

# **Verifying inequalities over real numbers in Coq**

Schloß Dagstuhl

11 January 2005

Roland Zumkeller

supervised by Gilles Dowek and Benjamin Werner

LogiCal, INRIA Futurs

## Motivation

- Inequalities over real numbers occur throughout T. Hales' proof of the Kepler conjecture.
- Example: linear relaxation (next talk) requires us to proof bounds on non-linear functions.

$x_1, x_2, x_5, x_6$  in  $[4 ; 6.3001]$                        $x_3$  in  $[6.3001 ; 8]$   
 $x_4$  in  $[6.3001 ; 6.3001]$  (sic)

$$\begin{aligned} & \pi/2 + \arctan( \\ & \quad -( -x_2*x_3 - x_1*x_4 + x_2* x_5 + \\ & \quad \quad x_3*x_6 - x_5*x_6 + x_1* (- x_1 + x_2 + x_3 - x_4 + x_5 + x_6)) \\ & / \sqrt{4*x_1*(x_1*x_4*(- x_1 + x_2 + x_3 - x_4 + x_5 + x_6) + \\ & \quad x_2*x_5*( x_1 - x_2 + x_3 + x_4 - x_5 + x_6) + \\ & \quad x_3*x_6*( x_1 + x_2 - x_3 + x_4 + x_5 - x_6) \\ & \quad - x_2*x_3*x_4 - x_1*x_3*x_5 - x_1*x_2*x_6 - x_4*x_5*x_6 )}) > 1.033 \end{aligned}$$

- The primitive status of programs in Coq allows to integrate reasoning and computation efficiently.

## Interval Arithmetic (1/2)

- The set of intervals is defined as  $\mathbb{I} = (\mathbb{R} \times \mathbb{R}) \cup \{\perp\}$
- $\hat{f} : \mathbb{I}^n \rightarrow \mathbb{I}$  is *correct* for  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  iff:  

$$\forall \vec{r} \in \mathbb{R}^n, \vec{i} \in \mathbb{I}^n. (\forall j < n. \vec{r}_j \in \vec{i}_j) \rightarrow f \vec{r} \in \hat{f} \vec{i}$$
- Expressions are modelled by an inductive type (signature).
- Two interpretations for each function symbol – a Coq module type with two implementations, and a functor connecting them:

	$\mathcal{R}$	$\mathcal{I}$	$\sim$ (compatibility)
num	$\mathbb{R}$	$(\mathbb{R} \times \mathbb{R}) \cup \{\perp\}$	$\in$
prop	Prop	$\{\text{T}, \text{F}, ?, \perp\}$	$\forall P. P \rightarrow P \sim \text{T}$ $\forall P. \neg P \rightarrow P \sim \text{F}$ $\forall P. P \sim ?$
$+$ : num $\rightarrow$ num $\rightarrow$ num	$+$	$\hat{+}$	
$<$ : num $\rightarrow$ num $\rightarrow$ prop	$<$	$\hat{<}$	

## Interval Arithmetic (2/2)

- **Examples:**

- $(a, b) \hat{+} (c, d) := (a + c, b + d)$  and  $\perp \hat{+} \dots = \dots \hat{+} \perp = \perp$

- $1 \hat{/} (a, b) := \begin{cases} (\frac{1}{b}, \frac{1}{a}) & \text{if } a > 0 \vee b < 0 \\ \perp & \text{if } a \leq 0 \leq b \end{cases}$  and  $1 \hat{/} \perp := \perp$

- $(a, b) \hat{<} (c, d) := \begin{cases} T & \text{if } b < c \\ F & \text{if } d < a \\ ? & \text{otherwise} \end{cases}$  and  $\perp \hat{<} \dots = \dots \hat{<} \perp = \perp$

- **Applying correctness:**  $x \in (1, 3) \wedge y \in (4, 5) \wedge z \in (2, 4)$  implies

- $x + y \in (1, 3) \hat{+} (4, 5) = (5, 8)$

- $x < y$  and  $\neg(y < x)$ , since  $3 < 4$

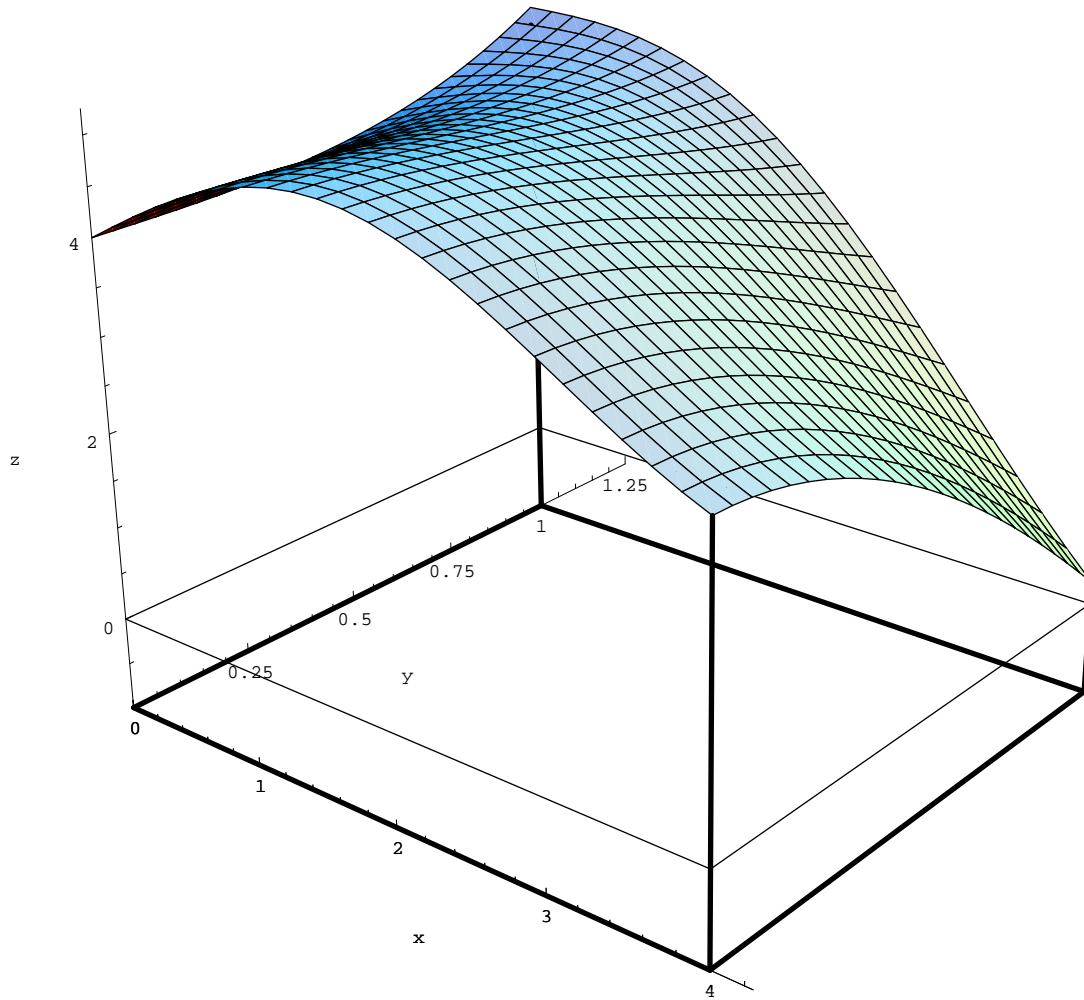
- ... but it does neither imply  $x < z$ , nor  $z < x$

## The Dependence Problem

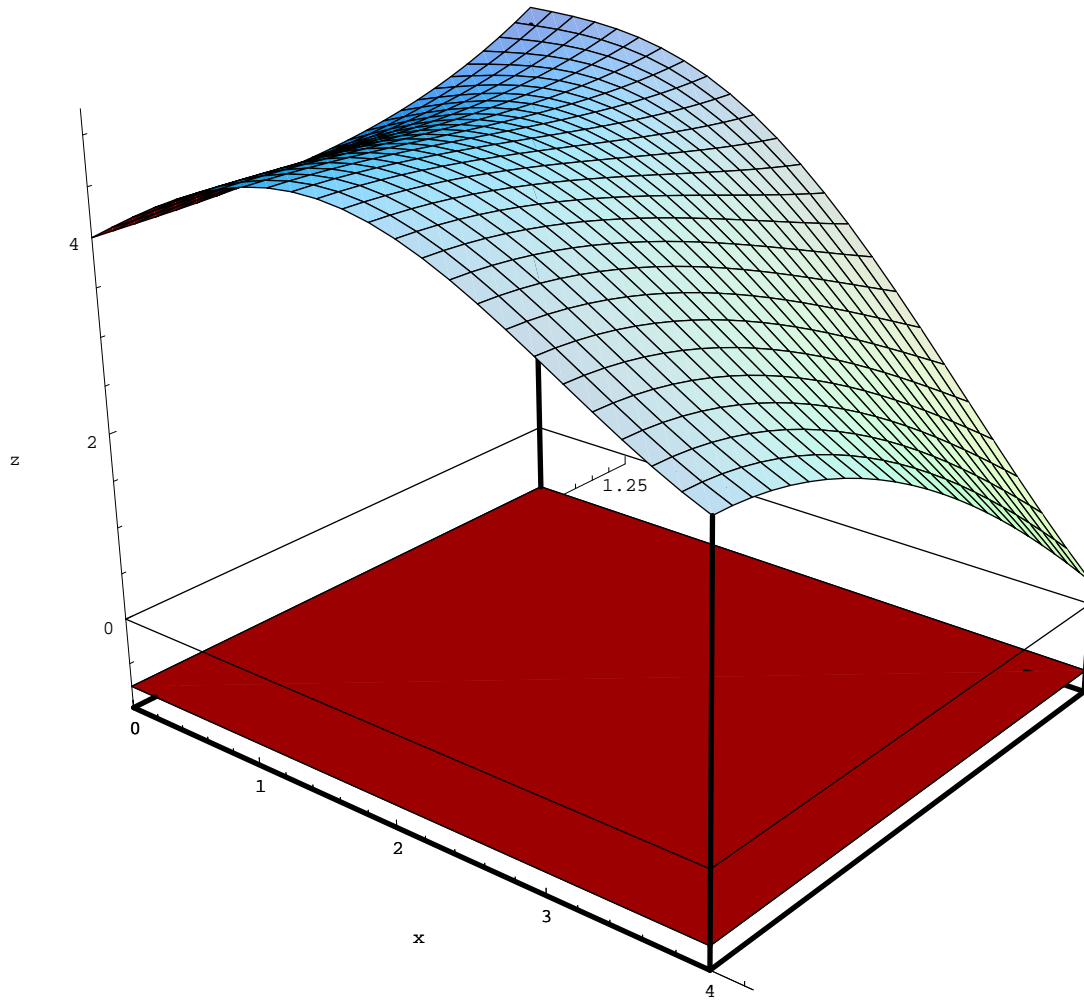
- $(a, b) \hat{-} (c, d) = (a - d, b - c)$
- Interval arithmetic fails to take into account variable dependencies: for  $x, y \in (2, 5)$  it yields the same result on  $x - y$  and  $x - x$ , namely  $(-3, 3)$
- ... but:  $x - x = 0$
- Term rewriting can improve the result, but doesn't solve the problem in general (e.g. take  $x - \sin x$ ). Ideas?
- “Solution”: break up the interval into pieces, evaluate each term, take the union of the results.  
 $((2, 3) \hat{-} (2, 3)) \cup ((3, 5) \hat{-} (3, 5)) = (-1, 1) \cup (-2, 2) = (-2, 2)$   
This is better (more precise) than  $(-3, 3)$ .

