НЕДООПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ

В начале 80-х гг. А.С.Нариньяни предложил идею нового аппарата представления знаний, который позволяет представлять частично определенные (или недоопределенные) понятия и оперировать ими.

Одной из разновидностей этого аппарата являются **Недоопределенные вычислительные модели**.

НЕДООПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ

Основная идея метода недоопределенных вычислительных моделей (н-моделей) заключается в том, что величинам (параметрам) решаемой задачи приписываются недоопределенные значения (н-значения), задающие некоторую область возможных значений (перечисление, интервал и т.п.).
Эти параметры связываются друг с другом ограничениями, которые представляются, как правило, в виде обычных математических и логических формул. Интерпретация ограничений, выполняемая по специальному алгоритму, позволяет уточнять значения связываемых ими параметров.

Таким образом, задав набор ограничений — вычислительную модель, можно получить в результате ее интерпретации более или менее точные значения искомых величин, удовлетворяющие наложенным ограничениям, или же обнаружить противоречие в этих ограничениях.

НЕДООПРЕДЕЛЕННЫЕ ТИПЫ ДАННЫХ

Введем понятие «**недоопределенный тип данных**» как расширение обычного типа данных.

Пусть ***T*** - это "обычный" тип данных с множеством значений ***A*** и соответствующим набором операций ***P*** над ***A***.

Обозначим через ***A**** множество всех подмножеств ***A***.

Элементы ***A**** будем называть ***недоопределенными значениями*** (н-значениями) и обозначать ***a****.

Значение ***a****, содержащее только один элемент из ***A*** будем называть ***точным значением***.

Значение ***a****, равное всему множеству ***A***, будем называть ***полной неопределенностью***.

Значение **a* = { }** – ***противоречивым*** или ***несовместным*** значением.

НЕДООПРЕДЕЛЕННЫЕ ТИПЫ ДАННЫХ

Для каждой операции $\boldsymbol{P : A^n \rightarrow A}$ типа $\boldsymbol{T}$ определим соответствующую операцию $\boldsymbol{P^* : A^{*n} \rightarrow A^*}$

как **недоопределенное расширение** операции $\boldsymbol{P}$:

$$\boldsymbol{P^* (a_1^*, ..., a_n^*) = \{ P (a_1, ..., a_n) \mid a_1 \in a_1^*, ..., a_n \in a_n^* \}.}$$

Семантика этих операций аналогична семантике традиционных операций, но они могут применяться к недоопределенным значениям и иметь в общем случае недоопределенный результат.
Таким образом на основе "точного" типа $\boldsymbol{T}$ мы можем построить недоопределенный тип $\boldsymbol{T^*}$.

В настоящее время н-расширения построены для различных типов данных: *целых, вещественных, символьных, логических* и др.

Метод недоопределенных вычислительных моделей

Недоопределенные типы данных могут использоваться для представления неточных данных в задачах, которые описываются в терминах ограничений на их параметры. Для решения таких задач используется метод недоопределенных вычислительных моделей (МНВ).

Формально ограничение - это булево выражение $C(v_1,...,v_n)$, которое должно быть истинным.

Переменные $v_1,...,v_n$, связанные с ограничением могут иметь любой недоопределенный тип.

Метод недоопределенных вычислительных моделей

Каждое ограничение *C* должно иметь функциональную интерпретацию, т.е. представляться множеством функций интерпретации:

$$f_i^* : A^{*(n-1)} \rightarrow A^*.$$

Если выражение в ограничении *C* очень сложное, то его можно упростить путем добавления вспомогательных переменных и расщепления на несколько простых выражений.

Метод недоопределенных вычислительных моделей

Например, ограничение

$$Z = X + Y \qquad (1)$$

может быть проинтерпретировано следующими тремя функциями:

$$Z = f1^*(X, Y),$$

$$X = f2^*(Z, Y),$$

$$Y = f3^*(Z, X),$$

где $f_1^*(x, y) = x + y$, $f_2^*(z, y) = z - y$, $f_3^*(z, y) = z - x$.

Здесь «-» и «+» обозначают недоопределенное расширение арифметических операций вычитания и сложения, соответственно.

