5,
  qn, ptsq1, ptsq2, ptsq3, ptsq4, ptsq5, ptsq6, fqn, p6];
p1[cn_, z2_, ca_] = DiracDelta[ca - cn - z2]
p2[z2_] = fpZ[z2]
gcn[qa_] = Piecewise[{{1100 + 4. qa - 400. e-qa/50, qamin ≤ qa ≤ qamax}}]
Clear[a3, b3, x31, x32, x33, x34, ptsq1];
a3 = qamin;
b3 = 42.;
ptsqa1 = {{a3, gcn[a3]}, {b3, gcn[b3]}};
a4 = 42.;
b4 = 104.;
ptsqa2 = {{a4, gcn[a4]}, {b4, gcn[b4]}};
a5 = 104.;
b5 = qamax;
ptsqa3 = {{a5, gcn[a5]}, {b5, gcn[b5]}};
gpcn[qa_] = Simplify[Piecewise[{
  {InterpolatingPolynomial[ptsqa1, qa], qamin ≤ qa < 42},
  {InterpolatingPolynomial[ptsqa2, qa], 42 ≤ qa < 104},
  {InterpolatingPolynomial[ptsqa3, qa], 104 ≤ qa ≤ qamax}
}]]
Plot[{gcn[qa], gpcn[qa]}, {qa, qamin, qamax}]
Plot[(gcn[qa] - gpcn[qa]) / gcn[qa], {qa, qamin, qamax}]
FindMaximum[{Abs[(gcn[qa] - gpcn[qa]) / gcn[qa]], qamin < qa < qamax}, {qa, 25}]
p3[qa_, cn_] = DiracDelta[cn - gpcn[qa]]
p4[qn_, z1_, qa_] = DiracDelta[qa - qn - z1]
p5[z1_] = fpZ[z1]
qn[p_] = 80 (Log[50] - Log[p])
c1 = 1.;
c2 = 3.;
c3 = 7.;
c4 = 13.;
c5 = 23.;
c6 = 36.;
c7 = 47.;
ptsq1 = {{c1, qn[c1]}, {c2, qn[c2]}};
ptsq2 = {{c2, qn[c2]}, {c3, qn[c3]}};
ptsq3 = {{c3, qn[c3]}, {c4, qn[c4]}};
ptsq4 = {{c4, qn[c4]}, {c5, qn[c5]}};
ptsq5 = {{c5, qn[c5]}, {c6, qn[c6]}};
ptsq6 = {{c6, qn[c6]}, {c7, qn[c7]}};
fqn1[p_] = InterpolatingPolynomial[ptsq1, p];
fqn2[p_] = InterpolatingPolynomial[ptsq2, p];
fqn3[p_] = InterpolatingPolynomial[ptsq3, p];
fqn4[p_] = InterpolatingPolynomial[ptsq4, p];
fqn5[p_] = InterpolatingPolynomial[ptsq5, p];
fqn6[p_] = InterpolatingPolynomial[ptsq6, p];
fqn[p_] = Piecewise[{
  {fqn1[p], c1 ≤ p < c2},
  {fqn2[p], c2 ≤ p < c3},
  {fqn3[p], c3 ≤ p < c4},
  {fqn4[p], c4 ≤ p < c5},
  {fqn5[p], c5 ≤ p < c6},
  {fqn6[p], c6 ≤ p ≤ c7}
}]
Plot[{qn[p], fqn[p]}, {p, 1, 47}]
Plot[(fqn[p] - qn[p]) / qn[p], {p, 1, 47}, PlotRange → All]
FindMaximum[{Abs[(fqn[p] - qn[p]) / qn[p]], 1 < p < 47}, {p, 43}]
p6[p_, qn_] = DiracDelta[qn - fqn[p]]
11.5073 / qn[2]
DiracDelta[ca - cn - z2]
```

$$\begin{cases} 0.41035 + 0.0949937 z_2 - 0.0978626 z_2^2 - 0.028503 z_2^3 & -3 \leq z_2 < 0 \\ 0.41035 - 0.0949937 z_2 - 0.0978626 z_2^2 + 0.028503 z_2^3 & 0 \leq z_2 \leq 3 \\ 0 & \text{True} \end{cases}$$

$$\begin{cases} 1100 - 400. e^{-qa/50} + 4. qa & 1.95003 \leq qa \leq 315.962 \\ 0 & \text{True} \end{cases}$$

$$\begin{cases} 1025.86 + 4.23236 qa & 104. \leq qa \leq 315.962 \\ 844.188 + 5.97923 qa & 42 \leq qa < 104. \\ 704.977 + 9.29378 qa & 1.95003 \leq qa < 42 \\ 0 & \text{True} \end{cases}$$



{0.0231646, {qa → 18.606}}

$$\text{DiracDelta}\left[\text{cn} - \begin{cases} 1025.86 + 4.23236 qa & 104. \leq qa \leq 315.962 \\ 844.188 + 5.97923 qa & 42 \leq qa < 104. \\ 704.977 + 9.29378 qa & 1.95003 \leq qa < 42 \\ 0 & \text{True} \end{cases} \right]$$

DiracDelta[qa - qn - z1]

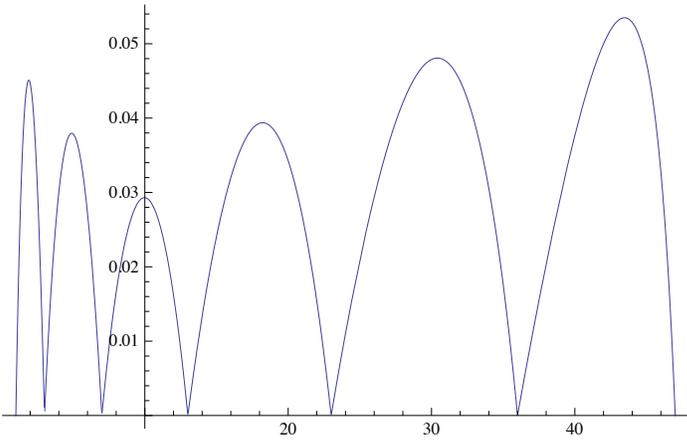
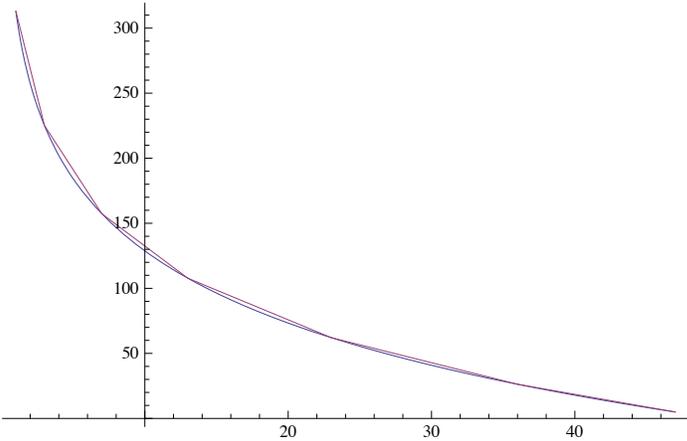
$$\begin{cases} 0.41035 + 0.0949937 z_1 - 0.0978626 z_1^2 - 0.028503 z_1^3 & -3 \leq z_1 < 0 \\ 0.41035 - 0.0949937 z_1 - 0.0978626 z_1^2 + 0.028503 z_1^3 & 0 \leq z_1 \leq 3 \\ 0 & \text{True} \end{cases}$$

80 (Log[50] - Log[p])

```

{ 225.073- 43.9445 (-3. + p)  1. ≤ p < 3.
 157.289- 16.946 (-7. + p)   3. ≤ p < 7.
 107.766- 8.25386 (-13. + p) 7. ≤ p < 13.
 62.1223- 4.56436 (-23. + p) 13. ≤ p < 23.
 26.2803- 2.75708 (-36. + p) 23. ≤ p < 36.
 4.95003- 1.93912 (-47. + p) 36. ≤ p ≤ 47.
 0                               True

```



```
{0.0534971, {p → 43.4628}}
```

```

DiracDelta [qn - { 225.073- 43.9445 (-3. + p)  1. ≤ p < 3.
                  157.289- 16.946 (-7. + p)   3. ≤ p < 7.
                  107.766- 8.25386 (-13. + p) 7. ≤ p < 13.
                  62.1223- 4.56436 (-23. + p) 13. ≤ p < 23.
                  26.2803- 2.75708 (-36. + p) 23. ≤ p < 36.
                  4.95003- 1.93912 (-47. + p) 36. ≤ p ≤ 47.
                  0                               True

```

```
0.0446868
```

```

(* Step 1: Marginalize Ca *)
Clear[pi, pil];
pi[p_, qa_, ca_] = p qa - ca;
Timing[pil[p_, qa_, cn_, z2_] =
  Assuming[cn ∈ Reals && z2 ∈ Reals, ∫-∞∞ pi[p, qa, ca] p1[cn, z2, ca] dca]]
{0.050161, -cn + p qa - z2}

```

```

(* Step 2: Marginalize Z2 *)
Clear[pi2];
Timing[pi2[p_, qa_, cn_] = Simplify[ $\int_{-3}^3$  pi1[p, qa, cn, z2] p2[z2] dz2]]
{0.402676, -1. cn + 1. p qa}

(* Step 3: Marginalize Cn *)
Clear[pi2a, pi3];
pi2a[p_, qa_, cn_] = p qa - cn;
Timing[pi3[p_, qa_] = Assuming[p ∈ Reals && qa ∈ Reals,  $\int_{-\infty}^{\infty}$  pi2a[p, qa, cn] p3[qa, cn] dcn]]
{0.109065, p qa -  $\begin{cases} 1025.86 + 4.23236 \text{ qa} & 104. \leq \text{qa} \leq 315.962 \\ 844.188 + 5.97923 \text{ qa} & 42 \leq \text{qa} < 104. \\ 704.977 + 9.29378 \text{ qa} & 1.95003 \leq \text{qa} < 42 \\ 0 & \text{True} \end{cases}$ }

(* Step 4: Marginalize Qa *)
Clear[pi4];
Timing[pi4[p_, qn_, z1_] =
  Assuming[qn ∈ Reals && p ∈ Reals && z1 ∈ Reals,  $\int_{-\infty}^{\infty}$  pi3[p, qa] p4[qn, z1, qa] dqa]]
{0.178617, p (qn + z1) -  $\begin{cases} 1025.86 + 4.23236 (qn + z1) & 104. \leq qn + z1 \leq 315.962 \\ 844.188 + 5.97923 (qn + z1) & 42 \leq qn + z1 < 104. \\ 704.977 + 9.29378 (qn + z1) & 1.95003 \leq qn + z1 < 42 \\ 0 & \text{True} \end{cases}$ }

(* Step 5: Marginalize Z1 *)

```

Clear[pi5]

Timing[pi5[p_, qn_] = Simplify[PiecewiseExpand[p qn - \int_{-3}^3 fpz[z1] gpcn[qn + z1] dz1]]]

{3.67883,

$0. + p qn$	$qn \geq 318.962 \mid \mid qn \leq -1.049!$
$-704.977 + (-9.29378 + p) qn$	$4.95003 \leq qn \leq 39.$
$-844.188 + (-5.97923 + p) qn$	$45. \leq qn \leq 101.$
$-1025.86 + (-4.23236 + p) qn$	$107. \leq qn \leq 312.962$
$-511.259 + (-2.11618 + p) qn$	$qn = 315.962$
$96.7698 + (-7.48594 + p) qn - 221.265 qn^2 + 62.8316 qn^3 - 4.9477 qn^4 - 0.013245 qn^5$	$1.95003 < qn < 4.95003$
$-1.46734 \times 10^{11} + (1.80526 \times 10^9 + p) qn - 8.1314 \times 10^6 qn^2 + 15 227. qn^3 - 7.34455 qn^4 - 0.00603175 qn^5$	$312.962 < qn < 315.962$
$529 778. + (-65 270.6 + p) qn + 3207.7 qn^2 - 78.7331 qn^3 + 0.964954 qn^4 - 0.00472374 qn^5$	$39. < qn < 42.$
$2.85944 \times 10^7 + (-1.39129 \times 10^6 + p) qn + 27 071.2 qn^2 - 263.315 qn^3 + 1.28032 qn^4 - 0.00248955 qn^5$	$101. < qn < 104.$
$-3.19217 \times 10^7 + (1.51933 \times 10^6 + p) qn - 28 919.6 qn^2 + 275.168 qn^3 - 1.30881 qn^4 + 0.00248955 qn^5$	$104. < qn < 107.$
$-697 149. + (81 162.4 + p) qn - 3778.53 qn^2 + 87.8154 qn^3 - 1.01902 qn^4 + 0.00472374 qn^5$	$42. < qn < 45.$
$1.50909 \times 10^{11} + (-1.84273 \times 10^9 + p) qn + 8.23619 \times 10^6 qn^2 - 15 293.9 qn^3 + 7.27552 qn^4 + 0.00603175 qn^5$	$315.962 < qn < 318.962$
$1.09833 \times 10^9 + (-4.20673 \times 10^7 + p) qn + 597 360. qn^2 - 3659.97 qn^3 + 7.34455 qn^4 + 0.00603175 qn^5$	$qn = 104.$
$2.06087 \times 10^7 + (-2.00228 \times 10^6 + p) qn + 72 267.1 qn^2 - 1133.47 qn^3 + 6.06424 qn^4 + 0.0085213 qn^5$	$qn = 42.$
$-13.9834 + (-46.6967 + p) qn - 55.0757 qn^2 - 16.8377 qn^3 + 5.09928 qn^4 + 0.013245 qn^5$	$True$

(* Step 6: Marginalize Qn *)

Clear[pi6];

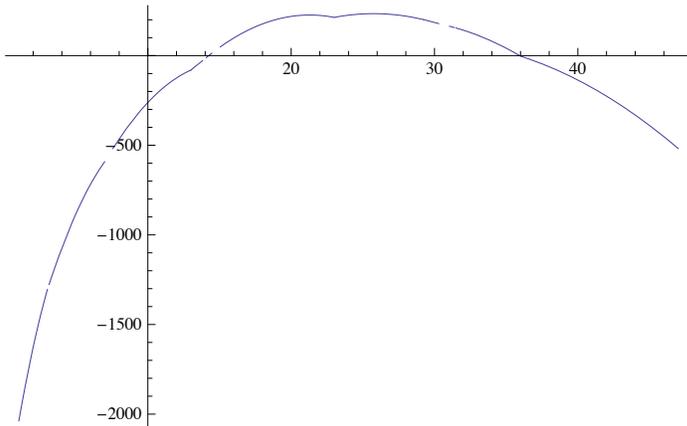
Timing[Assuming [p ∈ Reals, pi6[p_] = $\int_{-\infty}^{\infty}$ pi5[p, qn] p6[p, qn] dqn];]

{14.7687, Null}

(* Determining # pieces and degree of pi6 *)
pi6a[p_] = Simplify[PiecewiseExpand[pi6[p]]]

-13.9834	$p < 1. p > 47.$
-2536.42 + 542.895 p - 43.9445 p ²	$1. \leq p < 3.$
-2193.62 + 347.632 p - 16.946 p ²	$3. \leq p < 7.$
-1936.1 + 249.999 p - 8.25386 p ²	$7. \leq p < 13.$
-4.56436 (-26.531 + p) (-14.3117 + p)	$13. \leq p \leq 13.1678$
-4.56436 (-28.3384 + p) (-14.2511 + p)	$14.4823 \leq p < 23.$
-2.75708 (-34.9705 + p) (-16.5407 + p)	$23. \leq p \leq 29.2103$
-2.75708 (-35.9337 + p) (-18.8921 + p)	$31.3865 \leq p < 36.$
-1.93912 (-35.8762 + p) (-22.9703 + p)	$36. \leq p \leq 47.$
-11.9493 (-393.523 + p) (-13.8349 + p)	$p = 13.8251$
(-13.1172 + p) (215.502 - 29.3499 p + p ²)	
-4.93196 (-15.4318 + p) (-14.2468 + p)	$13.1678 < p < 13.8251$
(-11.3905 + p) (182.796 - 26.8024 p + p ²)	
-1.35753 (-362.93 + p) (-30.1568 + p)	$p = 30.2984$
(-29.2002 + p) (1008.29 - 63.4933 p + p ²)	
-0.752539 (-32.8989 + p) (970.482 - 62.0948 p + p ²) (743.778 - 54.4229 p + p ²)	$29.2103 < p < 30.2984$
0.752539 (-27.3438 + p) (1105.41 - 66.4375 p + p ²) (901.117 - 59.7863 p + p ²)	$30.2984 < p < 31.3865$
4.93196 (-14.2509 + p) (241.11 - 30.9596 p + p ²) (159.927 - 25.1685 p + p ²)	True

Plot[pi6a[p], {p, pmin, pmax}]



(* Step 7: Marginalize P *)

Timing[FindMaximum[pi6a[p], {p, 20}]]

{0.148605, {234.117, {p -> 25.7556}}}

Appendix II: Details of the Solution of the American Put Option Problem Using
Mathematica®