## Branch & Bound

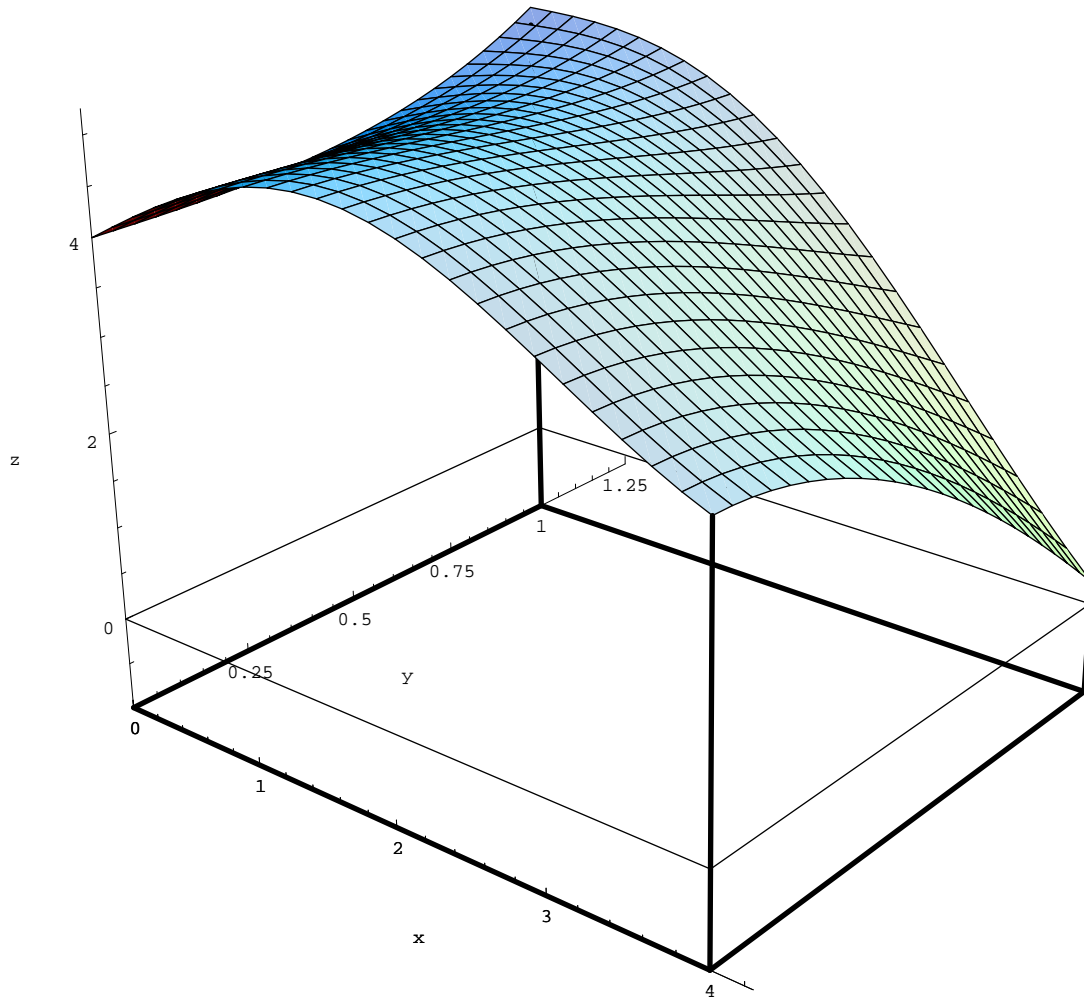
- Most of the inequalities contain 6 variables.
- Which variable's interval is to be cut? (choice: the largest)
- Evaluating formal partial derivatives can establish monotonicity in one direction. In this case we can reduce the domain to its border:
  - Goal: show positivity of  $x^2 - 2$  on  $x \in (1.42, 2)$
  - automated detection of monotonicity:  
evaluating  $\frac{d(x^2-2)}{dx} = 2x$  on  $(1.42, 2)$  gives  $(2.84, 4)$
  - Remaining subgoal: show positivity of  $x^2 - 2$  for  $x = 1.42$