Метод недоопределенных вычислительных моделей

Недоопределенная модель описывается четверкой

$$M = (X, R, W, C),$$

где X – **множество объектов** (переменных) из заданной ПО,

R – **множество функционально интерпретируемых отношений** на объектах из X,

W – **множество функций присваивания**, т.е. двуместных операторов вида: $w_i : A^* \times A^* \rightarrow A^*$,

результат выполнения этой операции получается как пересечение нового значения переменной со старым.

Примеры: $[1, 4] \cap [2,5] = [2, 4]$ или $\{1,2,3,4\} \cap \{2,3,4,5\} = \{2,3,4\}$

Функция присваивания срабатывает при каждой попытке присваивания переменной нового значения.

C – **множество функций проверки корректности**.

Метод недоопределенных вычислительных моделей

Функции проверки корректности следят за правильностью изменения значений переменных x_i и сообщают, если есть попытка присвоить переменной некорректное значение.

Например

- при перечислимом типе - пустого множества *{}*,

- при интервальном типе *[НГР, ВГР]*, *НГР > ВГР*.

Метод недоопределенных вычислительных моделей

Н-модель можно представить двудольным ориентированным графом (Н-сетью) в комплексе с некоторой дисциплиной его исполнения: *потоковым управлением вычислениями*.

В Н-сети выделены два типа вершин: *объекты* и *функции*. Дуги связывают функциональные и объектные вершины.

Входящие в вершину-функцию дуги соотносят с ней объекты, значения которых выступают в качестве входных аргументов для функции.

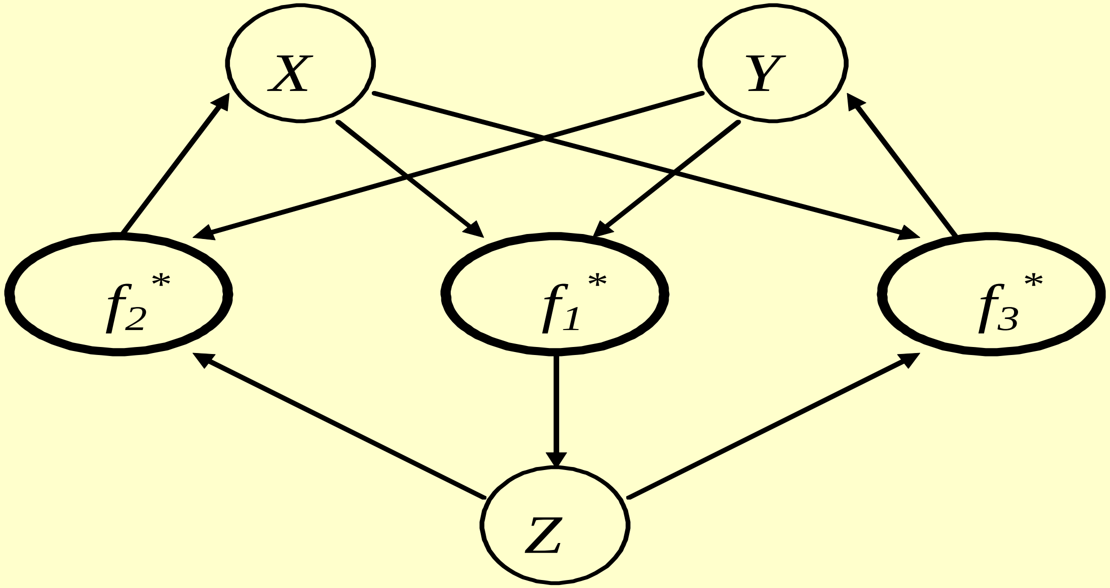
Исходящие дуги указывают на объекты, в которые должна производиться запись вырабатываемых функцией результатов.

Метод недоопределенных вычислительных моделей

Н-модель, соответствующая ограничению

$$Z = X + Y, \qquad (1)$$

имеет вид



Метод недоопределенных вычислительных моделей

Каждой *объектной вершине* сопоставляются:

тип, значение, функция присваивания w_i и функция проверки корректности c_i.

С каждой *функциональной вершиной* соотнесены:

процедура, ее вычисляющая; разметка входящих и исходящих дуг; возможно, приоритет (целое число).