```

(* This notebook has a solution of the 3-stage Americal Put Option Problem *)
(* r = annual risk-free rate *)
(* X = strike price of option *)
(* σ is volatility *)
(* T is period of option in years *)
(* t is time between exercise points in years *)
(* m0 = mean of log-normal *)
(* n0 = std dev of log normal *)
Clear[r, X, σ, T, t, m0, n0];
r = 0.0488;
X = 35;
σ = 0.3;
T = 7 / 12;
t = T / 3;
m0 =  $\left( r - \frac{\sigma^2}{2} \right) t$ ;
n0 = σ (t0.5);

```

```

(* Truncated Exact PDF of S1 *)
Clear[μ1, σ1, ω1, m1, ms1, vs1, g1, a, b, c, ftslu, kfts1, mts1, vts1, s1]
μ1 = Log[40] + m0;
σ1 = n0;
m1 = eμ1;
ω1 = eσ12;
ms1 = m1 e0.5 σ12
vs1 = m12 ω1 (ω1 - 1)
g1[s1_] = CDF[LogNormalDistribution[μ1, σ1], s1];
a = s1 /. FindRoot[g1[s1] - 0.0015, {s1, 30}]
c = s1 /. FindRoot[g1[s1] - 0.9985, {s1, 50}]
b = m1 / ω1
ftslu[s1_] = Simplify[Piecewise[{{PDF[LogNormalDistribution[μ1, σ1], s1], a ≤ s1 ≤ c}}]];
kfts1 = ∫-∞∞ ftslu[z] dz
fts1[s1_] = Simplify[ftslu[s1] / kfts1]
∫-∞∞ fts1[z] dz
mts1 = ∫-∞∞ z fts1[z] dz
vts1 = ∫-∞∞ z2 fts1[z] dz - mts12
40.3814
28.7876
27.0321
59.2765
39.3351
0.997
{ 0.
  { 3.64285 × 10-169 e-28.5714 Log[s1]2 s1209.835 True
    s1 > 59.2765 || s1 < 27.0321
  }
}

```

```

1.
40.371
27.879
(* LIP approx. of PDF of S1 *)
Clear[n, a1, b1, pts1];
n = 6;
a1 = a;
b1 = b;
pts1 = Table[{{ $\frac{a1+b1}{2} + \left(\frac{b1-a1}{2}\right) \cos\left[\frac{2i-1}{2n}\pi\right]$ ,  $\text{fts1}\left[\frac{a1+b1}{2} + \left(\frac{b1-a1}{2}\right) \cos\left[\frac{2i-1}{2n}\pi\right]\right]$ }}, {i, n}];
Clear[a1, b1, pts2];
a1 = b;
b1 = c;
pts2 = Table[{{ $\frac{a1+b1}{2} + \left(\frac{b1-a1}{2}\right) \cos\left[\frac{2i-1}{2n}\pi\right]$ ,  $\text{fts1}\left[\frac{a1+b1}{2} + \left(\frac{b1-a1}{2}\right) \cos\left[\frac{2i-1}{2n}\pi\right]\right]$ }}, {i, n}];
Clear[f11, f12, fs1u, kfs1, fs1, mS1, vS1, KLS1, MADS1, AEMS1, AEVS1];
f11[z_] = Expand[InterpolatingPolynomial[pts1, z]];
f12[z_] = Expand[InterpolatingPolynomial[pts2, z]];
MinValue[{f11[z], a ≤ z ≤ b}, z]
MinValue[{f12[z], b ≤ z ≤ c}, z]
fs1u[z_] = Piecewise[{{
    {f11[z], a ≤ z < b},
    {f12[z], b ≤ z ≤ c}
}]];
kfs1 =  $\int_{-\infty}^{\infty}$  fs1u[z] d[z];
fs1[z_] = Simplify[fs1u[z] / kfs1]
Plot[{{fts1[z], fs1[z]}, {z, a, c}]
 $\int_{-\infty}^{\infty}$  fs1[z] d[z]
mS1 =  $\int_{-\infty}^{\infty}$  z fs1[z] d[z]

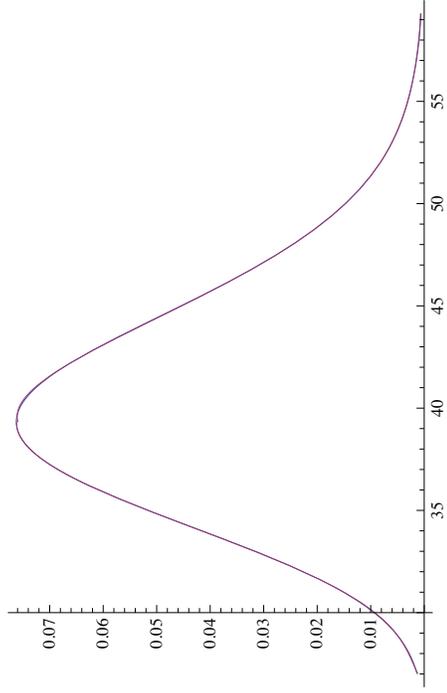
```

```

vs1 =  $\int_{-\infty}^{\infty} z^2 \text{fs1}[z] dz - \text{ms1}^2$ 
KLs1 = NIntegrate[Log[fts1[x] / fs1[x]] fts1[x], {x, a, c}]
(* MADS1 = MaxValue[{Abs[fts1[x]-fs1[x]], a ≤ x ≤ c}, x]*)
AEMS1 = Abs[mts1 - ms1]
AEVS1 = Abs[vtS1 - vS1]

0.00123792
0.000687413
0.9999903
 $\left( \begin{array}{l} -31.1694 + 4.75471 z - 0.286338 z^2 + 0.00849391 z^3 - 0.000123918 z^4 + 7.11102 \times 10^{-7} z^5 \\ -49.5655 + 4.8479 z - 0.187256 z^2 + 0.00358184 z^3 - 0.0000340028 z^4 + 1.28359 \times 10^{-7} z^5 \\ 0 \end{array} \right)$ 

```



- 1.
- 40.3701
- 27.875
- 8.65051×10^{-6}
- 0.000958009

```

0.00398066
(* Truncated PDF of S2*)
Clear[μ2, σ2, m2, ω2, ms2, vs2, g2, a, b, c, fts2u, kfts2, fts2, mts2, vts2, x]
μ2 = Log[40] + 2 m0;
σ2 =  $\sqrt{2}$  n0;
m2 =  $e^{\mu^2}$ ;
ω2 =  $e^{\sigma^2}$ ;
ms2 = m2  $e^{0.5 \sigma^2}$ 
vs2 = m22 ω2 (ω2 - 1)
g2[x_] = CDF[LogNormalDistribution[μ2, σ2], x]
a = x /. FindRoot[g2[x] - 0.0015, {x, 30}]
c = x /. FindRoot[g2[x] - 0.9985, {x, 50}]
b = m2 / ω2
fts2u[x_] = Simplify[Piecewise[{{PDF[LogNormalDistribution[μ2, σ2], x], a ≤ x ≤ c}}]];
kfts2 =  $\int_{-\infty}^{\infty}$  fts2u[x] dx;
fts2[x_] = Simplify[fts2u[x] / kfts2]
 $\int_{-\infty}^{\infty}$  fts2[x] dx
mts2 =  $\int_{-\infty}^{\infty}$  x fts2[x] dx
vts2 =  $\int_{-\infty}^{\infty}$  x2 fts2[x] dx - mts22
40.7664
59.1963
 $\left\{ \begin{array}{l} \frac{1}{2} \text{Erfc}[3.77964 (3.69036 - \text{Log}[x])] \quad x > 0 \\ 0 \end{array} \right.$  True
22.992
69.7955
38.6813

```

```

{ 0.
  { 6.86629 × 10-85 e-14.2857 Log[x]2 x104.439 True
    x > 69.7955 || x < 22.992
  }
}

1.
40.7453
57.1764

(* LIP approx. of PDF of S2 *)
Clear[n, a1, b1, pts1];
n = 6;
a1 = a;
b1 = b;
pts1 = Table[{{ $\frac{a1 + b1}{2} + \left(\frac{b1 - a1}{2}\right) \cos\left[\frac{2 i - 1}{2 n} \pi\right]$ , fts2[ $\frac{a1 + b1}{2} + \left(\frac{b1 - a1}{2}\right) \cos\left[\frac{2 i - 1}{2 n} \pi\right]$ }}, {i, n}];
Clear[a1, b1, pts2];
a1 = b;
b1 = c;
pts2 = Table[{{ $\frac{a1 + b1}{2} + \left(\frac{b1 - a1}{2}\right) \cos\left[\frac{2 i - 1}{2 n} \pi\right]$ , fts2[ $\frac{a1 + b1}{2} + \left(\frac{b1 - a1}{2}\right) \cos\left[\frac{2 i - 1}{2 n} \pi\right]$ }}, {i, n}];
Clear[f21, f22, kf2, f2u, fs2, ms2, vs2];
f21[x_] = Expand[InterpolatingPolynomial[pts1, x]];
f22[x_] = Expand[InterpolatingPolynomial[pts2, x]];
MinValue[{f21[x], a ≤ x ≤ b}, x]
MinValue[{f22[x], b ≤ x ≤ c}, x]
f2u[x_] = Piecewise[{{
  {f21[x], a ≤ x < b},
  {f22[x], b ≤ x ≤ c}
}]];
Clear[kf2, fs2];
kf2 =  $\int_{-\infty}^{\infty} f2u[x] dx$ 
fs2[x_] = Simplify[f2u[x] / kf2]
Plot[{fts2[x], fs2[x]}, {x, a, c}]

```

$$\int_{-\infty}^{\infty} fs2[x] dx$$

$$mS2 = \int_{-\infty}^{\infty} x fs2[x] dx$$

$$vS2 = \int_{-\infty}^{\infty} x^2 fs2[x] dx - mS2^2$$

KLS2 = NIntegrate[Log[fts2[x] / fs2[x]] fts2[x], {x, a, c}]

(* MADs2 = MaxValue[{Abs[fts2[x]-fs2[x]],as x ≤ c},x]*)

AEMS2 = Abs[mtS2 - mS2]

AEVS2 = Abs[vtS2 - vS2]

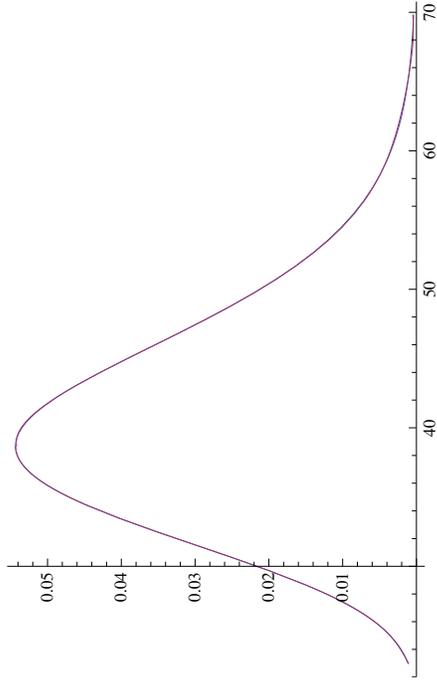
0.00106894

0.000408304

1.00004

0.999958