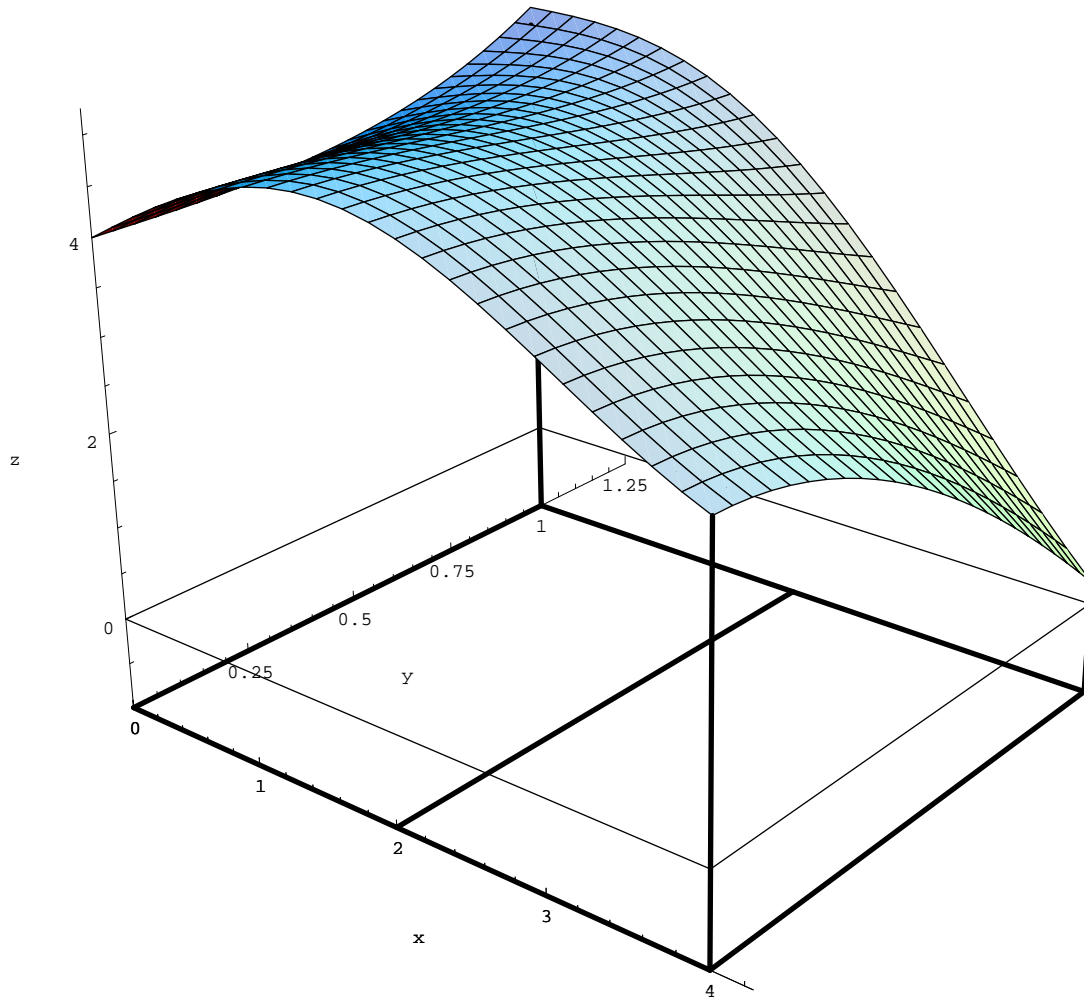
$$\sin x + y^2(y - x) + 4$$



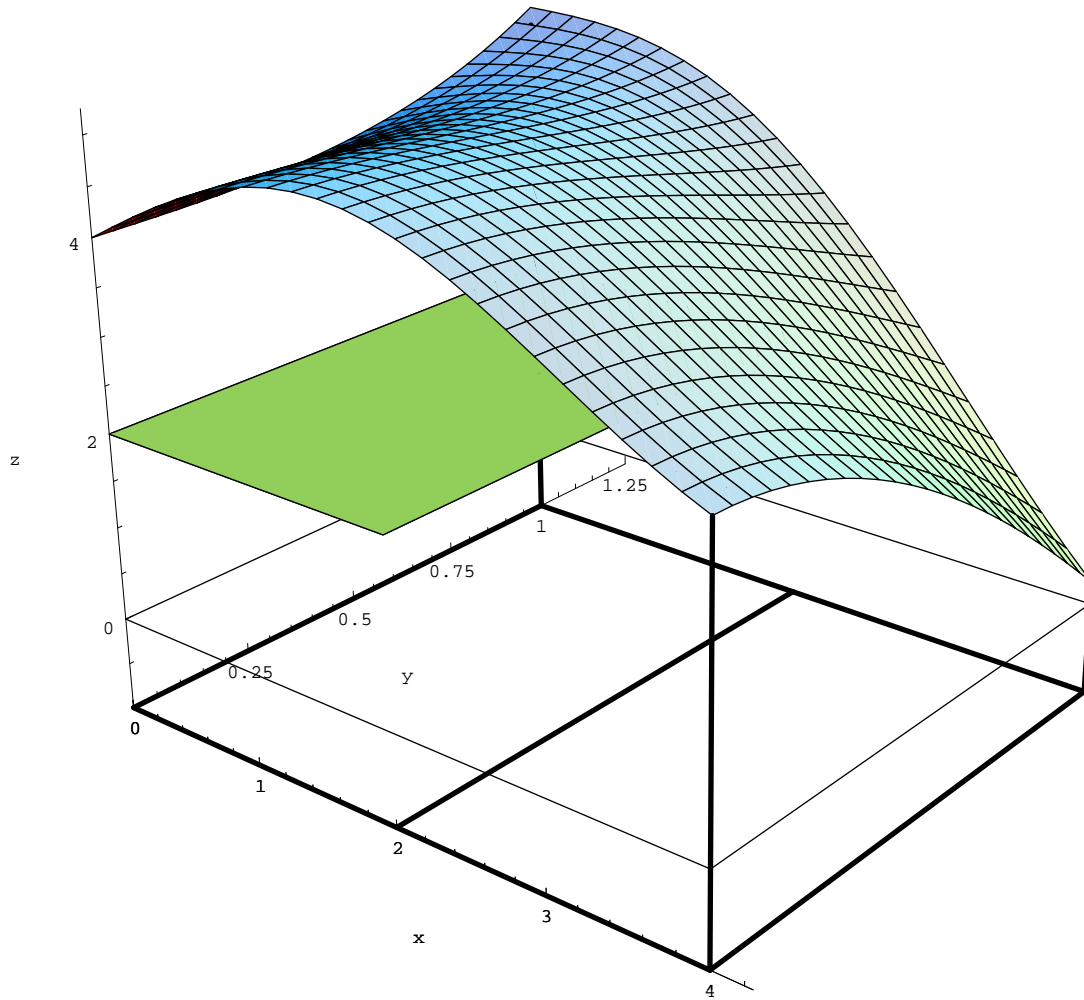
$$\sin x + y^2(y - x) + 4$$



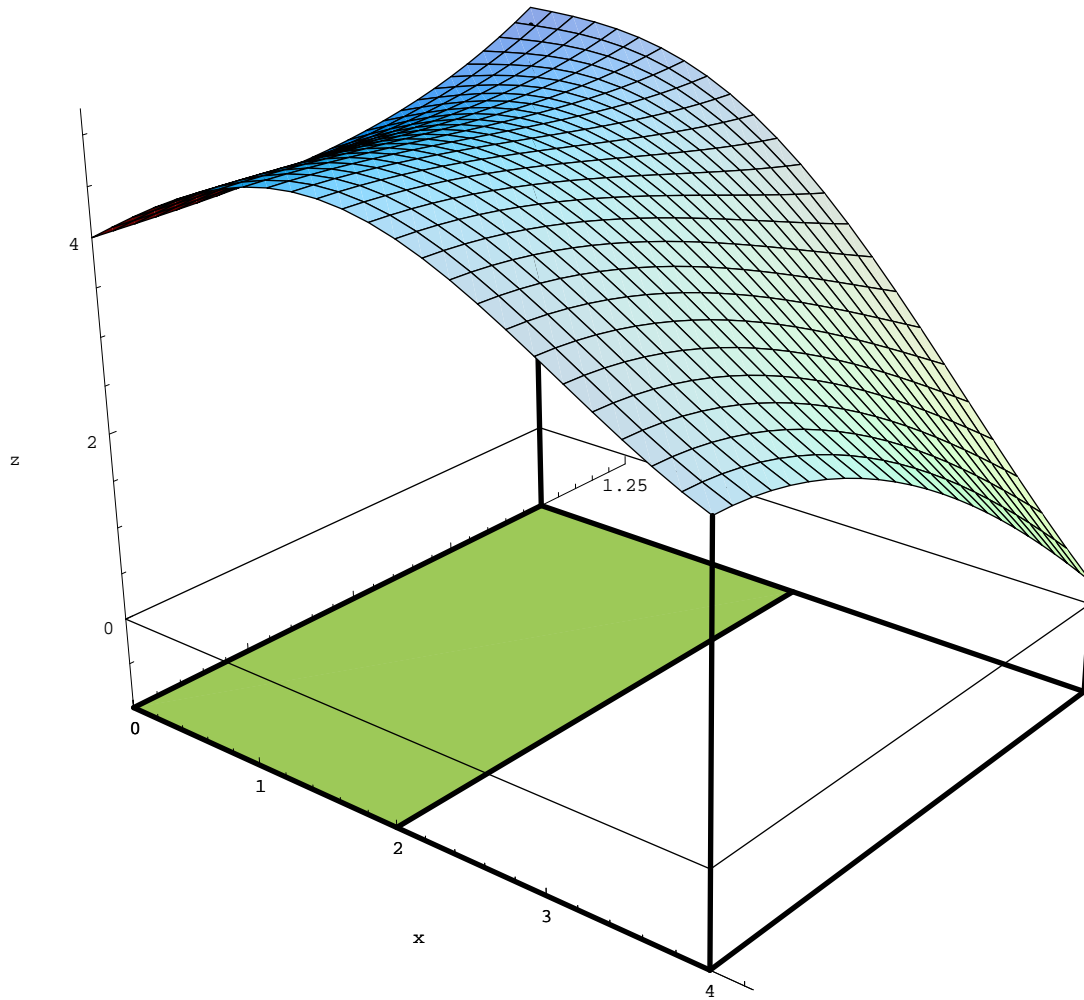
$$\sin x + y^2(y - x) + 4$$



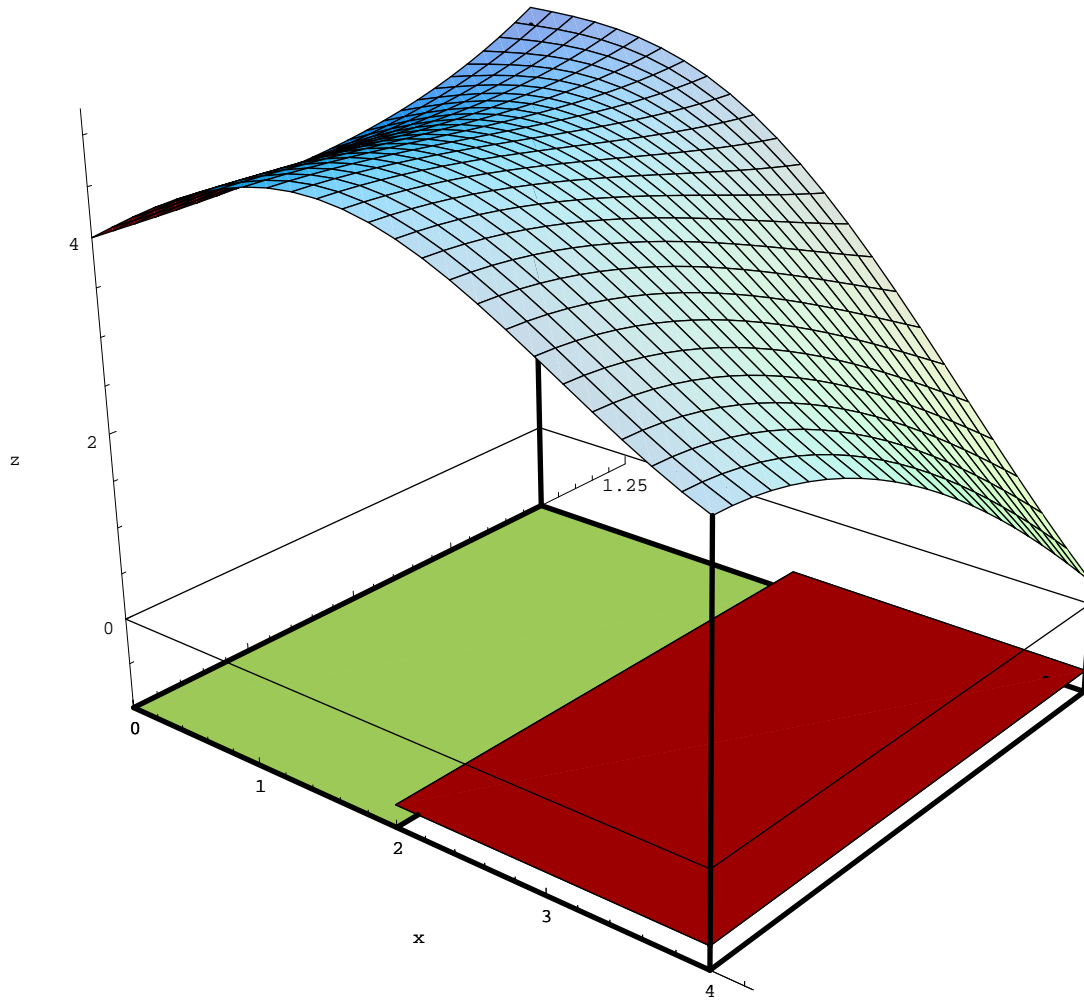
$$\sin x + y^2(y - x) + 4$$



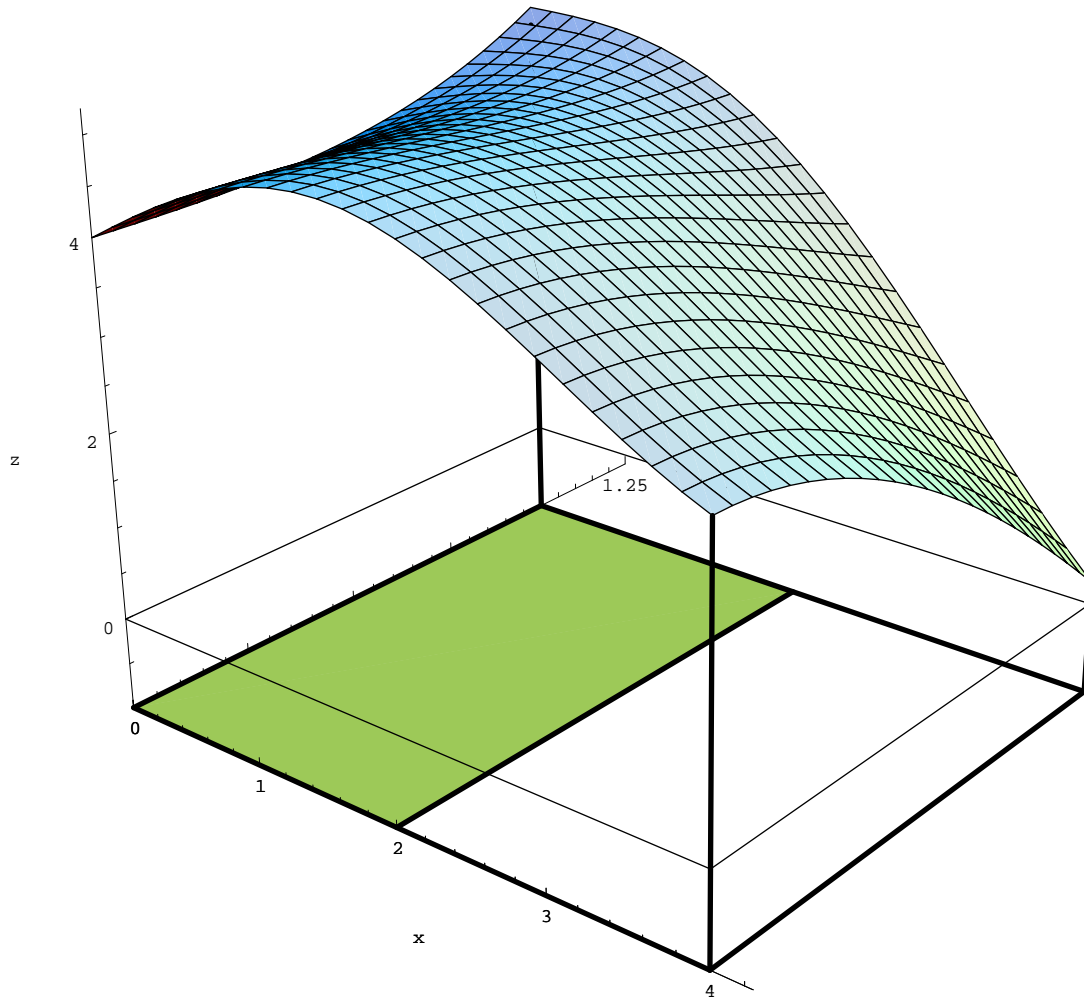
$$\sin x + y^2(y - x) + 4$$



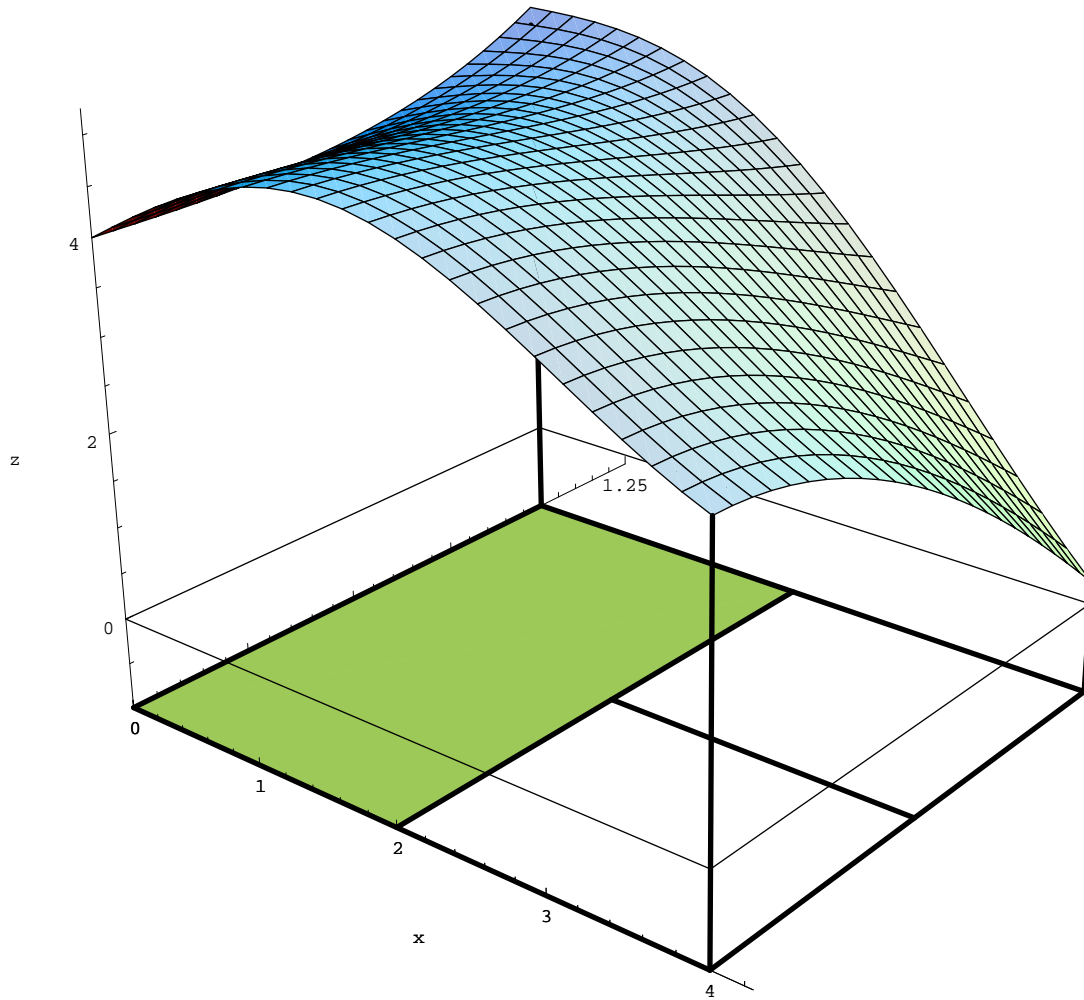
$$\sin x + y^2(y - x) + 4$$



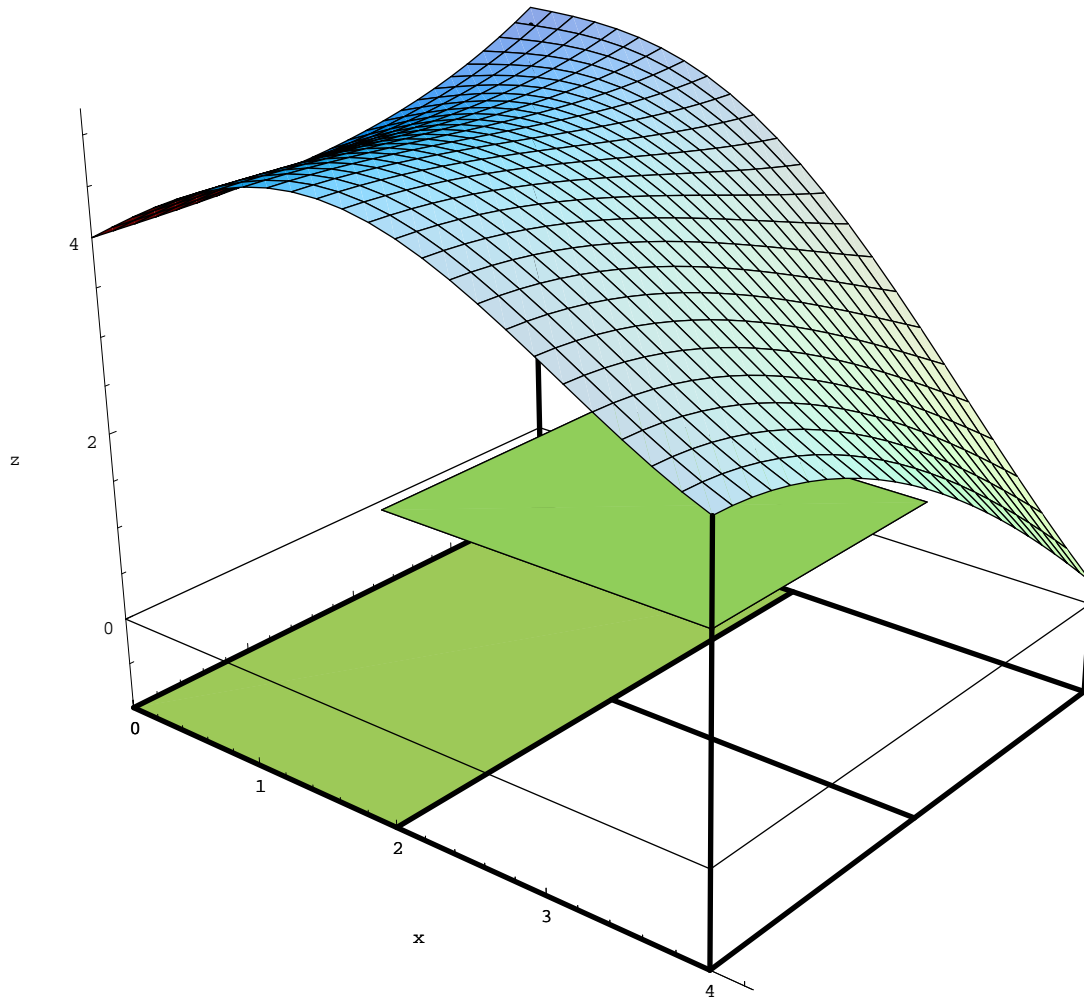
$$\sin x + y^2(y - x) + 4$$



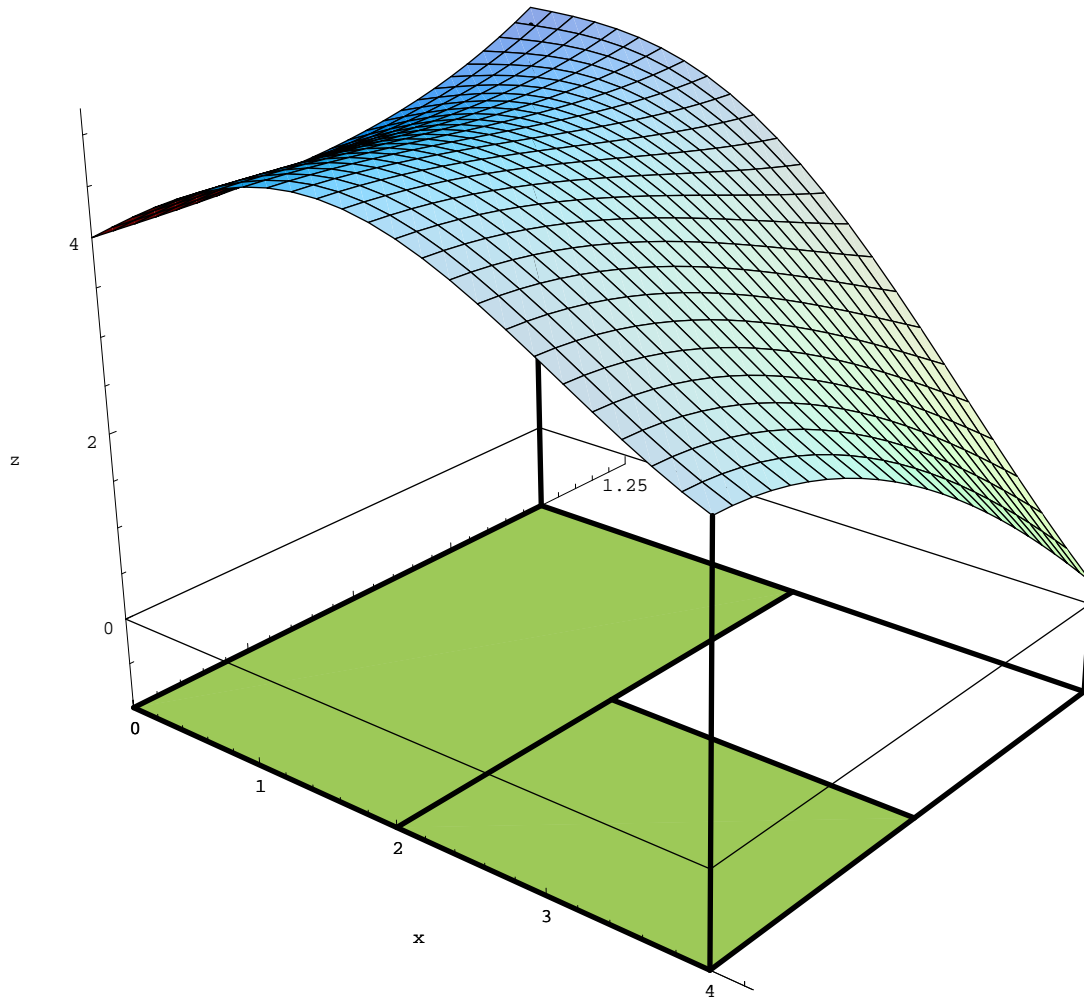
$$\sin x + y^2(y - x) + 4$$



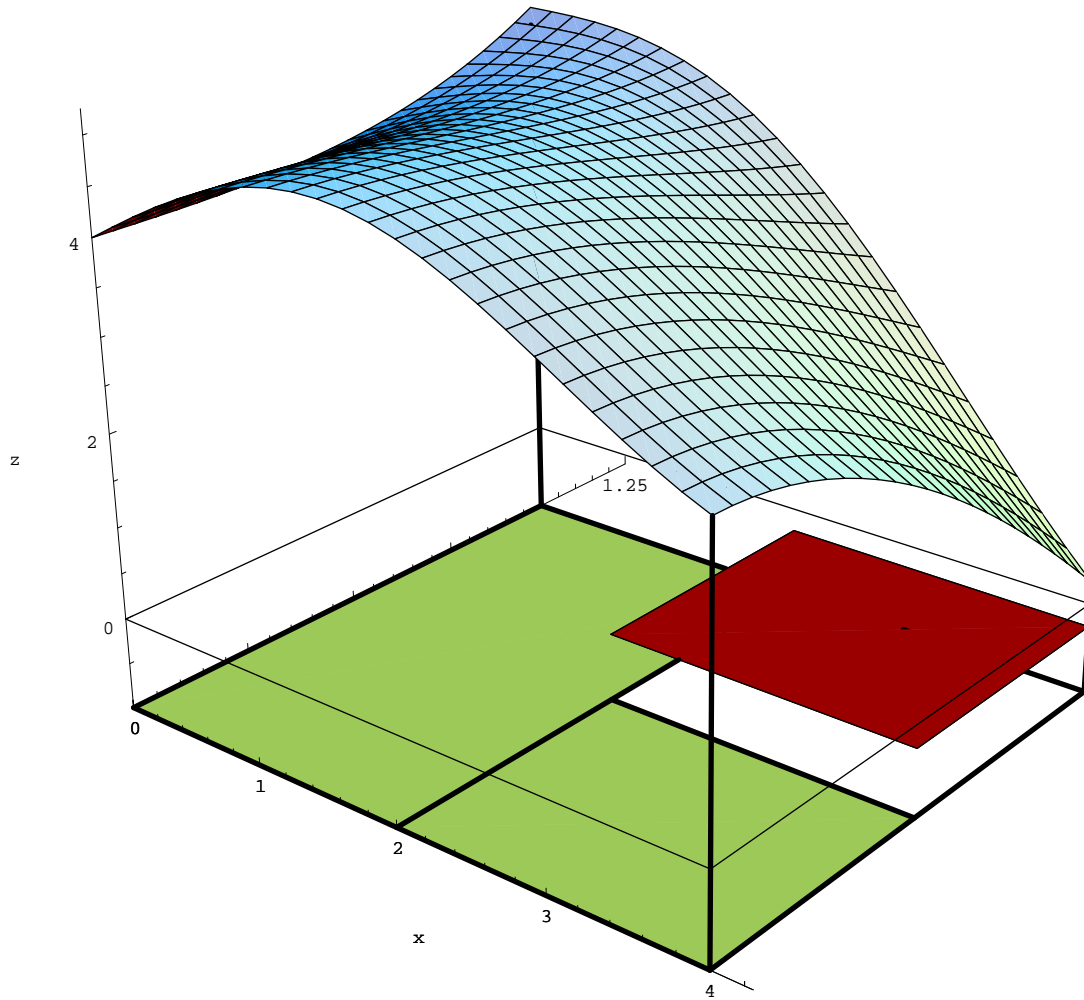
$$\sin x + y^2(y - x) + 4$$



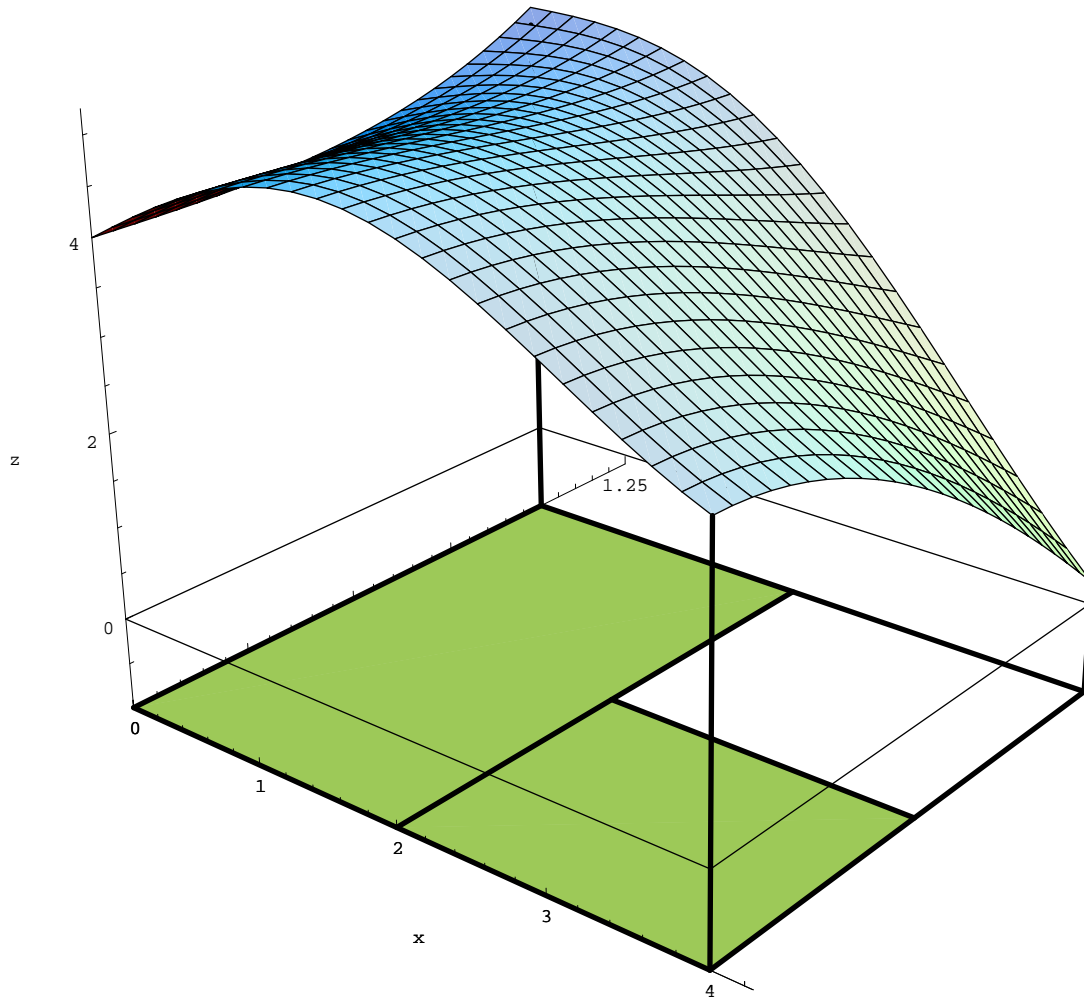