Принцип потокового управления вычислениями заключается в том, что изменение объектных вершин сети активирует (вызывает к исполнению) функциональные вершины, для которых эти объектные вершины являются входными аргументами, а исполнение функциональных вершин, в свою очередь, может вызывать изменение результирующих объектных вершин.

Метод недоопределенных вычислительных моделей

Исполнение Н-сети осуществляется так называемым виртуальным потоковым процессором и описывается следующим образом.

1. Создается очередь функциональных вершин (в соответствии с приоритетами, если они заданы) - список активных функциональных вершин сети (вначале все функциональные вершины считаются активными).

2. На каждом шаге исполнения Н-сети из очереди выбирается функциональная вершина с наивысшим приоритетом. После своего исполнения вершина-функция переходит в пассивное состояние.

3. Значения, полученные в результате исполнения функции, записываются в соответствующие вершины-объекты с помощью функций присваивания.

Метод недоопределенных вычислительных моделей

4. Если после исполнения функции присваивания объект изменил свое значение, то вызывается связанная с ним функция проверки корректности. Если получено некорректное значение происходит останов исполнения сети и выдается сообщение об ошибке. В этом случае н-модель будет рассматриваться как несовместная.

5. Активируются все функциональные вершины, для которых измененные объекты являются входными, причем если какая-либо вершина уже находится в активном состоянии, то ее повторная активация не происходит.

6. Исполнение Н-сети заканчивается, когда очередь активных вершин становится пустой.

Метод недоопределенных вычислительных моделей

Важной особенностью н-модели является то, что в **переменную-аргумент** записывается не вычисленное значение функции, а **результат его пересечения** со старым значением переменной. В следствие этого значение переменной может **только уточняться**. При этом значение переменной считается изменившимся, только в том случае, когда оно действительно уточнилось. Благодаря этому для всех типов данных, содержащих конечное множество н-значений, этот алгоритм заканчивается за конечное число шагов.

В случае бесконечного (неконечного) множества н-значений (например, интервалов вещественных чисел) критерий останова базируется на точности вычислений ε. Когда новое значение переменной отличается от старого на величину меньше ε, оно не присваивается переменной и процесс вычислений останавливается.

Пример работы алгоритма удовлетворения ограничений

Некоторые сведения из области интервальной математики

Значением переменной *x* является интервал, задаваемый нижней *Lo* и верхней *Up* границей, т.е $x = [x^{Lo}, x^{Up}]$.

Определим арифметические операции над интервалами:

$z = x \circ y$ ($\circ$ - любая арифметическая операция)

$z = [z^{Lo}, z^{Up}]$.

$z^{Lo} = \min (x^{Lo} \circ y^{Lo}, x^{Lo} \circ y^{Up}, x^{Up} \circ y^{Lo}, x^{Up} \circ y^{Up})$.

$z^{Up} = \max (x^{Lo} \circ y^{Lo}, x^{Lo} \circ y^{Up}, x^{Up} \circ y^{Lo}, x^{Up} \circ y^{Up})$.

Например, для $z = x + y$ :: $z^{Lo} = x^{Lo} + y^{Lo}$; $z^{Up} = x^{Up} + y^{Up}$.

Для $z = x - y$ :: $z^{Lo} = x^{Lo} - y^{Up}$; $z^{Up} = x^{Up} - y^{Lo}$.

Пример работы алгоритма удовлетворения ограничений

Поясним работу алгоритма удовлетворения ограничений для Н-моделей на примере достаточно простой системы двух линейных уравнений с двумя целочисленными неизвестными (переменными):

$$x + y = 12;$$
$$2 * x = y; \qquad (2)$$

Предположим, что значения x и y ограничены следующими неравенствами:

$$0 \leq x \leq 100;\ 0 \leq y \leq 100. \qquad (3)$$

Множество объектов X данной модели содержит две целочисленные переменные (x, y), множество ограничений R — два уравнения и четыре неравенства.