```
{
  { -4.4058 + 0.760957 x - 0.0515467 x^2 + 0.00170447 x^3 - 0.0000274073 x^4 + 1.7161 × 10^-7 x^5    22.992 ≤ x < 38.6813
  { -8.26243 + 0.761648 x - 0.027266 x^2 + 0.000478374 x^3 - 4.13649 × 10^-6 x^4 + 1.41524 × 10^-8 x^5    38.6813 ≤ x ≤ 69.7955
  { 0
```



1.

40.7428

```

57.132
0.0000449632
0.00252274
0.0444055
(*Truncated PDF of s3*)
(* Clear[μ3, σ3, m3, ω3, ms3, vs3, g3, a, b, c, fts3u, k, fts3, mts3, vts3]
μ3=Log[40]+3m0;
σ3=√3 n0;
m3=eμ3;
ω3=eσ32;
ms3=m3 e0.5σ32
vs3=m32 ω3 (ω3-1)
g3[x_]=CDF[LogNormalDistribution[μ3,σ3],x]
a=x/.FindRoot[g3[x]-0.005,{x,30}]
c=x/.FindRoot[g3[x]-0.995,{x,50}]
b=m3/ω3
fts3u[x_]=Simplify[Piecewise[{{PDF[LogNormalDistribution[μ3,σ3],x], a ≤ x ≤ c}}]];
kfts3 = ∫-∞∞ fts3u[x] dx
fts3[x_] = Simplify[fts3u[x]/kfts3]
∫-∞∞ fts3[x] dx
mts3 = ∫-∞∞ x fts3[x] dx
vts3 = ∫-∞∞ x2 fts3[x] dx - mts32 *)

```

```

(* LIP approx. of PDF of S3 *)
(* Clear[n, a1,b1,pts1];
n=5;
a1= a;
b1 = b;
pts1 = Table[{{(a1+b1)/2 + ((b1-a1)/2) Cos[2 i -1]π}, fts3[(a1+b1)/2 + ((b1-a1)/2) Cos[2 i -1]π]}],{i, n}];
Clear[a1, b1,pts2];
a1= b;
b1=c;
pts2 = Table[{{(a1+b1)/2 + ((b1-a1)/2) Cos[2 i -1]π}, fts3[(a1+b1)/2 + ((b1-a1)/2) Cos[2 i -1]π]}],{i, n}];
Clear[f31,f32, fs3u,kfs3,fs3,mS3,vS3]
f31[z_]=Expand[InterpolatingPolynomial[pts1,z]];
f32[z_]=Expand[InterpolatingPolynomial[pts2,z]];
MinValue[{f31[z],as z ≤ b},z]
MinValue[{f32[z],bs z ≤ c},z]
fs3u[z_] = Piecewise[{{
{f31[z],as z<b},
{f32[z],bs z ≤ c}
}]];
kfs3 = ∫-∞∞fs3u[z]dz
fs3[z_] = Simplify[fs3u[z]/kfs3]
Plot[{fts3[z], fs3[z]},{z,a,c]}
∫-∞∞fs3[x]dx
mS3 = ∫-∞∞x fs3[x]dx
vS3 = ∫-∞∞x2 fs3[x]dx - mS32
KLS3 = NIntegrate[Log[fts3[x]/fs3[x]] fts3[x],{x, a, c}]
(* MADS3 = MaxValue[{Abs[fts3[x]-fs3[x]],as x ≤ c},x]*
AEMS3 = Abs[mts3-mS3]
AEVS3 = Abs[vtS3-vS3] *)

```

```

(* LIP approximation of s2|s1*)
Clear[μ1, σ1, m1, ω1, g1, l1, u1, u2]
μ1 = Log[40] + m0;
σ1 = n0;
m1 = eμ1;
ω1 = eσ12;
g1[x_] = CDF[LogNormalDistribution[μ1, σ1], x];
l1 = x /. FindRoot[g1[x] - 0.0015, {x, 30}]
u2 = x /. FindRoot[g1[x] - 0.9985, {x, 50}]
u1 = m1 / ω1
27.0321
59.2765
39.3351

(* If S1 = l1, then S2|s1 ~ LN(Log[l1] + m0, n02) *)
Clear[μl1, gl1, ll1, ull1];
μl1 = Log[l1] + m0;
gl1[x_] = CDF[LogNormalDistribution[μl1, n0], x];
ll1 = x /. FindRoot[gl1[x] - 0.0015, {x, 20}]
ull1 = x /. FindRoot[gl1[x] - 0.9985, {x, 50}]
18.2683
40.0592

(* If S1 = u1, then S2|s1 ~ LN(Log[u1] + m0, n02) *)
Clear[μu1, gu1, lu1, uu1];
μu1 = Log[u1] + m0;
gu1[x_] = CDF[LogNormalDistribution[μu1, n0], x];
lu1 = x /. FindRoot[gu1[x] - 0.0015, {x, 20}]
uu1 = x /. FindRoot[gu1[x] - 0.9985, {x, 50}]
26.5827
58.2913

```

```
(* If S1 = u1, then S2|s1 ~ LN(Log[u2] + m0, n0^2) *)
Clear[μu2, gu2, lu2, uu2];
μu2 = Log[u2] + m0;
gu2[x_] = CDF[LogNormalDistribution[μu2, n0], x];
lu2 = x /. FindRoot[gu2[x] - 0.005, {x, 20}]
uu2 = x /. FindRoot[gu2[x] - 0.995, {x, 50}]
42.1908
83.4045
```

```

(* Piece # 1 *)
Clear[y, z, k, n, a, b, c, d, e];
a = 11
b = u1
c = 111
d = lu1

e[x_] = c +  $\frac{d-c}{b-a} (x-a)$ 

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[g, n, m, pts1, pts2, pts11, h1];
g[x1_, x2_] = PDF[LogNormalDistribution[Log[x1] + m0, n0], x2];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h1[x1_, x2_] = Piecewise[{{
Expand[InterpolatingPolynomial[pts11, {x1, x2}]], (a ≤ x1 < b) && (e[x1] ≤ x2 < d)}
}]
MinValue[{h1[x1, x2], (a ≤ x1 < b) && (e[x1] ≤ x2 < d)}, {x1, x2}]
ArgMin[{h1[x1, x2], (a ≤ x1 < b) && (e[x1] ≤ x2 < d)}, {x1, x2}]
Plot3D[{{Style[g[x1, x2], Red], Style[h1[x1, x2], Blue]}}, {x1, a, b}, {x2, c, d}]
(* Plot3D[g[x1, x2] - h1[x1, x2], {x1, a, b}, {x2, c, d}] *)
MaxValue[{Abs[g[x1, x2] - h1[x1, x2]], (a ≤ x1 < b) && (c ≤ x2 < d)}, {x1, x2}]

```

27.0321

39.3351

18.2683

```

26.5827
18.2683 + 0.675801 (-27.0321 + x)
[ 0.0100728 + 0.903551 x1 - 0.083381 x12 + 0.00342674 x13 -
  0.00006881 x14 + 5.24239 × 10-7 x15 - 1.33815 x2 +
  0.104008 x1 x2 - 0.00485269 x12 x2 + 0.000116256 x13 x2 -
  9.60186 × 10-7 x14 x2 + 0.0287354 x22 +
  0.000115866 x1 x22 - 0.0000320372 x12 x22 +
  2.72849 × 10-7 x13 x22 - 0.000650971 x23 +
  0.0000149885 x1 x23 + 1.23768 × 10-6 x24
] 0

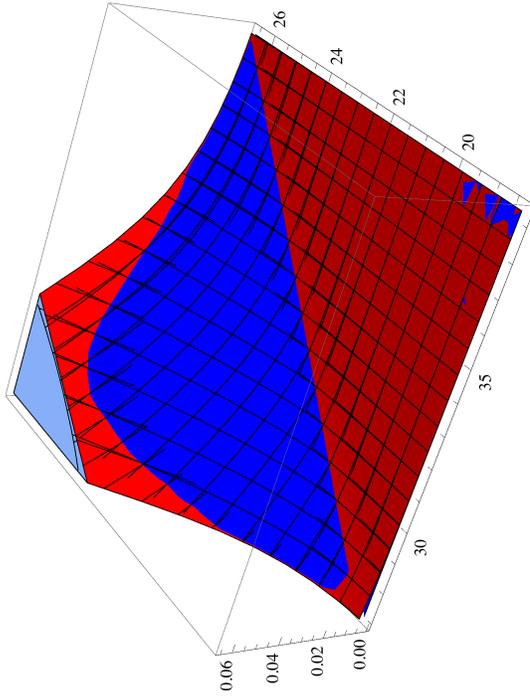
```

True

```

0.
{39.3351, 26.5827}

```



0.00648863

```

(* Piece # 2 *)
Clear[a, b, c, d, k, n, Y, z]
a = 11
b = u1
c = lu1
d = (u1+lu1) / 2
Y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h2];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{Y[2 i, n], z[2 j, n]}, g[Y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{Y[2 i+1, n], z[2 j+1, n]}, g[Y[2 i+1, n], z[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h2[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]], (a ≤ s1 < b) && (c ≤ s2 < d)}
}}]
MinValue[{h2[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
ArgMin[{h2[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h2[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h2[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h2[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

```

27.0321

39.3351

26.5827

33.321

$27.0321 \leq s1 < 39.3351 \ \&\& \ 26.5827 \leq s2 < 33.321$

```

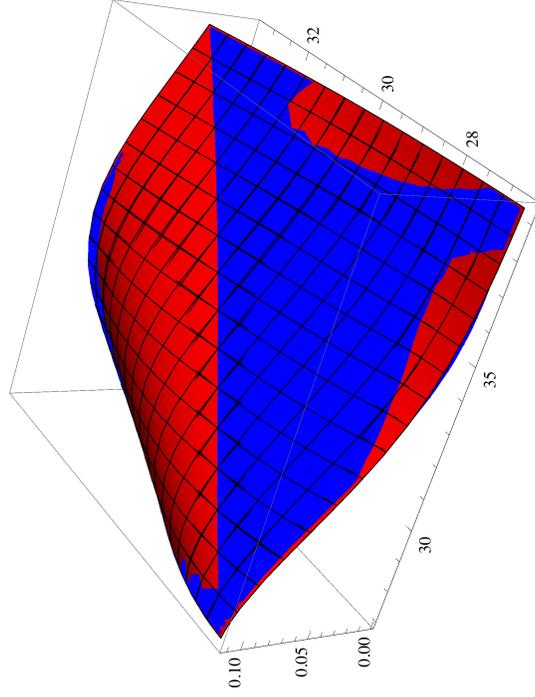
51.4975 - 3.72149 s1 - 0.174002 s1^2 + 0.00903354 s1^3 -
0.0000618109 s1^4 - 7.12729 x 10^-7 s1^5 - 5.65401 s2 +
0.845966 s1 s2 - 0.0125011 s1^2 s2 - 0.000304691 s1^3 s2 +
5.99302 x 10^-6 s1^4 s2 + 0.00711299 s2^2 - 0.0270791 s1 s2^2 +
0.000875708 s1^2 s2^2 - 8.35213 x 10^-6 s1^3 s2^2 +
0.00615153 s2^3 - 0.0000334451 s1 s2^3 - 0.0000411804 s2^4
0

```

True

0.001747

{36.8966, 26.5827}



0.0046018

```

(* Piece # 3 *)
Clear[a, b, c, d, k, n, Y, z]
a = 11
b = u1
c = (u11+1u1) / 2
d = u11

Y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h3];
n = 5;
m = (n + 1) / 2;
pts1 = Flatten[Table[{{Y[2 i, n], z[2 j, n]}, g[Y[2 i, n], z[2 j, n]]}, {i, 0, m - 1}, {j, 0, m - 1}], 1];
pts2 =
  Flatten[Table[{{Y[2 i + 1, n], z[2 j + 1, n]}, g[Y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m - 1}, {j, 0, m - 1}], 1];
pts11 = Join[pts1, pts2];
h3[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (c ≤ s2 < d)}
]
MinValue[{h3[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
ArgMin[{h3[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h3[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h3[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h3[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

27.0321
39.3351
33.321
40.0592

```

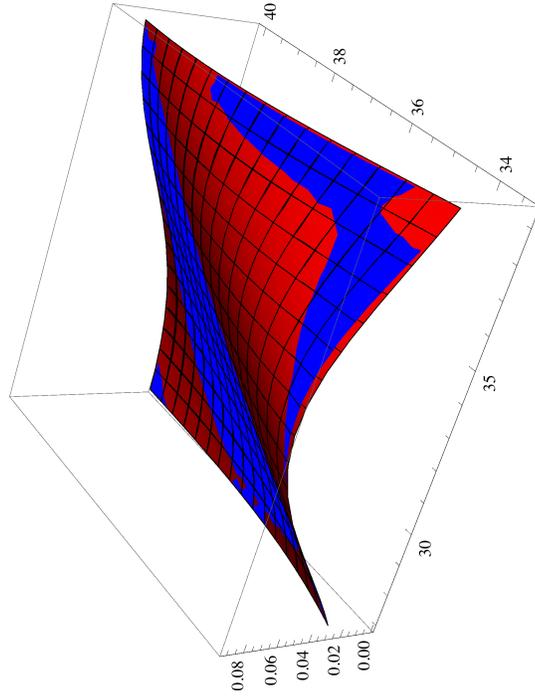
$27.0321 \leq s1 < 39.3351$ && $33.321 \leq s2 < 40.0592$

$$\left[\begin{array}{l} -65.8969 + 6.03949 s1 + 0.0313806 s1^2 - 0.00606532 s1^3 + \\ 0.0000539802 s1^4 + 5.36013 \times 10^{-7} s1^5 + 4.62905 s2 - \\ 0.695146 s1 s2 + 0.0135465 s1^2 s2 + 0.000118921 s1^3 s2 - \\ 3.74773 \times 10^{-6} s1^4 s2 - 0.00371257 s2^2 + 0.0158148 s1 s2^2 - \\ 0.000502397 s1^2 s2^2 + 5.0184 \times 10^{-6} s1^3 s2^2 - \\ 0.0029105 s2^3 + 5.39134 \times 10^{-6} s1 s2^3 + 0.0000184492 s2^4 \\ 0 \end{array} \right]$$

True

0.00133551

{27.0321, 39.4314}



0.0770818