$$\sin x + y^2(y - x) + 4$$



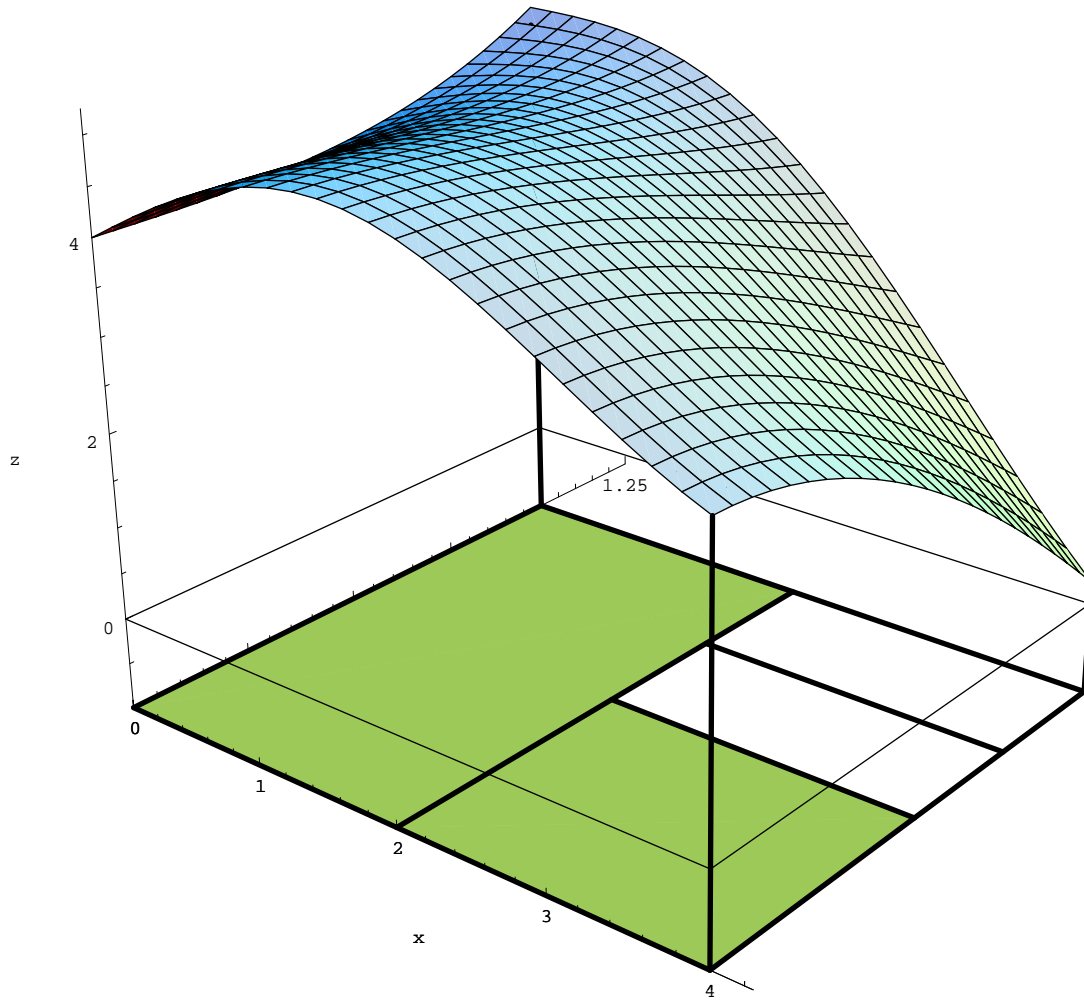
$$\sin x + y^2(y - x) + 4$$



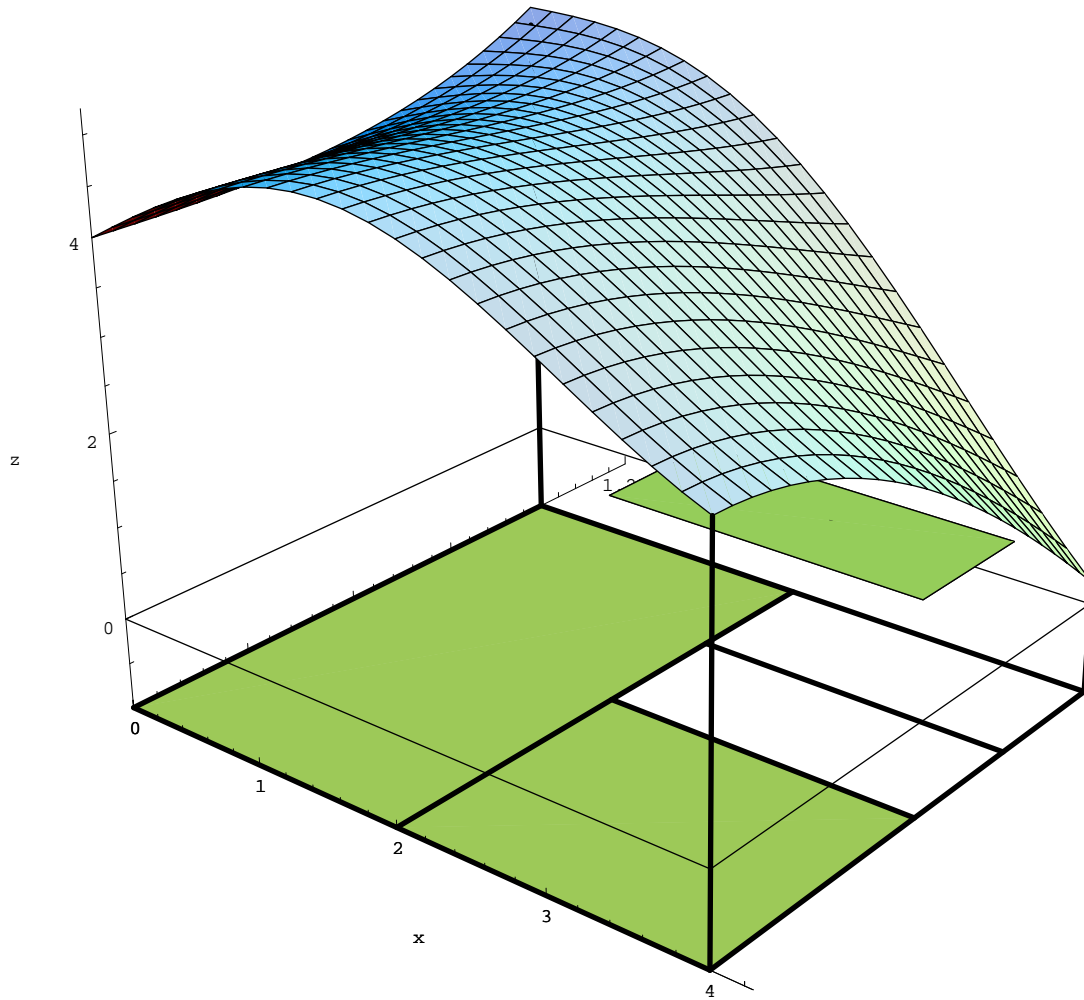
$$\sin x + y^2(y - x) + 4$$



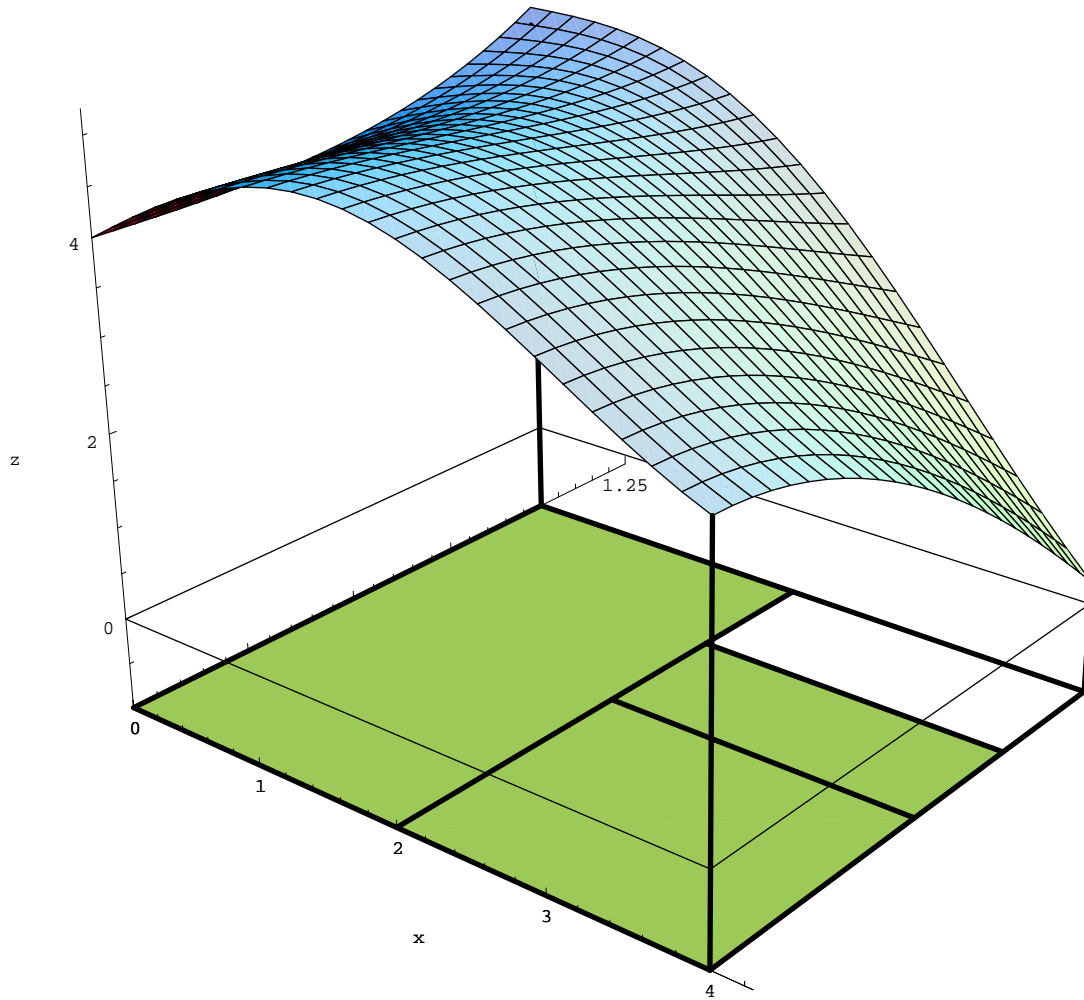
$$\sin x + y^2(y - x) + 4$$



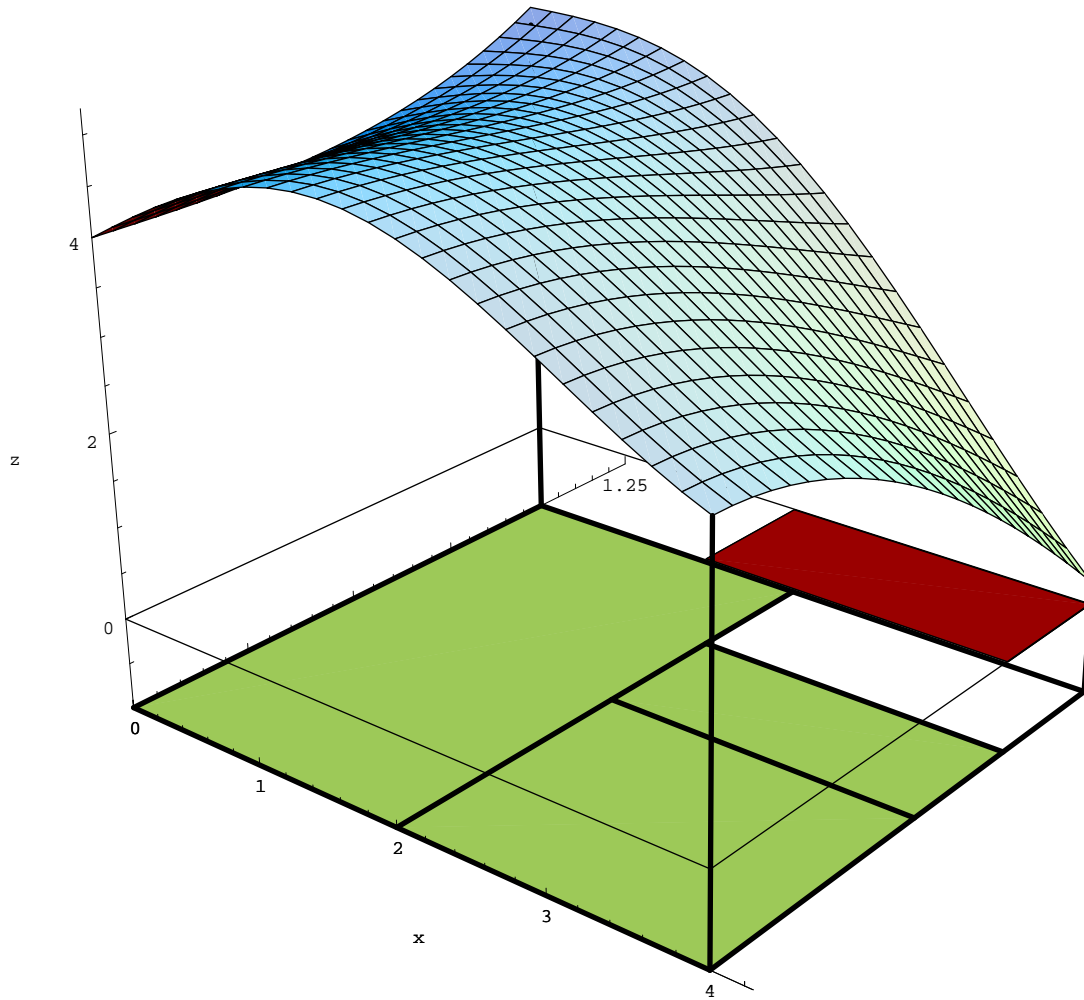
$$\sin x + y^2(y - x) + 4$$



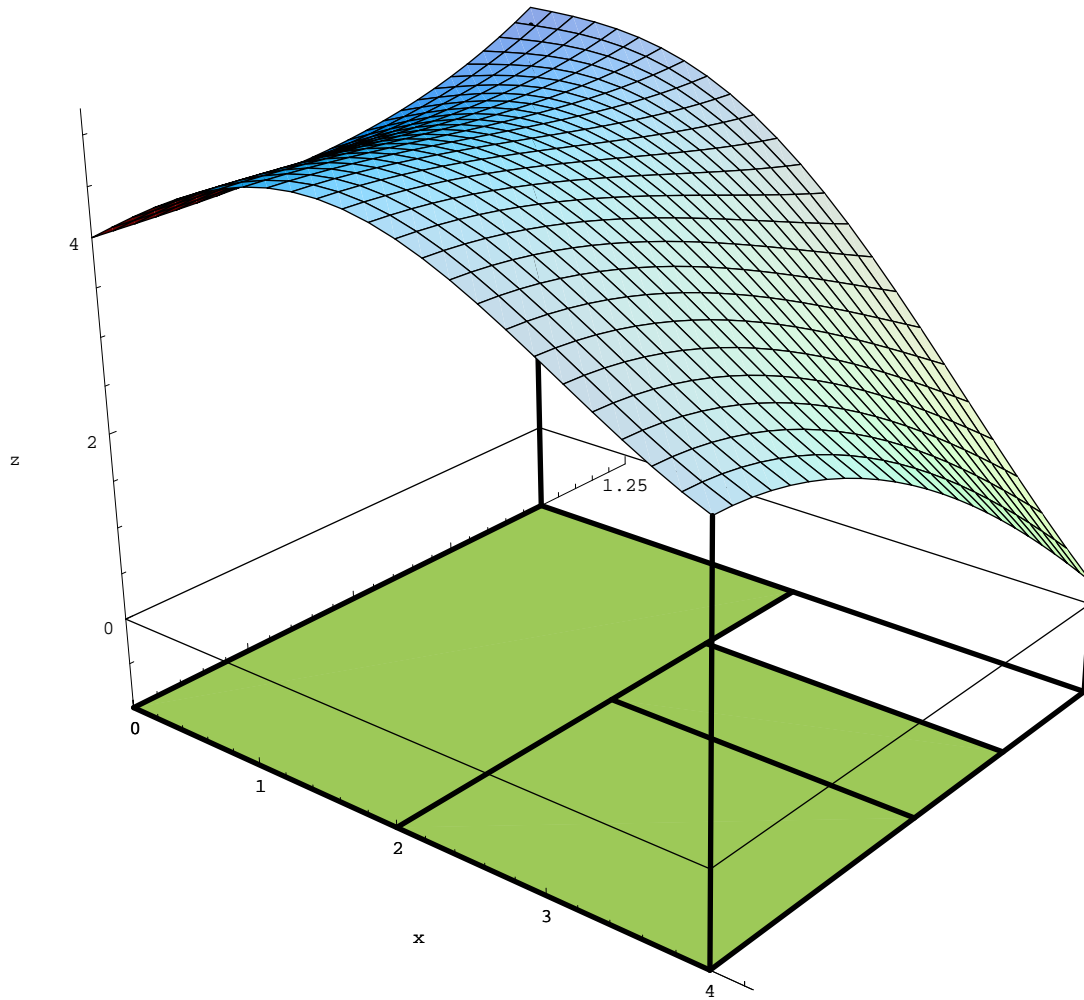
$$\sin x + y^2(y - x) + 4$$



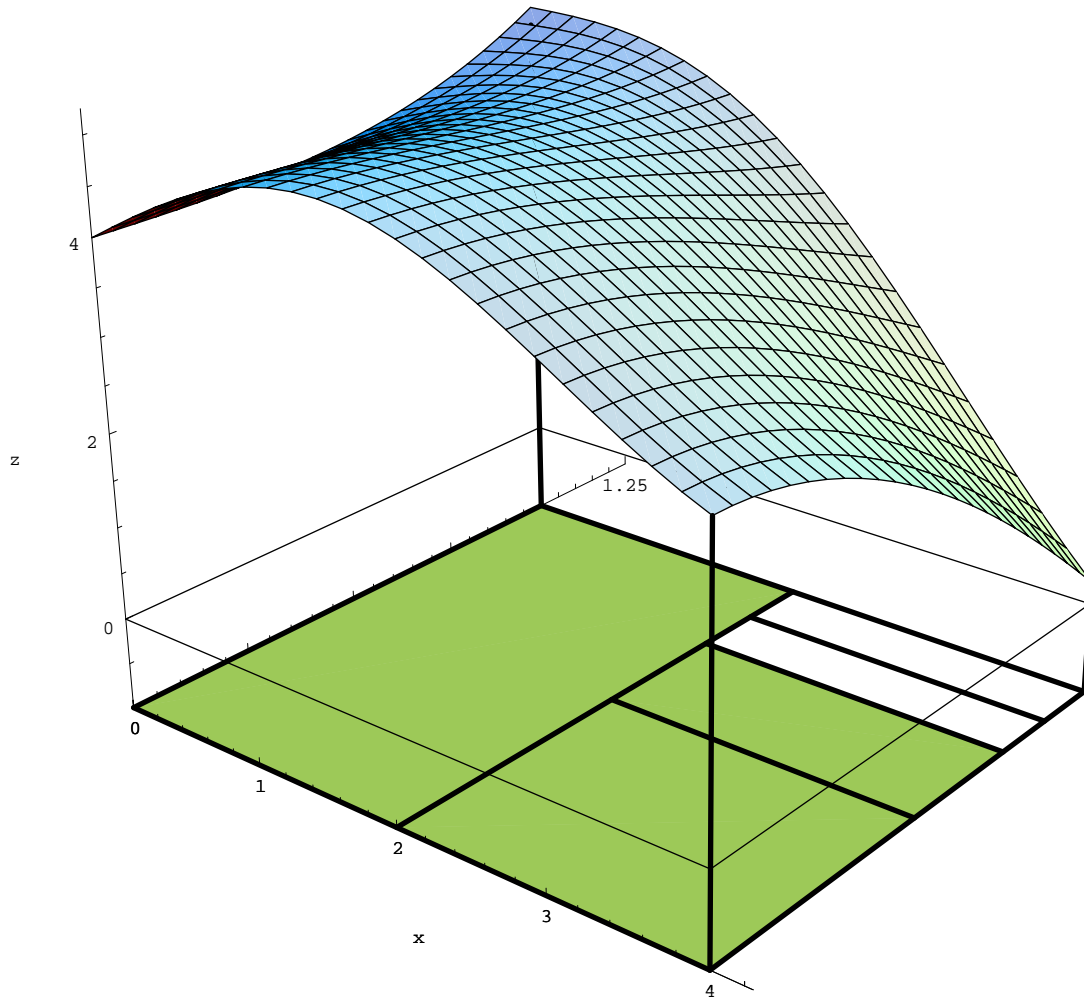
$$\sin x + y^2(y - x) + 4$$



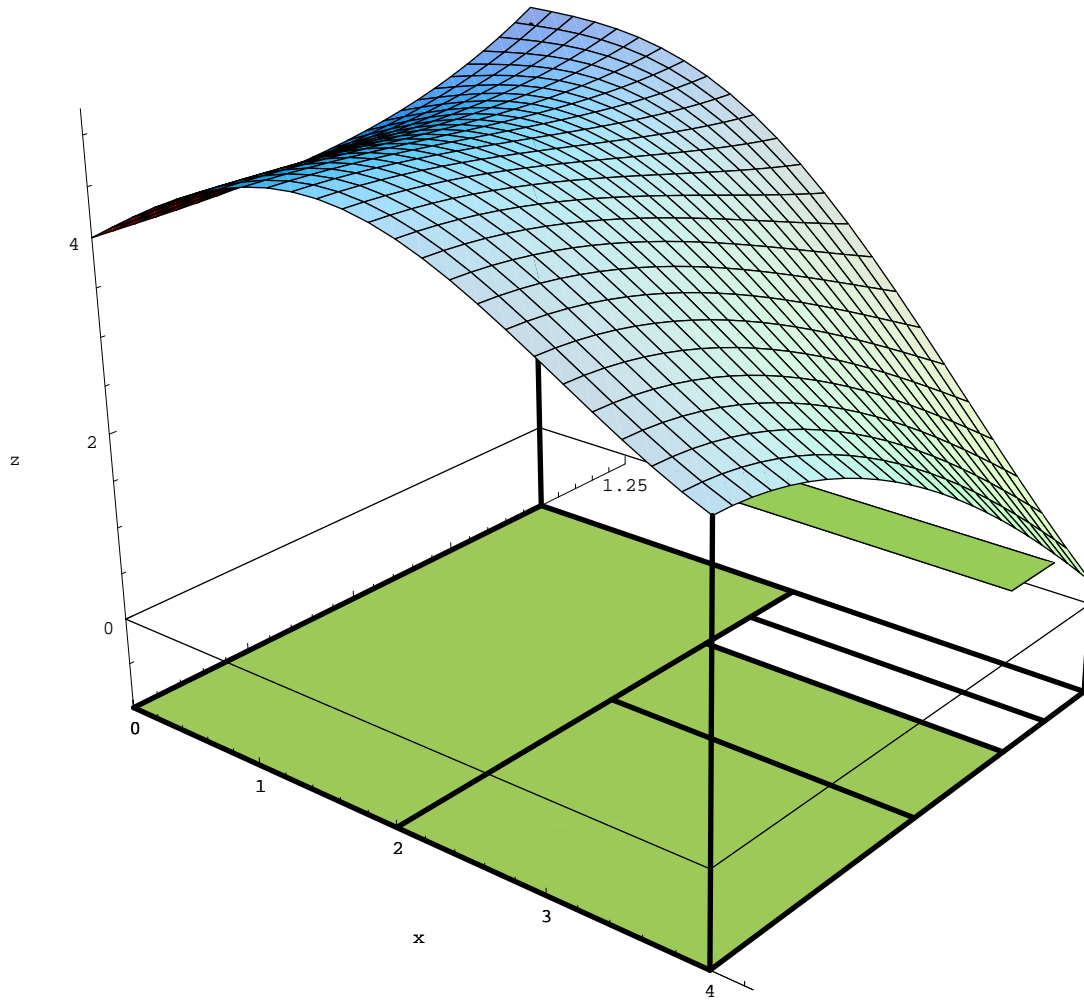
$$\sin x + y^2(y - x) + 4$$



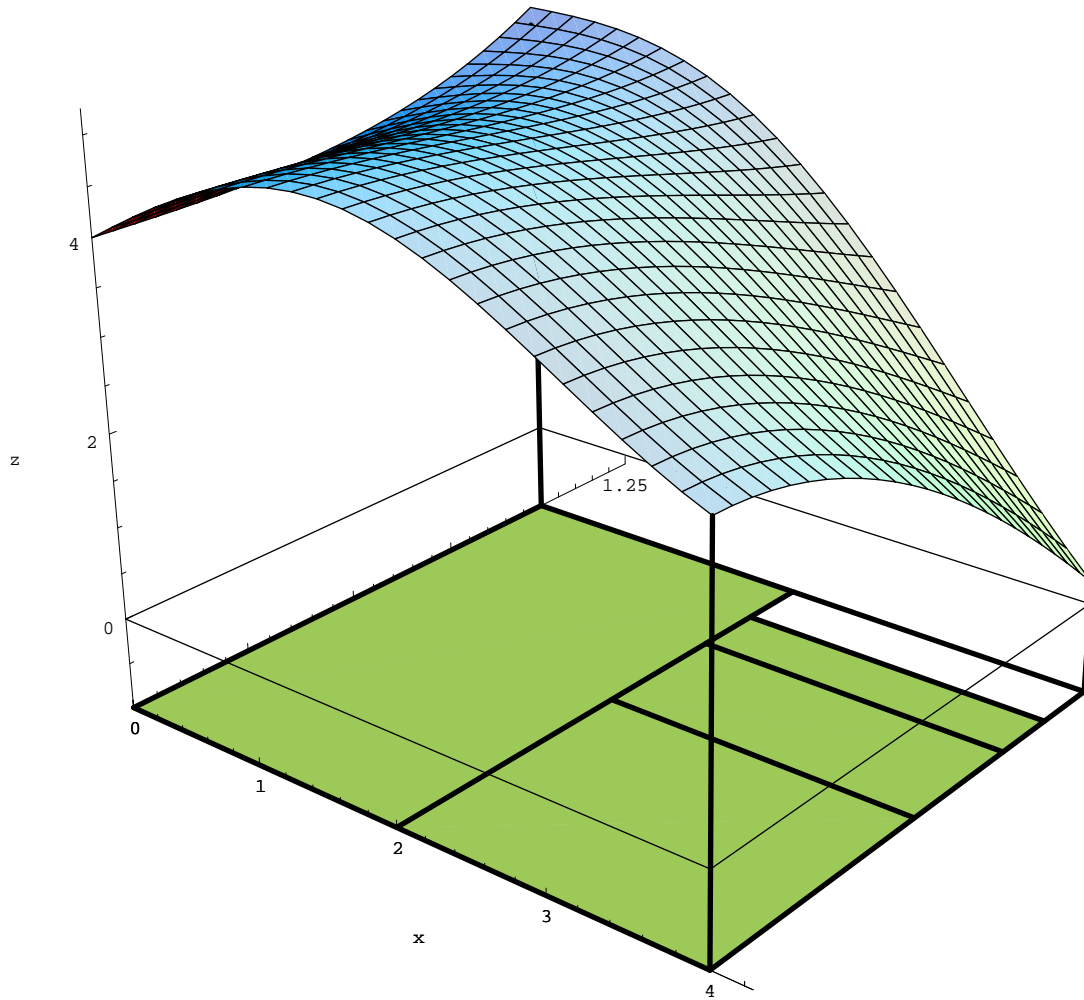
$$\sin x + y^2(y - x) + 4$$



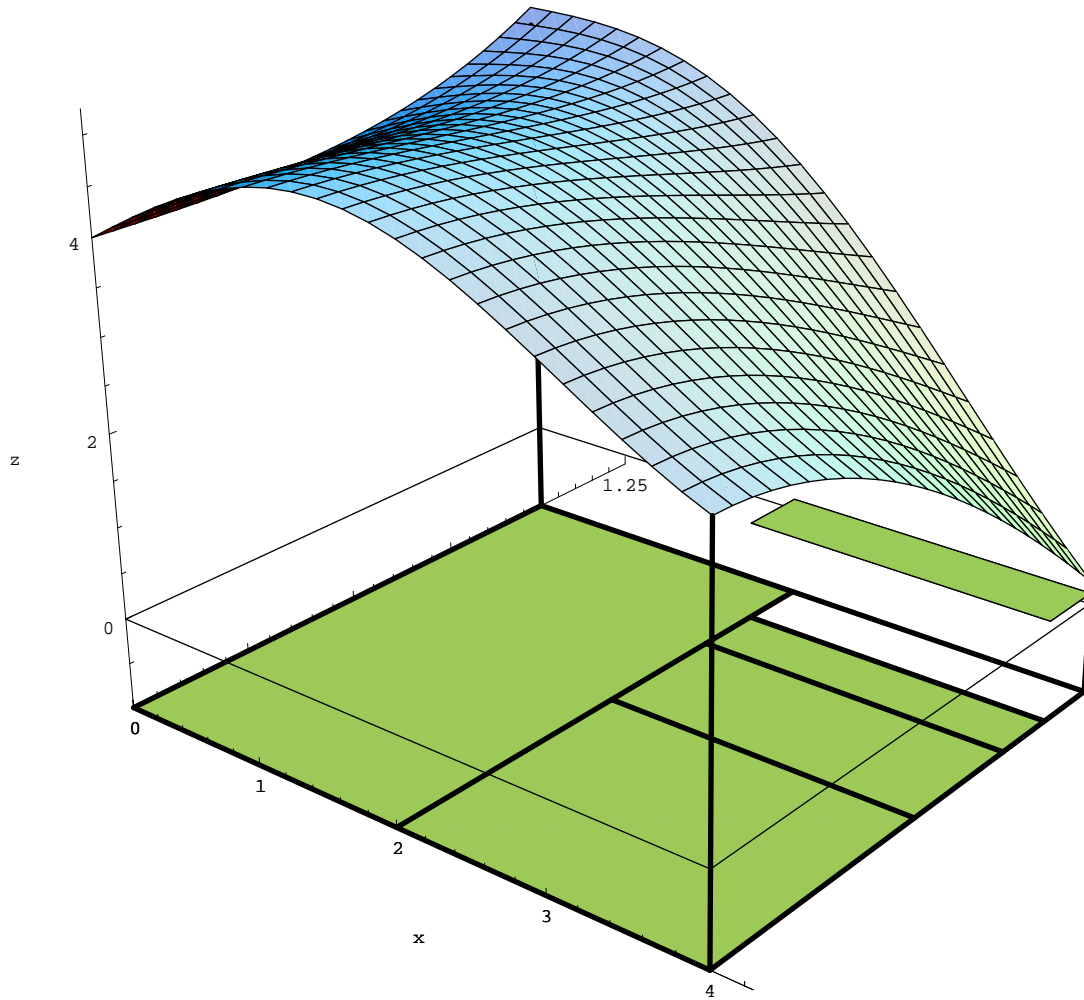