Пример работы алгоритма удовлетворения ограничений

Множество функций интерпретации уравнений (2) состоит из четырех элементов ($f_1 - f_4$):

f_1: $y \leftarrow 12 - x$; f_2: $x \leftarrow 12 - y$;

f_3: $y \leftarrow 2 * x$; f_4: $x \leftarrow y / 2$;

Согласно правилам интервальной арифметики, семантика функций интерпретации представляется следующим образом:

$$f_1 : [y^{Lo}, y^{Up}] \leftarrow [12 - x^{Up}, 12 - x^{Lo}];$$

$$f_2 : [x^{Lo}, x^{Up}] \leftarrow [12 - y^{Up}, 12 - y^{Lo}];$$

$$f_3 : [y^{Lo}, y^{Up}] \leftarrow [\min \{2 * x^{Lo}, 2 * x^{Up} \}, \max \{ 2 * x^{Lo}, 2 * x^{Up} \}];$$

$$f_4 : [x^{Lo}, x^{Up}] \leftarrow [\min \{ y^{Lo} / 2, y^{Up} / 2 \}, \max \{ y^{Lo} / 2, y^{Up} / 2 \}];$$

Пример работы алгоритма удовлетворения ограничений

Четыре ограничения (3) можно проинтерпретировать на этапе генерации Н-сети, и тогда нижние и верхние границы *x* и *y* станут равными *0* и *100* соответственно. В результате, на первом шаге процесса потоковых вычислений имеем:

$$x = [0,100]; \qquad y = [0,100];$$

Покажем ход процесса вычислений на заданной Н-сети.

На первом шаге итерации исполняется функция f_1:
В результате $y = 12 - x = 12 - [0,100] = [-88, 12]$.
После этого вызывается функция присваивания:
$y = [0, 100] \cap [-88, 12] = [0, 12]$.

Условие корректности для *y* не нарушается, поэтому процесс вычислений продолжается. Активируются f_2 и f_4, для которых *y* является входным аргументом. Ввиду того что эти функции уже входят в очередь, их повторная активация не происходит.

Пример работы алгоритма удовлетворения ограничений

Далее исполняется следующая функция из очереди:

f_2: $x = 12 - y = 12 - [0, 12] = [0, 12]$

После функции присваивания: $x = [0, 100] \cap [0, 12] = [0, 12]$.

f_3: $y = 2 * x = 2 * [0, 12] = [0, 24]$

После присваивания: $y = [0, 12] \cap [0, 24] = [0, 12]$.

Как видим значение y не изменилось.

f_4: $x = y / 2 = [0, 12] / 2 = [0, 6]$

$x = [0, 12] \cap [0, 6] = [0, 6]$.

И так далее.

Пример работы алгоритма удовлетворения ограничений

Вычисления заканчиваются тогда, когда нижняя и верхняя границы как x, так и y становятся равными друг другу.