```

(* Piece # 4 *)
Clear[a, b, c, d, e, y, z, k, n]
a = 11
b = u1
c = u11
d = uu1
e[x_] = c +  $\frac{d-c}{b-a}$  (x-a)
y[k_, n_] = 0.5 (a + b + (b-a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d-c) Cos[ $\frac{k\pi}{n}$ ]);
Clear[n, m, pts1, pts2, pts11, h4];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]], {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{y[2 i+1, n], z[2 j+1, n]}, g[y[2 i+1, n], z[2 j+1, n]]], {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h4[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (c ≤ s2 ≤ e[s1])}
}]
MinValue[{h4[s1, s2], (a ≤ s1 < b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
ArgMin[{h4[s1, s2], (a ≤ s1 < b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h4[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h4[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h4[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

27.0321
39.3351
40.0592

```

```

58.2913
40.0592 + 1.48191 (-27.0321 + x)
┌ 0.00459354 + 0.135755 s1 - 0.0359396 s12 +
  0.00192429 s13 - 0.0000387404 s14 +
  2.81319 × 10-7 s15 - 0.0918454 s2 + 0.031561 s1 s2 -
  0.0014331 s12 s2 + 0.0000240759 s13 s2 -
  1.57177 × 10-7 s14 s2 - 0.00492555 s22 -
  4.21302 × 10-6 s1 s22 + 2.76421 × 10-6 s12 s22 -
  2.20469 × 10-8 s13 s22 + 0.0000640246 s23 -
  5.28575 × 10-7 s1 s23 - 2.66288 × 10-7 s24
└ 0

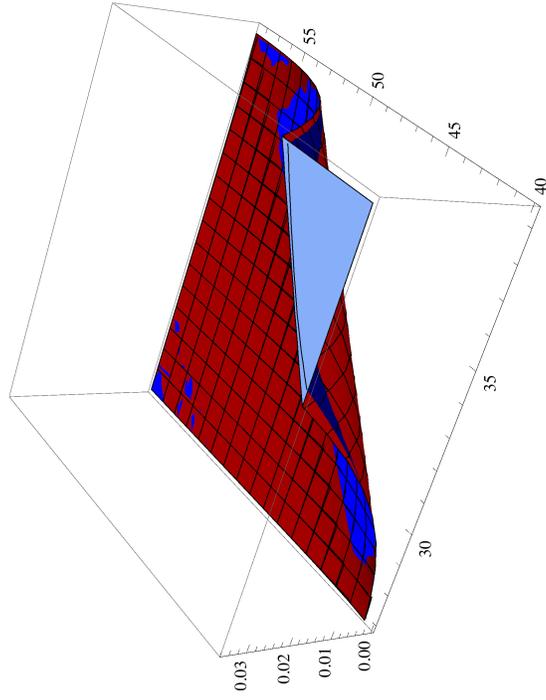
```

True

```

0.
{27.0321, 40.0592}

```



```

0.0746258

```

```

(* Piece # 5 *)
Clear[a, b, c, d, y, z, k, n]
a = u1
b = u2
c = lu1
d = lu2
e[x_] = c +  $\frac{d-c}{b-a}$  (x-a);
y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);
Clear[n, m, pts1, pts2, pts11, h5];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h5[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]}], (a ≤ s1 ≤ b) && (e[s1] ≤ s2 < d)}
}]
MinValue[{h5[s1, s2], (a ≤ s1 ≤ b) && (e[s1] ≤ s2 < d)}, {s1, s2}]
ArgMin[{h5[s1, s2], (a ≤ s1 ≤ b) && (e[s1] ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h5[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h5[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h5[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

39.3351
59.2765
26.5827

```

```

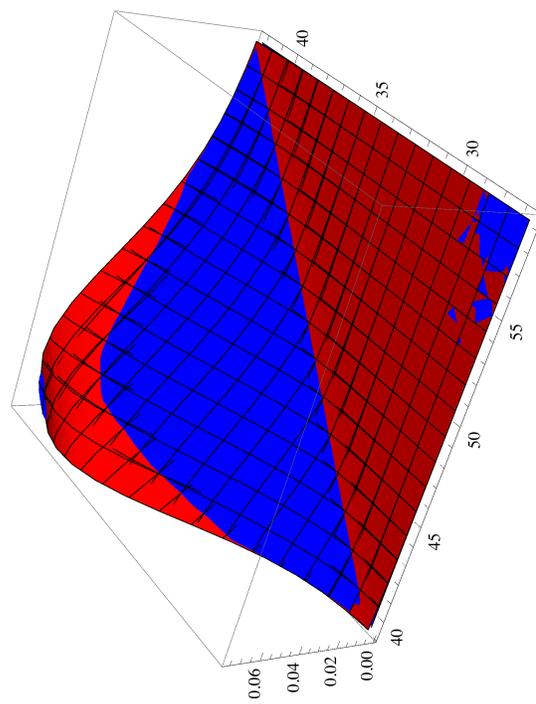
42.1908
-2.092 + 0.525227 s1 - 0.0293338 s1^2 +
0.000825722 s1^3 - 0.0000115601 s1^4 +
5.92933 × 10-8 s15 - 0.484226 s2 + 0.0306829 s1 s2 -
0.00121995 s12 s2 + 0.0000219424 s13 s2 -
1.19691 × 10-7 s14 s2 + 0.00277949 s22 +
0.000388344 s1 s22 - 0.0000121509 s12 s22 +
5.61031 × 10-8 s13 s22 - 0.000149294 s23 +
3.91694 × 10-6 s1 s23 - 4.72229 × 10-7 s24
0

```

```

0.
{59.2765, 42.1908}
True

```



```

0.00886195

```

```

(* Piece # 6 *)
Clear[a, b, c, d, y, z, k, n];
a = u1
b = u2
c = 1u2
d = (uu1+1u2) / 2

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h6];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n]], g[y[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i+1, n], z[2 j+1, n]}, g[y[2 i+1, n]], g[y[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h6[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]}], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}
}]
MinValue[{h6[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}, {s1, s2}]
ArgMin[{h6[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h6[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h6[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h6[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

39.3351
59.2765
42.1908
50.241

```

39.3351 ≤ s1 ≤ 59.2765 && 42.1908 ≤ s2 < 50.241

```

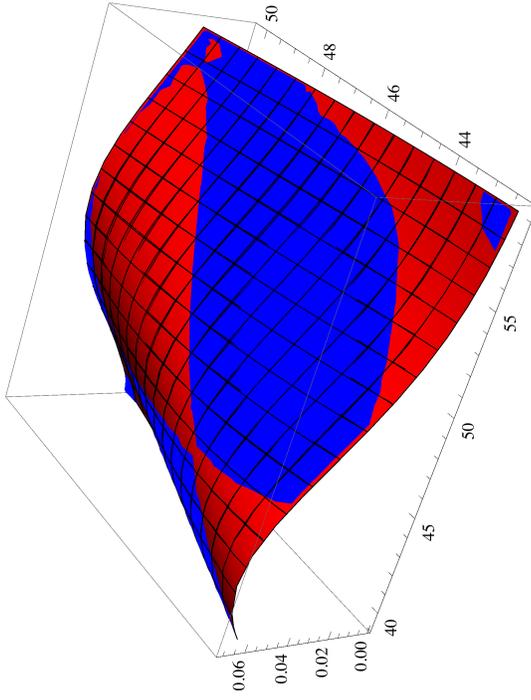
{
85.0521 - 3.1744 s1 + 0.00392855 s12 + 0.000778935 s13 +
2.31225 × 10-6 s14 - 1.0553 × 10-7 s15 - 5.80302 s2 +
0.266597 s1 s2 - 0.0029304 s12 s2 - 0.0000394179 s13 s2 +
5.27798 × 10-7 s14 s2 + 0.106664 s22 - 0.00536586 s1 s22 +
0.000122468 s12 s22 - 7.57452 × 10-7 s13 s22 -
0.000355207 s23 - 8.08149 × 10-6 s1 s23 + 4.27373 × 10-6 s24
}
0

```

True

0.00358208

{59.2765, 42.9874}



0.00118593

```

(* Piece # 7 *)
Clear[a, b, c, d, y, z, k, n]
a = u1
b = u2
c = (uu1+lu2) / 2
d = uu1

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h7];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n]], g[y[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i+1, n], z[2 j+1, n]}, g[y[2 i+1, n]], g[y[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h7[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}
}}]
MinValue[{h7[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}, {s1, s2}]
ArgMin[{h7[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h7[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h7[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h7[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]

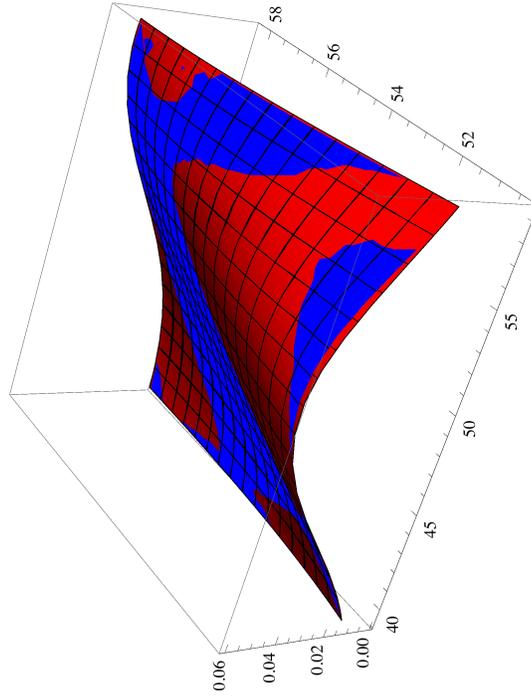
39.3351
59.2765
50.241
58.2913

```

$39.3351 \leq s1 \leq 59.2765$ && $50.241 \leq s2 < 58.2913$

$$\begin{aligned} & -30.3157 + 4.35442 s1 - 0.0289735 s1^2 - 0.000805247 s1^3 + \\ & 4.44482 \times 10^{-6} s1^4 + 6.50434 \times 10^{-8} s1^5 + 0.109743 s2 - \\ & 0.263079 s1 s2 + 0.00373838 s1^2 s2 + 0.0000112195 s1^3 s2 - \\ & 3.63596 \times 10^{-7} s1^4 s2 + 0.0679827 s2^2 + 0.0038108 s1 s2^2 - \\ & 0.0000804011 s1^2 s2^2 + 5.43321 \times 10^{-7} s1^3 s2^2 - \\ & 0.00155237 s2^3 + 4.67509 \times 10^{-7} s1 s2^3 + 6.97476 \times 10^{-6} s2^4 \\ & \left. \begin{aligned} & 0 \\ & 0.000811763 \\ & \{39.3351, 57.8881\} \end{aligned} \right\} \end{aligned}$$

True



0.000632805

```

(* Piece # 8 *)
Clear[a, b, c, d, e, y, z, k, n]
a = u1
b = u2
c = uu1
d = uu2

e[x_] = c +  $\frac{d-c}{b-a} (x-a)$ ;

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h8];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h8[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]}], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}
}}]
MinValue[{h8[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
ArgMin[{h8[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h8[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h8[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h8[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
39.3351
59.2765
58.2913
83.4045

```

```

0.0878255 - 0.045921 s1 - 0.006222224 s1^2 +
0.000274939 s1^3 - 4.00757 × 10^-6 s1^4 +
1.91609 × 10^-8 s1^5 + 0.0108817 s2 + 0.0094821 s1 s2 -
0.000259956 s1^2 s2 + 3.13348 × 10^-6 s1^3 s2 -
1.16335 × 10^-8 s1^4 s2 - 0.00258998 s2^2 -
0.0000187232 s1 s2^2 + 1.68111 × 10^-7 s1^2 s2^2 -
2.84739 × 10^-9 s1^3 s2^2 + 0.0000269859 s2^3 +
1.29726 × 10^-7 s1 s2^3 - 1.18479 × 10^-7 s2^4
0

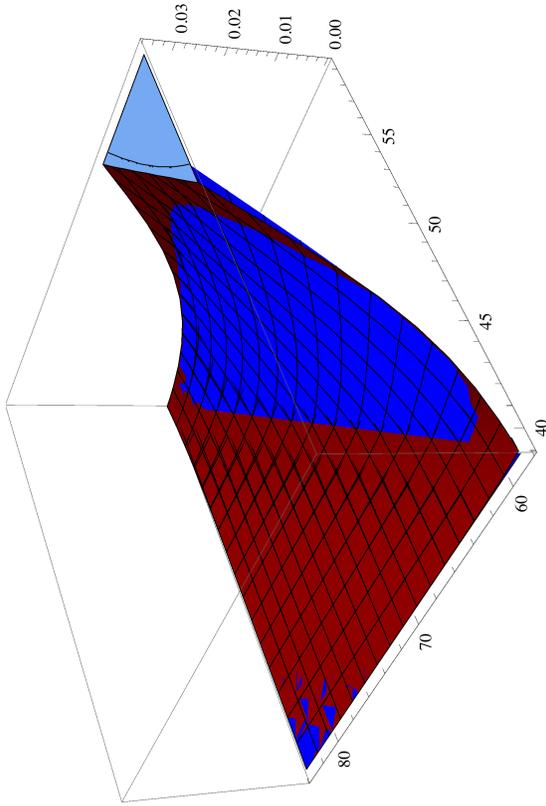
```

```

0.
{39.3351, 58.2913}

```

True



```

0.00443363

```

```

(* Putting all 8 pieces together *)
Clear[glip, h, cg, x1, x2];
h[x1_, x2_] = PiecewiseExpand[
  h1[x1, x2] + h2[x1, x2] + h3[x1, x2] + h4[x1, x2] + h5[x1, x2] + h6[x1, x2] + h7[x1, x2] + h8[x1, x2]];
Timing[N[cg =  $\int_{l11}^{\text{uu2}} \int_{l11}^{\text{uu2}} h[x1, x2] fs1[x1] dx1 dx2$ ]]
glip[x1_, x2_] =  $\frac{1}{cg} h[x1, x2]$ 
Plot3D[Style[g[x1, x2], Red], Style[glip[x1, x2], Blue]], {x1, l1, uu2}, {x2, l11, uu2}, PlotRange -> All]