$$\sin x + y^2(y - x) + 4$$



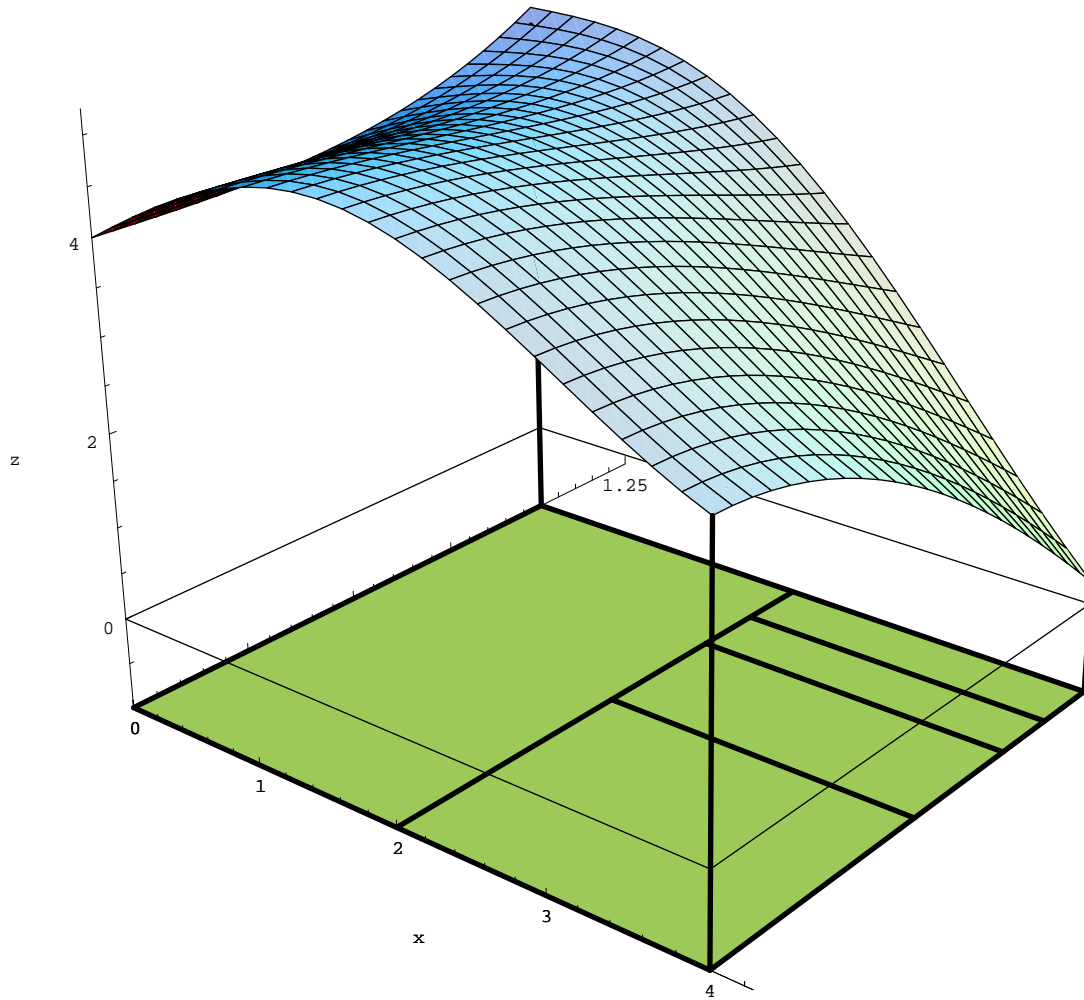
$$\sin x + y^2(y - x) + 4$$



$$\sin x + y^2(y - x) + 4$$



$$\sin x + y^2(y - x) + 4$$



$$\sin x + y^2(y - x) + 4$$

## Interval Bounds

- Most existing interval arithmetic software uses rationals, or floating-point numbers with directed rounding.
- This means that transcendent functions have to be approximated to a certain precision:  $\sqrt{(0, 2)} = (\sqrt{0}, \sqrt{2}) \subset (0, 1.42)$
- Then  $\sqrt{x} \leq 1.416$  cannot be deduced from  $x \in (0, 2)$  ! As a consequence the tactic would blindly continue to break up intervals.
- Constructive real numbers<sup>a</sup>  $\mathbb{R} = \mathbb{Z} \rightarrow \mathbb{Z}$  and for any real  $r$ :  