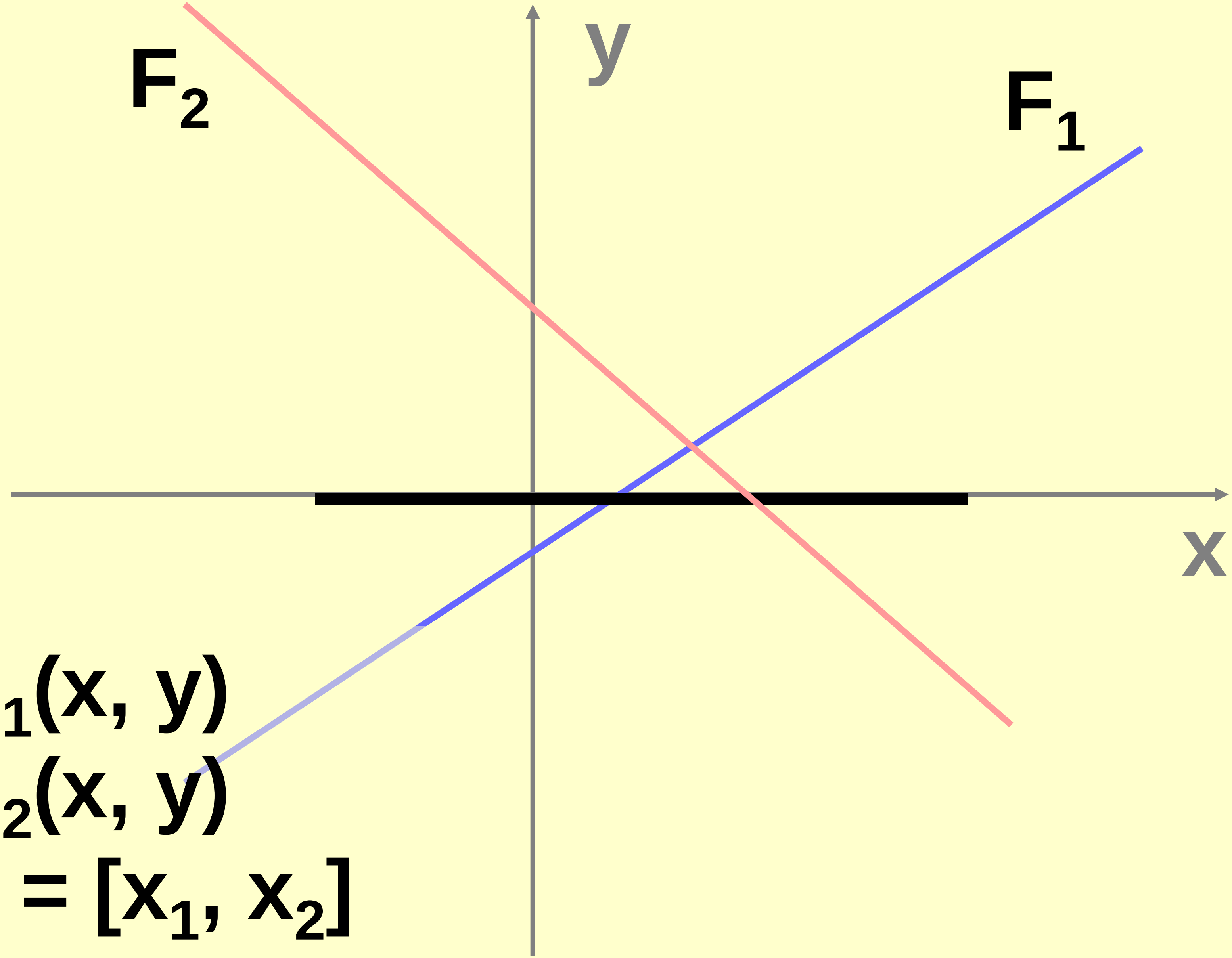
Это произойдет при значениях x^{Lo} и x^{Up}, равных 8 , и y^{Lo} и y^{Up} равных 4. При таких значениях исполнение любой функции $f_1 - f_4$ не изменяет значения своего результата, и множество активных функций становится пустым.

На следующем слайде приведена таблица, описывающая последовательность шагов исполнения алгоритма. В таблице предполагается, что на каждом шаге исполняется первая функция из множества активных функций (она отделена от остальных символом ‘|’)