{98.2105, 1.01817}
(
51.4975 - 3.72149 x1 - 0.174002 x12 + 0.00903354 x13 -
0.0000618109 x14 - 7.12729 x10-7 x15 - 5.65401 x2 +
0.845966 x1 x2 - 0.0125011 x12 x2 - 0.000304691 x13 x2 +
5.99302 x10-6 x14 x2 + 0.00711299 x22 - 0.0270791 x1 x22 +
0.000875708 x12 x22 - 8.35213 x10-6 x13 x22 +
0.00615153 x23 - 0.0000334451 x1 x23 - 0.0000411804 x24
- 2.092 + 0.525227 x1 - 0.0293338 x12 +
0.000825722 x13 - 0.0000115601 x14 +
5.92933 x10-8 x15 - 0.484226 x2 + 0.0306829 x1 x2 -
0.00121995 x12 x2 + 0.0000219424 x13 x2 -
1.19691 x10-7 x14 x2 + 0.00277949 x22 +
0.000388344 x1 x22 - 0.0000121509 x12 x22 +
5.61031 x10-8 x13 x22 - 0.000149294 x23 +
3.91694 x10-6 x1 x23 - 4.72229 x10-7 x24
0.00459354 + 0.135755 x1 - 0.0359396 x12 +
0.00192429 x13 - 0.0000387404 x14 +
2.81319 x10-7 x15 - 0.0918454 x2 + 0.031561 x1 x2 -
0.0014331 x12 x2 + 0.0000240759 x13 x2 -
1.57177 x10-7 x14 x2 - 0.00492555 x22 -
- 1.48191 x1 + x2  $\leq$  1.20792 x10-13
27.0321  $\leq$  x1 < 39.3351 && x2  $\geq$  40.0592 &&
- 1.48191 x1 + x2  $\leq$  1.20792 x10-13
39.3351  $\leq$  x1  $\leq$  59.2765 &&
0.782696 x1 - x2  $\leq$  4.20473 && x2 < 42.1908
27.0321  $\leq$  x1 < 39.3351 && 26.5827  $\leq$  x2 < 33.321
)

```

```

0.982156
4.21302 × 10-9 x1 x24 + 2.76421 × 10-9 x14 x24 -
2.20469 × 10-8 x13 x22 + 0.0000640246 x23 -
5.28575 × 10-7 x1 x23 - 2.66288 × 10-7 x24
0.0878255 - 0.045921 x1 - 0.006222224 x12 +
0.000274939 x13 - 4.00757 × 10-6 x14 +
1.91609 × 10-8 x15 + 0.0108817 x2 + 0.0094821 x1 x2 -
0.000259956 x12 x2 + 3.13348 × 10-6 x13 x2 -
1.16335 × 10-8 x14 x2 - 0.00258998 x22 -
0.0000187232 x1 x22 + 1.68111 × 10-7 x12 x22 -
2.84739 × 10-9 x13 x22 + 0.0000269859 x23 +
1.29726 × 10-7 x1 x23 - 1.18479 × 10-7 x24
0.0100728 + 0.903551 x1 - 0.083381 x12 + 0.00342674 x13 -
0.00006881 x14 + 5.24239 × 10-7 x15 - 1.33815 x2 +
0.104008 x1 x2 - 0.00485269 x12 x2 + 0.000116256 x13 x2 -
9.60186 × 10-7 x14 x2 + 0.0287354 x22 +
0.000115866 x1 x22 - 0.0000320372 x12 x22 +
2.72849 × 10-7 x13 x22 - 0.000650971 x23 +
0.0000149885 x1 x23 + 1.23768 × 10-6 x24
85.0521 - 3.1744 x1 + 0.00392855 x12 +
0.000778935 x13 + 2.31225 × 10-6 x14 - 1.0553 × 10-7 x15 -
5.80302 x2 + 0.266597 x1 x2 - 0.0029304 x12 x2 -
0.0000394179 x13 x2 + 5.27798 × 10-7 x14 x2 +
0.106664 x22 - 0.00536586 x1 x22 + 0.000122468 x12 x22 -
7.57452 × 10-7 x13 x22 - 0.000355207 x23 -
8.08149 × 10-6 x1 x23 + 4.27373 × 10-6 x24
-30.3157 + 4.35442 x1 - 0.0289735 x12 -
0.000805247 x13 + 4.44482 × 10-6 x14 +
6.50434 × 10-8 x15 + 0.109743 x2 - 0.263079 x1 x2 +
0.00373838 x12 x2 + 0.0000112195 x13 x2 -
3.63596 × 10-7 x14 x2 + 0.0679827 x22 + 0.0038108 x1 x22 -
0.0000804011 x12 x22 + 5.43321 × 10-7 x13 x22 -
0.00155237 x23 + 4.67509 × 10-7 x1 x23 + 6.97476 × 10-6 x24
-65.8969 + 6.03949 x1 + 0.0313806 x12 - 0.00606532 x13 +
0.0000539802 x14 + 5.36013 × 10-7 x15 + 4.62905 x2 -
39.3351 ≤ x1 ≤ 59.2765 &&
x2 ≥ 58.2913 && - 1.25935 x1 + x2 ≤ 8.75438
39.3351 ≤ x1 ≤ 59.2765 &&
27.0321 ≤ x1 < 39.3351 &&
0.675801 x1 - x2 ≤ 5.68434 × 10-14 && x2 < 26.582
39.3351 ≤ x1 ≤ 59.2765 && 42.1908 ≤ x2 < 50.241
39.3351 ≤ x1 ≤ 59.2765 && 50.241 ≤ x2 < 58.2913
27.0321 ≤ x1 < 39.3351 && 33.321 ≤ x2 < 40.0592

```



```

(*LIP approximation of s3|s2*)
Clear[μ2, σ2, m2, ω2, g2, x, l1, u1, u2]
μ2 = Log[40] + 2 m0;
σ2 =  $\sqrt{2}$  n0;
m2 =  $e^{\mu^2}$ ;
ω2 =  $e^{\sigma^2}$ ;
g2[x_] = CDF[LogNormalDistribution[μ2, σ2], x];
l1 = x /. FindRoot[g2[x] - 0.0015, {x, 30}]
u2 = x /. FindRoot[g2[x] - 0.9985, {x, 50}]
u1 = m2 / ω2
22.992
69.7955
38.6813
(* If S2 = l1, then S3|s2 ~ LN(Log[l1] + m0, n0^2) *)
Clear[μl1, gl1, ll1, ull1];
μl1 = Log[l1] + m0;
gl1[x_] = CDF[LogNormalDistribution[μl1, n0], x];
ll1 = x /. FindRoot[gl1[x] - 0.0015, {x, 20}]
ull1 = x /. FindRoot[gl1[x] - 0.9985, {x, 50}]
15.538
34.0721
(* If S2 = u1, then S3|s2 ~ LN(Log[u1] + m0, n0^2) *)
Clear[μu1, gu1, lu1, uu1];
μu1 = Log[u1] + m0;
gu1[x_] = CDF[LogNormalDistribution[μu1, n0], x];
lu1 = x /. FindRoot[gu1[x] - 0.0015, {x, 20}]
uu1 = x /. FindRoot[gu1[x] - 0.9985, {x, 50}]
26.1409
57.3224

```

```
(* If S2 = u2, then S3 | s2 ~ LN(Log[u2] + m0, n0^2) *)
Clear[μu2, gu2, lu2, uu2];
μu2 = Log[u2] + m0;
gu2[x_] = CDF[LogNormalDistribution[μu2, n0], x];
lu2 = x /. FindRoot[gu2[x] - 0.0015, {x, 40}]
uu2 = x /. FindRoot[gu2[x] - 0.9985, {x, 60}]
47.1679
103.431
```

```

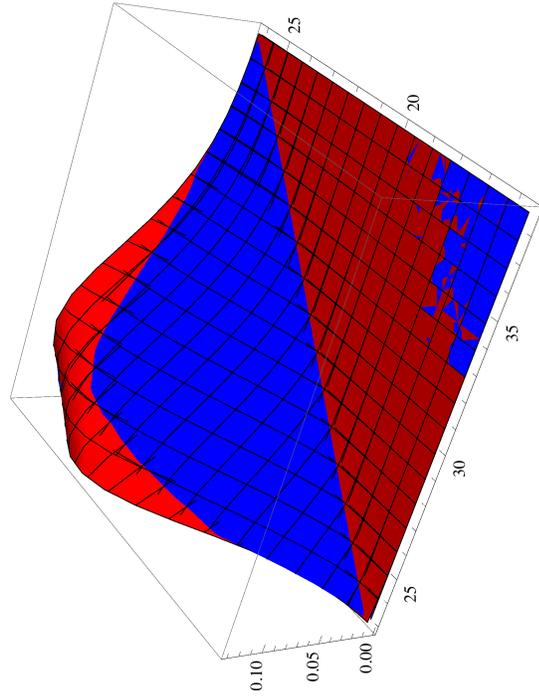
(* Piece # 1 *)
Clear[Y, z, k, n, a, b, c, d, e];
a = 11
b = u1
c = 111
d = lu1
e[x_] = c +  $\frac{d-c}{b-a}$  (x-a)
y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);
Clear[g, n, m, pts1, pts2, pts11, h1];
g[x1_, x2_] = PDF[LogNormalDistribution[Log[x1] + m0, n0], x2]
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h1[x1_, x2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {x1, x2}]], (a ≤ x1 < b) && (e[x1] ≤ x2 < d)}
}}]
MinValue[{h1[x1, x2], (a ≤ x1 ≤ b) && (e[x1] ≤ x2 ≤ d)}, {x1, x2}]
ArgMin[{h1[x1, x2], (a ≤ x1 ≤ b) && (e[x1] ≤ x2 ≤ d)}, {x1, x2}]
Plot3D[{{Style[g[x1, x2], Red], Style[h1[x1, x2], Blue]}}, {x1, a, b}, {x2, c, d}]
(* Plot3D[g[x1, x2] - h1[x1, x2], {x1, a, b}, {x2, c, d}] *)
MaxValue[{Abs[g[x1, x2] - h1[x1, x2]], (a ≤ x1 < b) && (c ≤ x2 < d)}, {x1, x2}]
22.992
38.6813
15.538

```

```

26.1409
15.538 + 0.675801 (-22.992 + x)
{
  3.01572 e-28.5714 (-0.000738889-Log[x1]+Log[x2])^2
  x2 > 0
}
0
True
22.992 ≤ x1 < 38.6813 &&
15.538 + 0.675801 (-22.992 + x1) ≤ x2 < 26.1409
0.0110439 + 0.302365 x1 - 0.0249214 x1^2 +
0.00182781 x1^3 - 0.0000595643 x1^4 + 6.12861 × 10^-7 x1^5 -
0.448784 x2 + 0.0365057 x1 x2 - 0.0053943 x1^2 x2 +
0.00020207 x1^3 x2 - 2.06196 × 10^-6 x1^4 x2 +
0.00063875 x2^2 + 0.00498985 x1 x2^2 -
0.000192094 x1^2 x2^2 + 1.70906 × 10^-6 x1^3 x2^2 -
0.00149767 x2^3 + 0.000032069 x1 x2^3 + 4.08163 × 10^-6 x2^4
}
0
True
0.
{38.6813, 26.1409}

```



0.0228236

```

(* Piece # 2 *)
Clear[a, b, c, d, k, n, Y, z]
a = 11
b = u1
c = lu1
d = (u1+lu1) / 2
Y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h2];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{Y[2 i, n], z[2 j, n]}, g[Y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{Y[2 i+1, n], z[2 j+1, n]}, g[Y[2 i+1, n], z[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h2[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]], (a ≤ s1 < b) && (c ≤ s2 < d)}
}}]
MinValue[{h2[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ d)}, {s1, s2}]
ArgMin[{h2[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h2[s1, s2], Blue]}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h2[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h2[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
22.992
38.6813
26.1409
30.1065

```

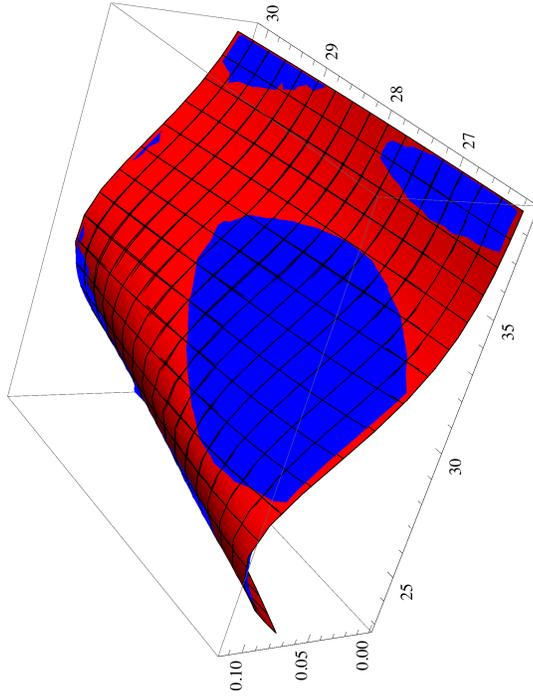
```

-4.34694 - 10.3641 s1 + 0.213972 s12 - 0.000509293 s13 +
0.0000731935 s14 - 1.74344 × 10-6 s15 + 6.41104 s2 +
1.02576 s1 s2 - 0.0220854 s12 s2 - 0.000250209 s13 s2 +
7.12012 × 10-6 s14 s2 - 0.641053 s22 - 0.0306427 s1 s22 +
0.00115808 s12 s22 - 0.0000117664 s13 s22 +
0.0214098 s23 - 0.0000789261 s1 s23 - 0.000162833 s24
0

```

0.000876837

{38.6813, 26.1409}



0.00275841

22.992 ≤ s1 < 38.6813 && 26.1409 ≤ s2 < 30.1065

True