$$\forall z \in \mathbb{Z}. |4^z - r(z)| < 1$$
- They allow to dynamically increase precision:  
$$\sqrt{(0, 2)} = (\sqrt{0}, \sqrt{2}) = (0, 1.4\dots) = (0, 1.41\dots) = (0, 1.414\dots)$$
- Decidability of  $=$  and  $\leq$  is lost. No proofs for sharp irrational bounds.

---

<sup>a</sup>model: Valérie Ménéssier-Morain

## Semi-Decision Tactics in Type Theory

- The tactic is entirely written in Coq (reflection, no OCaml)
- Its type is  $t = \text{forall } p : I, \text{nat} \rightarrow \text{option } p$ 
  - $I$  is the type of inequalities
  - $p$  is a particular inequality
  - the natural number is the search depth
  - the returned value is ...
    - \*  $t \ p \ 5 = \text{None}$  ... nothing (tactic failure)
    - \*  $t \ p \ 50 = \text{Some } (\text{fun } x1 \ x2 \ \dots \Rightarrow \dots)$  ... or a proof of  $p$
- Variant: simultaneous search for counter-examples

## Prototype Architecture

- The Coq system is written in OCaml.
- Coq terms (modules) can be extracted to OCaml (Haskell, Scheme) terms (modules)
- Here:
  - Our tactic is a Coq functor (its argument: a number module).
  - The Creal library<sup>a</sup> is written in OCaml (no proofs of  $\mathbb{R}$ -axioms).
  - The tactic is extracted (into an OCaml-functor), then instantiated with the Creal library.
  - Result: a certified (semi-)decision procedure
  - On the Coq-level there have been experiments with  $\mathbb{Q}$ .
- Next: computational reals in Coq, compilation in Coq

---

<sup>a</sup>implementation: Jean-Christophe Filliâtre

## Conclusion

- Refinements can make interval arithmetic powerful.
- Its correctness appears to be relatively easy to prove.
- Its computation is relatively efficient. *Is it?*
- Constructive reals solve the precision problem.
- Drawback: no sharp irrational bounds (equality undecidable)
- Prototyping with mixture of Coq and OCaml is feasible.
- Open question: How can rewriting improve the evaluation scheme?
- work in progress . . .

**Thank You!**