```

(* Piece # 3 *)
Clear[a, b, c, d, k, n, Y, z]
a = 11
b = u1
c = (u11+1u1) / 2
d = u11

Y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h3];
n = 9;
m = (n+1) / 2;
pts1 = Flatten[Table[{{Y[2 i, n], z[2 j, n]}, g[Y[2 i, n]], g[Y[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{Y[2 i+1, n], z[2 j+1, n]}, g[Y[2 i+1, n]], g[Y[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h3[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (c ≤ s2 < d)}
]
MinValue[{h3[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h3[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h3[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h3[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
22.992
38.6813
30.1065
34.0721

```

$22.992 \leq s1 < 38.6813 \ \&\& \ 30.1065 \leq s2 < 34.0721$

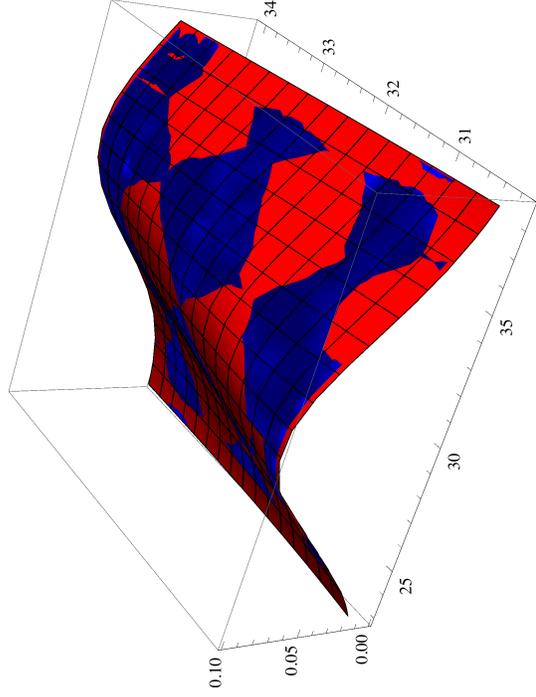
```

2.73565 × 107 + 93 046.3 s1 - 31.1204 s12 + 7.25218 s13 +
0.0706092 s14 - 0.00205511 s15 - 7.65097 × 10-7 s16 -
3.9083 × 10-7 s17 + 1.22694 × 10-8 s18 - 3.7903 × 10-11 s19 -
6.92091 × 106 s2 - 20 552.9 s1 s2 - 2.15096 s12 s2 -
1.58563 s13 s2 - 0.00148635 s14 s2 + 0.000280589 s15 s2 +
2.18603 × 10-6 s16 s2 - 6.829 × 10-8 s17 s2 - 2.8486 × 10-11 s18 s2 +
765 856. s22 + 1948.68 s1 s22 + 1.3061 s12 s22 + 0.123842 s13 s22 -
0.000603671 s14 s22 - 0.0000188095 s15 s22 + 1.68893 × 10-7 s16 s22 +
1.05174 × 10-9 s17 s22 - 48 417.2 s23 - 103.05 s1 s23 -
0.123091 s12 s23 - 0.00424446 s13 s23 + 0.0000485119 s14 s23 +
4.41239 × 10-8 s15 s23 - 3.90208 × 10-9 s16 s23 + 1912.73 s24 +
3.29041 s1 s24 + 0.00482423 s12 s24 + 0.0000502634 s13 s24 -
7.79807 × 10-7 s14 s24 + 5.10526 × 10-9 s15 s24 - 48.3544 s25 -
0.0635263 s1 s25 - 0.0000794573 s12 s25 - 1.98781 × 10-8 s13 s25 +
0.763977 s26 + 0.000685063 s1 s26 + 4.22035 × 10-7 s12 s26 -
0.00689751 s27 - 3.16332 × 10-6 s1 s27 + 0.0000272449 s28
0

```

True

0.00118017



```

0.000138834
(* Piece # 4 *)
Clear[a, b, c, d, e, y, z, k, n]
a = l1
b = u1
c = ul1
d = uu1
e[x_] = c +  $\frac{d-c}{b-a}$  (x-a)
y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);
Clear[n, m, pts1, pts2, pts11, h4];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n]], g[y[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n]], g[y[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h4[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]], (a ≤ s1 < b) && (c ≤ s2 < e[s1])}
}}]
MinValue[{h4[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < e[s1])}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h4[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h4[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h4[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
22.992
38.6813
34.0721
57.3224

```

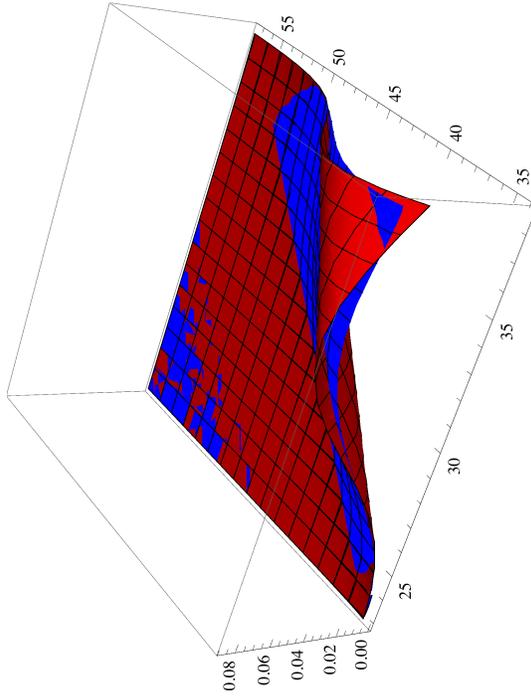
```

34.0721 + 1.48191 (-22.992 + x)
{
0.00503638 - 0.32572 s1 - 0.00232359 s1^2 +
0.00120251 s1^3 - 0.0000445679 s1^4 +
4.63587 × 10-7 s15 + 0.219513 s2 + 0.0219948 s1 s2 -
0.00176639 s12 s2 + 0.000058434 s13 s2 -
5.72571 × 10-7 s14 s2 - 0.0137756 s22 +
0.000433431 s1 s22 - 0.0000197638 s12 s22 +
1.75259 × 10-7 s13 s22 + 0.000142217 s23 +
2.04994 × 10-6 s1 s23 - 1.09665 × 10-6 s24
}

```

True

0.



0.0134794

```

(* Piece # 5 *)
Clear[a, b, c, d, y, z, k, n]
a = u1
b = u2
c = lu1
d = lu2
e[x_] = c +  $\frac{d-c}{b-a}$  (x-a);
y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ])
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ])
Clear[n, m, pts1, pts2, pts11, h5];
n = 7;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h5[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (e[s1] ≤ s2 < d)}
}]
MinValue[{h5[s1, s2], (a ≤ s1 ≤ b) && (e[s1] ≤ s2 ≤ d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h5[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h5[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h5[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
38.6813
69.7955
26.1409
47.1679

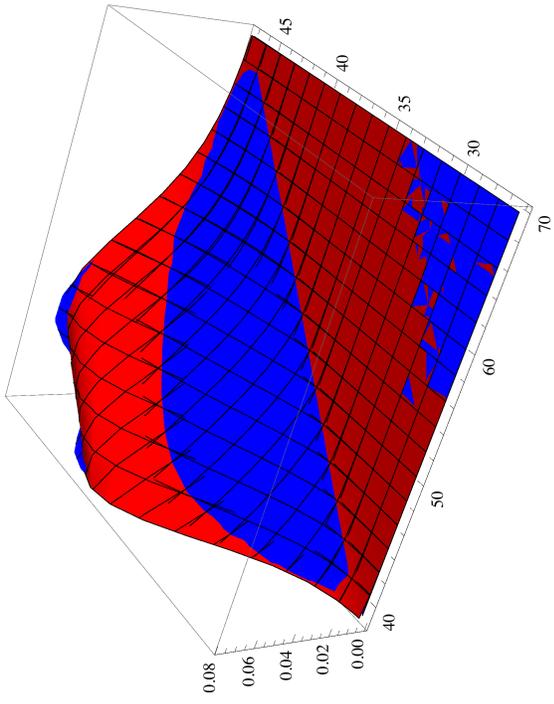
```

$$\begin{aligned}
 & 0.5 \left(108.477 + 31.1142 \operatorname{Cos} \left[\frac{k\pi}{n} \right] \right) \\
 & 0.5 \left(73.3088 + 21.027 \operatorname{Cos} \left[\frac{k\pi}{n} \right] \right) \\
 & \left[\begin{aligned}
 & 0.00845015 - 0.489908 s_1 + 0.0109915 s_1^2 - \\
 & 0.00134597 s_1^3 + 0.0000613108 s_1^4 - 1.27909 \times 10^{-6} s_1^5 + \\
 & 1.30054 \times 10^{-8} s_1^6 - 5.1639 \times 10^{-11} s_1^7 + \\
 & 0.724088 s_2 + 0.0325102 s_1 s_2 + 0.00524443 s_1^2 s_2 - \\
 & 0.000249651 s_1^3 s_2 + 5.12409 \times 10^{-6} s_1^4 s_2 - \\
 & 5.25354 \times 10^{-8} s_1^5 s_2 + 2.10504 \times 10^{-10} s_1^6 s_2 - \\
 & 0.0721255 s_2^2 - 0.00961583 s_1 s_2^2 + 0.000285689 s_1^2 s_2^2 - \\
 & 5.27621 \times 10^{-6} s_1^3 s_2^2 + 5.71101 \times 10^{-8} s_1^4 s_2^2 - \\
 & 2.36288 \times 10^{-10} s_1^5 s_2^2 + 0.0071049 s_2^3 + \\
 & 0.0000370307 s_1 s_2^3 - 4.61662 \times 10^{-8} s_1^2 s_2^3 - \\
 & 1.19015 \times 10^{-8} s_1^3 s_2^3 + 5.6031 \times 10^{-11} s_1^4 s_2^3 - \\
 & 0.000165374 s_2^4 - 5.22615 \times 10^{-8} s_1 s_2^4 + \\
 & 6.05324 \times 10^{-9} s_1^2 s_2^4 + 1.78053 \times 10^{-6} s_2^5 - \\
 & 3.69064 \times 10^{-9} s_1 s_2^5 - 6.85696 \times 10^{-9} s_2^6 \\
 & 0
 \end{aligned} \right] \\
 & 0.
 \end{aligned}$$

38.6813 ≤ s1 < 69.7955 &&

26.1409 + 0.675801 (-38.6813 + s1) ≤ s2 < 47.1679

True



0.0209488

```

(* Piece # 6*)
Clear[a, b, c, d, y, z, k, n];
a = u1
b = u2
c = 1u2
d = (uu1+1u2) / 2

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h6];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n]], g[y[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i+1, n], z[2 j+1, n]}, g[y[2 i+1, n]], g[y[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h6[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (c ≤ s2 < d)}
}]
MinValue[{h6[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ d)}, {s1, s2}]
ArgMin[{h6[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ d)}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h6[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h6[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h6[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
38.6813
69.7955
47.1679
52.2452

```

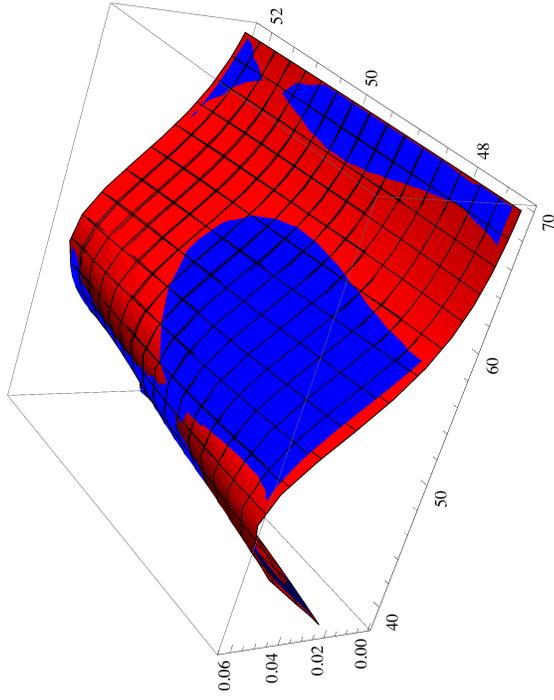
$$38.6813 \leq s1 < 69.7955 \ \&\& \ 47.1679 \leq s2 < 52.2452$$

$$\begin{bmatrix} 488.307 - 2.60128 s1 + 0.0573156 s1^2 - 0.000583962 s1^3 + \\ 7.86221 \times 10^{-6} s1^4 - 5.68183 \times 10^{-8} s1^5 - 38.2036 s2 + \\ 0.102034 s1 s2 - 0.00168641 s1^2 s2 - 9.30597 \times 10^{-6} s1^3 s2 + \\ 1.60062 \times 10^{-7} s1^4 s2 + 1.13854 s2^2 - 0.00160182 s1 s2^2 + \\ 0.0000482531 s1^2 s2^2 - 2.76731 \times 10^{-7} s1^3 s2^2 - \\ 0.0151343 s2^3 - 7.66166 \times 10^{-6} s1 s2^3 + 0.0000788206 s2^4 \\ 0 \end{bmatrix}$$

True

0.

{69.7955, 47.8029}



0.00204684

```

(* Piece # 7 *)
Clear[a, b, c, d, y, z, k, n]
a = u1
b = u2
c = (uu1+lu2) / 2
d = uu1

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h7];
n = 9;
m = (n + 1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m - 1}, {j, 0, m - 1}], 1];
pts2 =
  Flatten[Table[{{y[2 i + 1, n], z[2 j + 1, n]}, g[y[2 i + 1, n], z[2 j + 1, n]]}, {i, 0, m - 1}, {j, 0, m - 1}], 1];
pts11 = Join[pts1, pts2];
h7[s1_, s2_] = Piecewise[{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]]}, (a ≤ s1 < b) && (c ≤ s2 < d)}
}]
MinValue[{h7[s1, s2], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
Plot3D[Style[g[s1, s2], Red], Style[h7[s1, s2], Blue]], {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h7[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h7[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
38.6813
69.7955
52.2452
57.3224

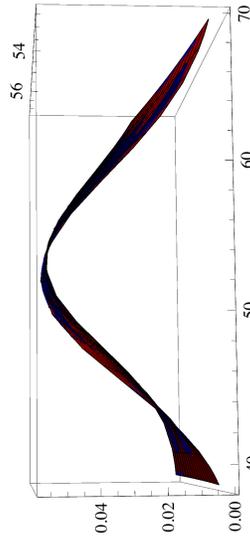
```

$$38.6813 \leq s1 < 69.7955 \ \&\& \ 52.2452 \leq s2 < 57.3224$$

$$\begin{aligned} & 2.77433 \times 10^8 + 190388 \cdot s1 - 252.453 s1^2 + 10.0109 s1^3 - \\ & 0.0715069 s1^4 - 0.0000948063 s1^5 + 3.28169 \times 10^{-6} s1^6 - \\ & 1.38528 \times 10^{-8} s1^7 + 8.1024 \times 10^{-11} s1^8 - 2.48558 \times 10^{-13} s1^9 - \\ & 4.07386 \times 10^7 s2 - 24122.2 s1 s2 + 10.9127 s1^2 s2 - 0.839352 s1^3 s2 + \\ & 0.00705597 s1^4 s2 - 0.0000120492 s1^5 s2 - 9.01851 \times 10^{-8} s1^6 s2 - \\ & 1.18634 \times 10^{-10} s1^7 s2 + 8.19925 \times 10^{-13} s1^8 s2 + 2.61666 \times 10^6 s2^2 + \\ & 1319.96 s1 s2^2 + 0.232328 s1^2 s2^2 + 0.0253947 s1^3 s2^2 - \\ & 0.00023145 s1^4 s2^2 + 6.04173 \times 10^{-7} s1^5 s2^2 + 2.09206 \times 10^{-9} s1^6 s2^2 - \\ & 2.58799 \times 10^{-12} s1^7 s2^2 - 96.023.2 s2^3 - 40.4342 s1 s2^3 - \\ & 0.0207169 s1^2 s2^3 - 0.000333091 s1^3 s2^3 + 3.28629 \times 10^{-6} s1^4 s2^3 - \\ & 1.16157 \times 10^{-8} s1^5 s2^3 - 6.03003 \times 10^{-12} s1^6 s2^3 + 2202.01 s2^4 + \\ & 0.747962 s1 s2^4 + 0.000433238 s1^2 s2^4 + 1.52128 \times 10^{-6} s1^3 s2^4 - \\ & 1.60524 \times 10^{-8} s1^4 s2^4 + 6.14479 \times 10^{-11} s1^5 s2^4 - 32.3138 s2^5 - \\ & 0.00833815 s1 s2^5 - 3.69274 \times 10^{-6} s1^2 s2^5 + 4.92795 \times 10^{-10} s1^3 s2^5 + \\ & 0.296338 s2^6 + 0.0000517256 s1 s2^6 + 1.09691 \times 10^{-8} s1^2 s2^6 - \\ & 0.00155275 s2^7 - 1.37364 \times 10^{-7} s1 s2^7 + 3.55914 \times 10^{-6} s2^8 \end{aligned}$$

True

0.000771761



0.000115752

```

(* Piece # 8 *)
Clear[a, b, c, d, e, y, z, k, n]
a = u1
b = u2
c = uu1
d = uu2

e[x_] = c +  $\frac{d-c}{b-a}$  (x-a);

y[k_, n_] = 0.5 (a + b + (b - a) Cos[ $\frac{k\pi}{n}$ ]);
z[k_, n_] = 0.5 (c + d + (d - c) Cos[ $\frac{k\pi}{n}$ ]);

Clear[n, m, pts1, pts2, pts11, h8];
n = 5;
m = (n+1) / 2;
pts1 = Flatten[Table[{{y[2 i, n], z[2 j, n]}, g[y[2 i, n], z[2 j, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts2 =
  Flatten[Table[{{y[2 i+1, n], z[2 j+1, n]}, g[y[2 i+1, n], z[2 j+1, n]]}, {i, 0, m-1}, {j, 0, m-1}], 1];
pts11 = Join[pts1, pts2];
h8[s1_, s2_] = Piecewise[{{
  {Expand[InterpolatingPolynomial[pts11, {s1, s2}]}], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}
}}]
MinValue[{h8[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
ArgMin[{h8[s1, s2], (a ≤ s1 ≤ b) && (c ≤ s2 ≤ e[s1])}, {s1, s2}]
Plot3D[{{Style[g[s1, s2], Red], Style[h8[s1, s2], Blue]}}, {s1, a, b}, {s2, c, d}]
(* Plot3D[g[s1,s2]-h8[s1,s2],{s1,a,b},{s2,c,d}] *)
MaxValue[{Abs[g[s1, s2] - h8[s1, s2]], (a ≤ s1 < b) && (c ≤ s2 < d)}, {s1, s2}]
38.6813
69.7955
57.3224
103.431

```

```

38.6813 ≤ s1 ≤ 69.7955 &&
57.3224 ≤ s2 ≤ 57.3224 + 1.48191 (-38.6813 + s1)

```

```

0.00289535 - 0.140516 s1 + 0.0041131 s12 -
9.93525 × 10-6 s13 - 1.04563 × 10-6 s14 +
8.71352 × 10-9 s15 + 0.094727 s2 + 0.000287002 s1 s2 -
0.0000925354 s12 s2 + 2.59238 × 10-6 s13 s2 -
1.60709 × 10-8 s14 s2 - 0.00206501 s22 +
0.0000517256 s1 s22 - 1.36216 × 10-6 s12 s22 +
6.8764 × 10-9 s13 s22 + 0.0000102699 s23 +
1.38568 × 10-7 s1 s23 - 5.29214 × 10-8 s24
0

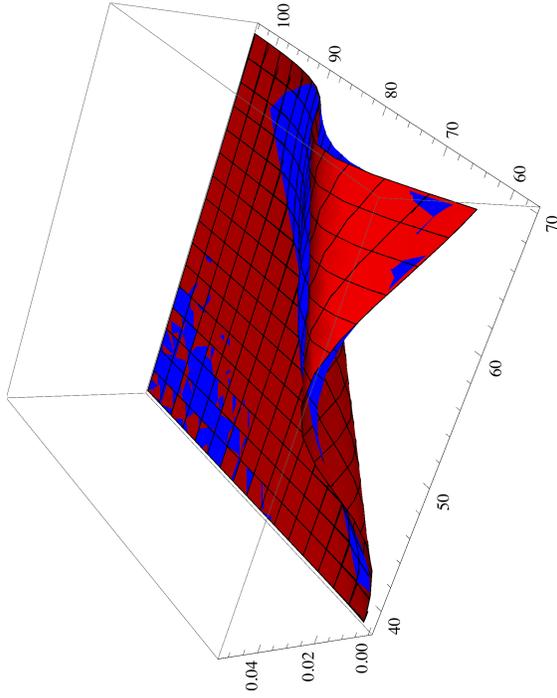
```

True

```

0.
{38.6814, 57.3225}

```



0.0147875

```

(* Putting all 8 pieces together *)
Clear[qlip, h, cq, x1, x2];
h[x1_, x2_] = PiecewiseExpand[
  h1[x1, x2] + h2[x1, x2] + h3[x1, x2] + h4[x1, x2] + h5[x1, x2] + h6[x1, x2] + h7[x1, x2] + h8[x1, x2]];
Timing[N[cq =  $\int_{l11}^{uu2} \int_{l11}^{uu2} h[x1, x2] fs2[x1] dx1 dx2$ ]]
qlip[x1_, x2_] =  $\frac{1}{cq} h[x1, x2]$ 
Plot3D[{Style[g[x1, x2], Red], Style[qlip[x1, x2], Blue]}, {x1, l1, u2}, {x2, l11, uu2}, PlotRange -> All]
{301.23, 1.0598}

```

```

-4.34694 - 10.3641 x1 + 0.213972 x12 - 0.000509293 x13 +
0.0000731935 x14 - 1.74344 × 10-6 x15 + 6.41104 x2 +
1.02576 x1 x2 - 0.0220854 x12 x2 - 0.000250209 x13 x2 +
7.12012 × 10-6 x14 x2 - 0.641053 x22 - 0.0306427 x1 x22 +
0.00115808 x12 x22 - 0.0000117664 x13 x22 +
0.0214098 x23 - 0.0000789261 x1 x23 - 0.000162833 x24
0.00503638 - 0.32572 x1 - 0.00232359 x12 + 0.00120251 x13 -
0.0000445679 x14 + 4.63587 × 10-7 x15 + 0.219513 x2 +
0.0219948 x1 x2 - 0.00176639 x12 x2 + 0.000058434 x13 x2 -
5.72571 × 10-7 x14 x2 - 0.0137756 x22 + 0.000433431 x1 x22 -
0.0000197638 x12 x22 + 1.75259 × 10-7 x13 x22 +
0.000142217 x23 + 2.04994 × 10-6 x1 x23 - 1.09665 × 10-6 x24
0.00289535 - 0.140516 x1 + 0.0041131 x12 - 9.93525 × 10-6 x13 -
1.04563 × 10-6 x14 + 8.71352 × 10-9 x15 + 0.094727 x2 +
0.000287002 x1 x2 - 0.0000925354 x12 x2 + 2.59238 × 10-6 x13 x2 -
1.60709 × 10-8 x14 x2 - 0.00206501 x22 + 0.0000517256 x1 x22 -
1.36216 × 10-6 x12 x22 + 6.8764 × 10-9 x13 x22 +
0.0000102699 x23 + 1.38568 × 10-7 x1 x23 - 5.29214 × 10-8 x24
0.0110439 + 0.302365 x1 - 0.0249214 x12 + 0.00182781 x13 -
0.0000595643 x14 + 6.12861 × 10-7 x15 - 0.448784 x2 +
0.0365057 x1 x2 - 0.0053943 x12 x2 + 0.00020207 x13 x2 -
2.06196 × 10-6 x14 x2 + 0.00063875 x22 + 0.00498985 x1 x22 -
0.000192094 x12 x22 + 1.70906 × 10-6 x13 x22 -
0.00149767 x23 + 0.000032069 x1 x23 + 4.08163 × 10-6 x24
488.307 2.60128 x1 0.0573156 x12 0.000593062 x13
38.6813 - x1 - 60.7055 x12 16.70 - x2 - 59.2452
22.992 ≤ x1 < 38.6813 && 26.1409 ≤ x2 < 30.1065
22.992 ≤ x1 < 38.6813 &&
x2 ≥ 34.0721 && -1.48191 x1 + x2 < 9.9476 × 10-14
38.6813 ≤ x1 ≤ 69.7955 &&
x2 ≥ 57.3224 && -1.48191 x1 + x2 ≤ 2.13163 × 10-14
22.992 ≤ x1 < 38.6813 &&
0.675801 x1 - x2 ≤ -8.88178 × 10-15 && x2 < 26.1409

```

```

T00.0001 - 4.000140001 T 0.000140001 - 0.000140001 0.000140001 T
7.86221 × 10-6 x14 - 5.68183 × 10-8 x15 - 38.2036 x2 +
0.102034 x1 x2 - 0.00168641 x12 x2 - 9.30597 × 10-6 x13 x2 +
1.60062 × 10-7 x14 x2 + 1.13854 x22 - 0.00160182 x1 x22 +
0.0000482531 x12 x22 - 2.76731 × 10-7 x13 x22 -
0.0151343 x23 - 7.66166 × 10-6 x1 x23 + 0.0000788206 x24
0.00845015 - 0.489908 x1 + 0.0109915 x12 - 0.00134597 x13 +
0.0000613108 x14 - 1.27909 × 10-6 x15 + 1.30054 × 10-8 x16 -
5.1639 × 10-11 x17 + 0.724088 x2 + 0.0325102 x1 x2 +
0.00524443 x12 x2 - 0.000249651 x13 x2 + 5.12409 × 10-6 x14 x2 -
5.25354 × 10-8 x15 x2 + 2.10504 × 10-10 x16 x2 - 0.0721255 x22 -
0.00961583 x1 x22 + 0.000285689 x12 x22 - 5.27621 × 10-6 x13 x22 +
5.71101 × 10-8 x14 x22 - 2.36288 × 10-10 x15 x22 +
0.0071049 x23 + 0.0000370307 x1 x23 - 4.61662 × 10-8 x12 x23 -
1.19015 × 10-8 x13 x23 + 5.6031 × 10-11 x14 x23 -
0.000165374 x24 - 5.22615 × 10-8 x1 x24 + 6.05324 × 10-9 x12 x24 +
1.78053 × 10-6 x25 - 3.69064 × 10-9 x1 x25 - 6.85696 × 10-9 x26
2.77433 × 108 + 190388. x1 - 252.453 x12 + 10.0109 x13 -
0.0715069 x14 - 0.0000948063 x15 + 3.28169 × 10-6 x16 -
1.38528 × 10-8 x17 + 8.1024 × 10-11 x18 - 2.48558 × 10-13 x19 -
4.07386 × 107 x2 - 24122.2 x1 x2 + 10.9127 x12 x2 -
0.839352 x13 x2 + 0.00705597 x14 x2 - 0.0000120492 x15 x2 -
9.01851 × 10-8 x16 x2 - 1.18634 × 10-10 x17 x2 +
8.19925 × 10-13 x18 x2 + 2.61666 × 106 x22 + 1319.96 x1 x22 +
0.232328 x12 x22 + 0.0253947 x13 x22 - 0.00023145 x14 x22 +
6.04173 × 10-7 x15 x22 + 2.09206 × 10-9 x16 x22 -
2.58799 × 10-12 x17 x22 - 96023.2 x23 - 40.4342 x1 x23 -
0.0207169 x12 x23 - 0.00033091 x13 x23 + 3.28629 × 10-6 x14 x23 -
1.16157 × 10-8 x15 x23 - 6.03003 × 10-12 x16 x23 + 2202.01 x24 +
0.747962 x1 x24 + 0.000432328 x12 x24 + 1.52128 × 10-6 x13 x24 -
1.60524 × 10-8 x14 x24 + 6.14479 × 10-11 x15 x24 - 32.3138 x25 -
0.00833815 x1 x25 - 3.69274 × 10-6 x12 x25 + 4.92795 × 10-10 x13 x25 +
0.296338 x26 + 0.0000517256 x1 x26 + 1.09691 × 10-8 x12 x26 -
0.00155275 x27 - 1.37364 × 10-7 x1 x27 + 3.55914 × 10-6 x28
2.73565 × 107 + 93046.3 x1 - 31.1204 x12 + 7.25218 x13 +
0.0706092 x14 - 0.00205511 x15 - 7.65097 × 10-7 x16 -
3.9083 × 10-7 x17 + 1.22694 × 10-8 x18 - 3.7903 × 10-11 x19 -
6.92091 × 106 x2 - 20552.9 x1 x2 - 2.15096 x12 x2 -
1.58563 x13 x2 - 0.00148635 x14 x2 + 0.000280589 x15 x2 +
2.18603 × 10-6 x16 x2 - 6.829 × 10-8 x17 x2 -

```

0.943578

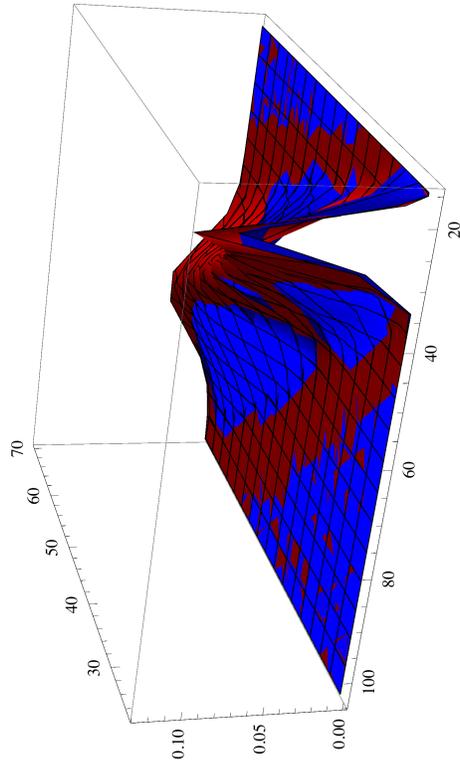
38.6813 ≤ x1 < 69.7955 &&
0.675801 x1 - x2 ≤ 3.90799 × 10⁻¹⁴ && x2 < 47.1679

38.6813 ≤ x1 < 69.7955 && 52.2452 ≤ x2 < 57.3224

22.992 ≤ x1 < 38.6813 && 30.1065 ≤ x2 < 34.0721

$$\begin{pmatrix} 2.8486 \times 10^{-11} x_1^8 x_2 + 765856. x_2^2 + 1948.68 x_1 x_2^2 + \\ 1.3061 x_1^2 x_2^2 + 0.123842 x_1^3 x_2^2 - 0.000603671 x_1^4 x_2^2 - \\ 0.0000188095 x_1^5 x_2^2 + 1.68893 \times 10^{-7} x_1^6 x_2^2 + \\ 1.05174 \times 10^{-9} x_1^7 x_2^2 - 48417.2 x_2^3 - 103.05 x_1 x_2^3 - \\ 0.123091 x_1^2 x_2^3 - 0.00424446 x_1^3 x_2^3 + 0.0000485119 x_1^4 x_2^3 + \\ 4.41239 \times 10^{-8} x_1^5 x_2^3 - 3.90208 \times 10^{-9} x_1^6 x_2^3 + 1912.73 x_2^4 + \\ 3.29041 x_1 x_2^4 + 0.00482423 x_1^2 x_2^4 + 0.0000502634 x_1^3 x_2^4 - \\ 7.79807 \times 10^{-7} x_1^4 x_2^4 + 5.10526 \times 10^{-9} x_1^5 x_2^4 - 48.3544 x_2^5 - \\ 0.0635263 x_1 x_2^5 - 0.0000794573 x_1^2 x_2^5 - 1.98781 \times 10^{-8} x_1^3 x_2^5 + \\ 0.763977 x_2^6 + 0.000685063 x_1 x_2^6 + 4.22035 \times 10^{-7} x_1^2 x_2^6 - \\ 0.00689751 x_2^7 - 3.16332 \times 10^{-6} x_1 x_2^7 + 0.0000272449 x_2^8 \\ 0 \end{pmatrix}$$

True



```

(*Marginalize S3 and D3*)
Clear[v3, v3prime, v2];

v3[s3_] = e-3 r t Max[35 - s3, 0]
Timing[v3prime[s2_] = PiecewiseExpand[ $\int_0^{35} v3[s3] \text{qlip}[s2, s3] \text{dis}3$ ]]
v2[s2_] = e-2 r t Max[35 - s2, 0]
Plot[v3prime[s2], {s2, 23, 53}]
Plot[{v3prime[s2], v2[s2]}, {s2, 23, 53}, PlotStyle → {Thick, Red}]
bs2 = s2 /. FindRoot[v2[s2] - v3prime[s2], {s2, 23}]

0.971935 Max[0, 35 - s3]
{
  0.
  9417.84027730523 - 3217.174505447693 s2 + 457.0971320890711 s22 -
  36.21592882145977 s23 + 1.785384387748644 s24 - 0.05721718435464534 s25 +
  0.001197593815242200 s26 - 0.00001583642610431321 s27 +
  1.203407887357981 × 10-7 s28 - 4.012324327999484 × 10-10 s29
  9273.64869980459 - 3232.351952393582 s2 + 464.4488769283489 s22 -
  36.99846317193435 s23 + 1.825375088662643 s24 - 0.05831631335985938 s25 +
  0.001213255181242093 s26 - 0.00001592741914019565 s27 +
  1.203407887357981 × 10-7 s28 - 4.012324327999484 × 10-10 s29
  15.5716,
  9416.06966415970 - 3216.769332468704 s2 + 457.0630113714257 s22 -
  36.21457899297507 s23 + 1.785359018222564 s24 - 0.05721700131492950 s25 +
  0.001197593815242200 s26 - 0.00001583642610431321 s27 +
  1.203407887357981 × 10-7 s28 - 4.012324327999484 × 10-10 s29
  6544.871645423386 - 1084.049828121013 s2 + 78.85785359975835 s22 -
  3.298713182512359 s23 + 0.08724339910577998 s24 - 0.001508925967934041 s25 +
  0.00001701007272921493 s26 - 1.199559213147473 × 10-7 s27 +
  4.767708726894697 × 10-10 s28 - 8.0405988819198726 × 10-13 s29
}
0.981201 Max[0, 35 - s2]

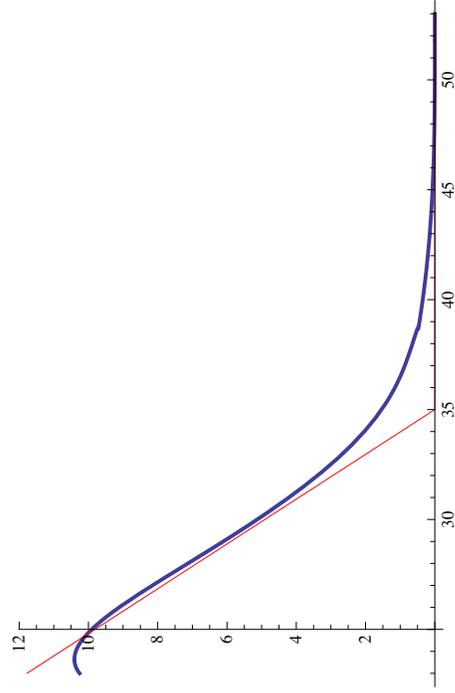
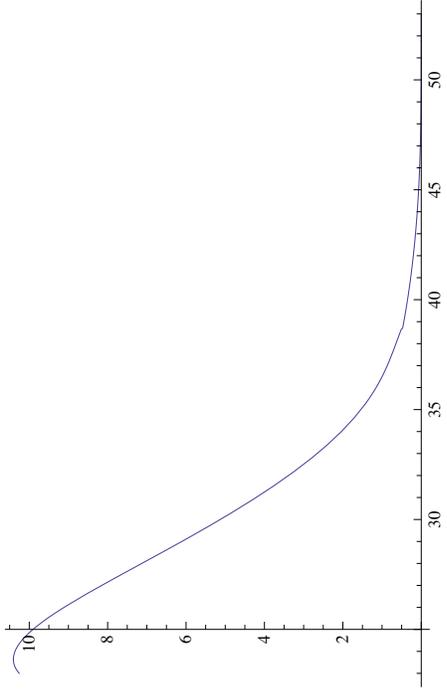
```

s2 ≥ 51.7904 || s
s2 == 22.992

22.992 < s2 ≤ 23.

23.6181 < s2 < 38

True



24.7469

```

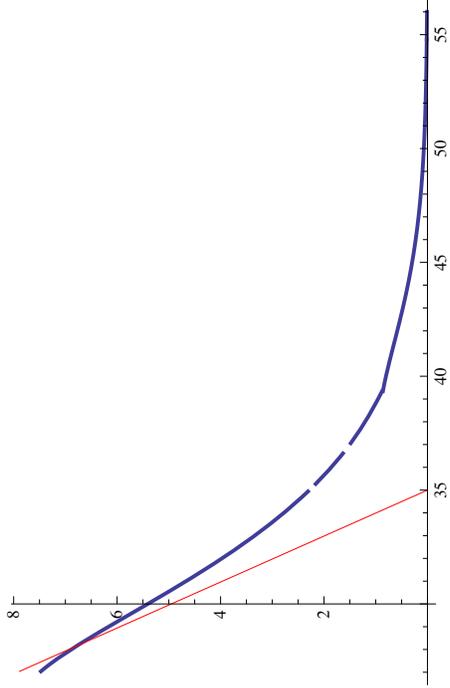
(* Marginalize s2 and D2*)
Clear[v2prime, v1];
Timing[v2prime[s1_] = PiecewiseExpand[
  Assuming[0 < s2 && 0 < s1,  $\int_0^{bs2} e^{-2xt} (35 - s2) \text{glip}[s1, s2] ds2 + \int_{bs2}^{\infty} \text{glip}[s1, s2] v3prime[s2] ds2$  ]]]]
v1[s1_] = e-x t Max[35 - s1, 0]
Plot[{v2prime[s1], v1[s1]}, {s1, 27.04, 56}, PlotStyle -> {Thick, Red}]
bs1 = s1 /. FindRoot[v1[s1] - v2prime[s1], {s1, 28}]

```

```

0.
104.228 - 29.8326 s1 + 2.6259 s12 - 0.100795 s13 + 0.00178214 s14 - 0.0000119434 s15
- 20.7138 + 1.75833 s1 - 0.0571624 s12 +
  0.000888969 s13 - 6.5608 × 10-6 s14 + 1.80192 × 10-8 s15
- 5758.810172008773 + 1055.217912157842 s1 - 81.71246461105299 s12 +
  3.470835100624118 s13 - 0.08717450268259004 s14 + 0.001288730552530572 s15 -
  0.00001030092584979610 s16 + 3.387838043072102 × 10-8 s17
15840.57795505419 - 8901.239240106074 s1 + 2039.075989893900 s12 -
  273.3349585754317 s13 + 24.66040715812114 s14 - 1.603360565276065 s15 +
  0.07790695810322976 s16 - 0.002883508916651445 s17 +
  0.00008194722348749106 s18 - 1.786898929785377 × 10-6 s19 +
  2.962083269313683 × 10-8 s110 - 3.660398918484226 × 10-10 s111 +
  3.257616460984637 × 10-12 s112 - 1.965491788502346 × 10-14 s113 +
  7.158741951345138 × 10-17 s114 - 1.178799762436335 × 10-19 s115
- 188999. + 41587.6 s1 - 4237.5 s12 + 265.125 s13 - 11.3856 s14 +
  0.355292 s15 - 0.00831755 s16 + 0.000148644 s17 - 2.0429 × 10-6 s18 +
  2.15714 × 10-8 s19 - 1.73366 × 10-10 s110 + 1.03988 × 10-12 s111 -
  4.4974 × 10-15 s112 + 1.32048 × 10-17 s113 - 2.34457 × 10-20 s114 + 1.887 × 10-23 s115
- 133557. + 49316.7 s1 - 8133.95 s12 + 802.479 s13 - 53.3381 s14 + 2.54308 s15 -
  0.0901915 s16 + 0.0024295 s17 - 0.0000502175 s18 + 7.97676 × 10-7 s19 -
  9.66842 × 10-9 s110 + 8.78898 × 10-11 s111 - 5.80404 × 10-13 s112 +
  2.62995 × 10-15 s113 - 7.31458 × 10-18 s114 + 9.4163 × 10-21 s115
- 49521.3 + 23780.9 s1 - 4831.12 s12 + 570.17 s13 - 44.511 s14 + 2.46105 s15 -
  0.100271 s16 + 0.00308069 s17 - 0.0000722201 s18 + 1.29523 × 10-6 s19 -
  1.76608 × 10-8 s110 + 1.8006 × 10-10 s111 - 1.33027 × 10-12 s112 +
  6.72943 × 10-15 s113 - 2.08601 × 10-17 s114 + 2.9894 × 10-20 s115
} 22.9829,
0.990556 Max[0, 35 - s1]
s1 > 59.2765 || s1
s1 == 36.6186
s1 == 54.7927
34.9483 ≤ s1 < 36.
27.0321 ≤ s1 < 34.
54.7927 < s1 ≤ 59.
39.3351 ≤ s1 < 54.
True

```



28.1478

(* Marginalize S1 and D1 *)

Timing [N [Assuming [0 < s1, $\int_0^{bs1} e^{-r \cdot t} (35 - s1) fs1[s1] ds1 + \int_{bs1}^{\infty} v2prime[s1] fs1[s1] ds1$]]]

{5.78167, 1.21901}