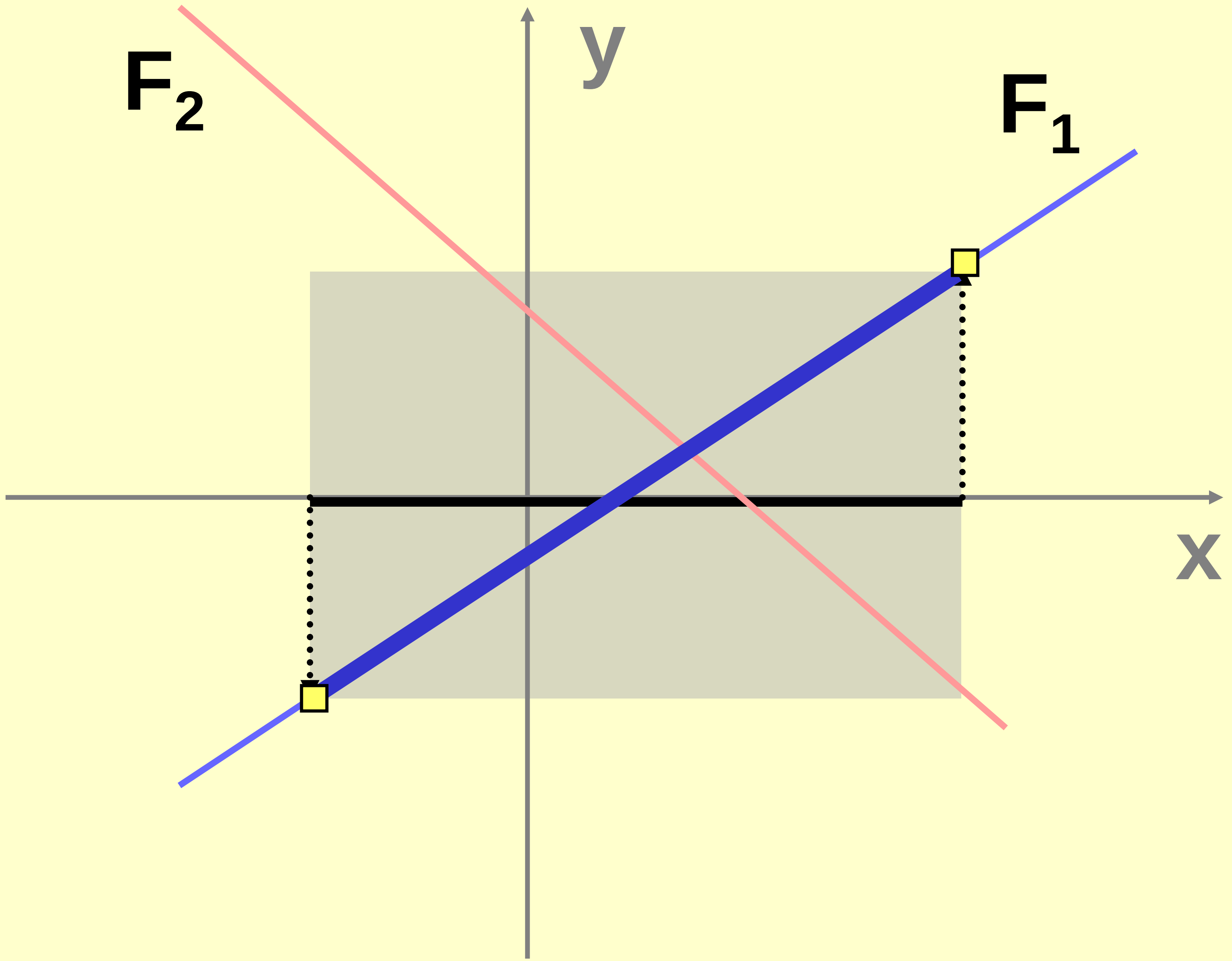
N	Активные функции	Н-значения текущее | новое	Флаг	Добавить функции
1	$f_1 \mid f_2 - f_4$	$y = [0, 100] \mid [0, 12]$	да	$f_2,\ f_4$
2	$f_2 \mid f_3,\ f_4$	$x = [0, 100] \mid [0, 12]$	да	$f_1,\ f_3$
3	$f_3 \mid f_4,\ f_1$	$y = [0, 12] \mid [0, 12]$	нет	
4	$f_4 \mid f_1$	$x = [0, 12] \mid [0, 6]$	да	f_1, f_3
5	$f_1 \mid f_3$	$y = [0, 12] \mid [6, 12]$	да	$f_2,\ f_4$
6	$f_3 \mid f_2,\ f_4$	$y = [6, 12] \mid [6, 12]$	нет	
7	$f_2 \mid f_4$	$x = [0, 6] \mid [0, 6]$	нет	
8	$f_4 \mid$	$x = [0, 6] \mid [3, 6]$	да	$f_1,\ f_3$
9	$f_1 \mid f_3$	$y = [6, 12] \mid [6, 9]$	да	$f_2,\ f_4$
10	$f_3 \mid f_2,\ f_4$	$y = [6, 9] \mid [6, 9]$	нет	
11	$f_2 \mid f_4$	$x = [3, 6] \mid [3, 6]$	нет	
12	$f_4 \mid$	$x = [3, 6] \mid [3, 4]$	да	$f_1,\ f_3$
13	$f_1 \mid f_3$	$y = [6, 9] \mid [8, 9]$	да	$f_2,\ f_4$
14	$f_3 \mid f_2,\ f_4$	$y = [8, 9] \mid [8, 8]$	да	$f_2,\ f_4$
15	$f_2 \mid f_4$	$x = [3, 4] \mid [4, 4]$	да	$f_1,\ f_3$

Графическая иллюстрация вычислительного процесса

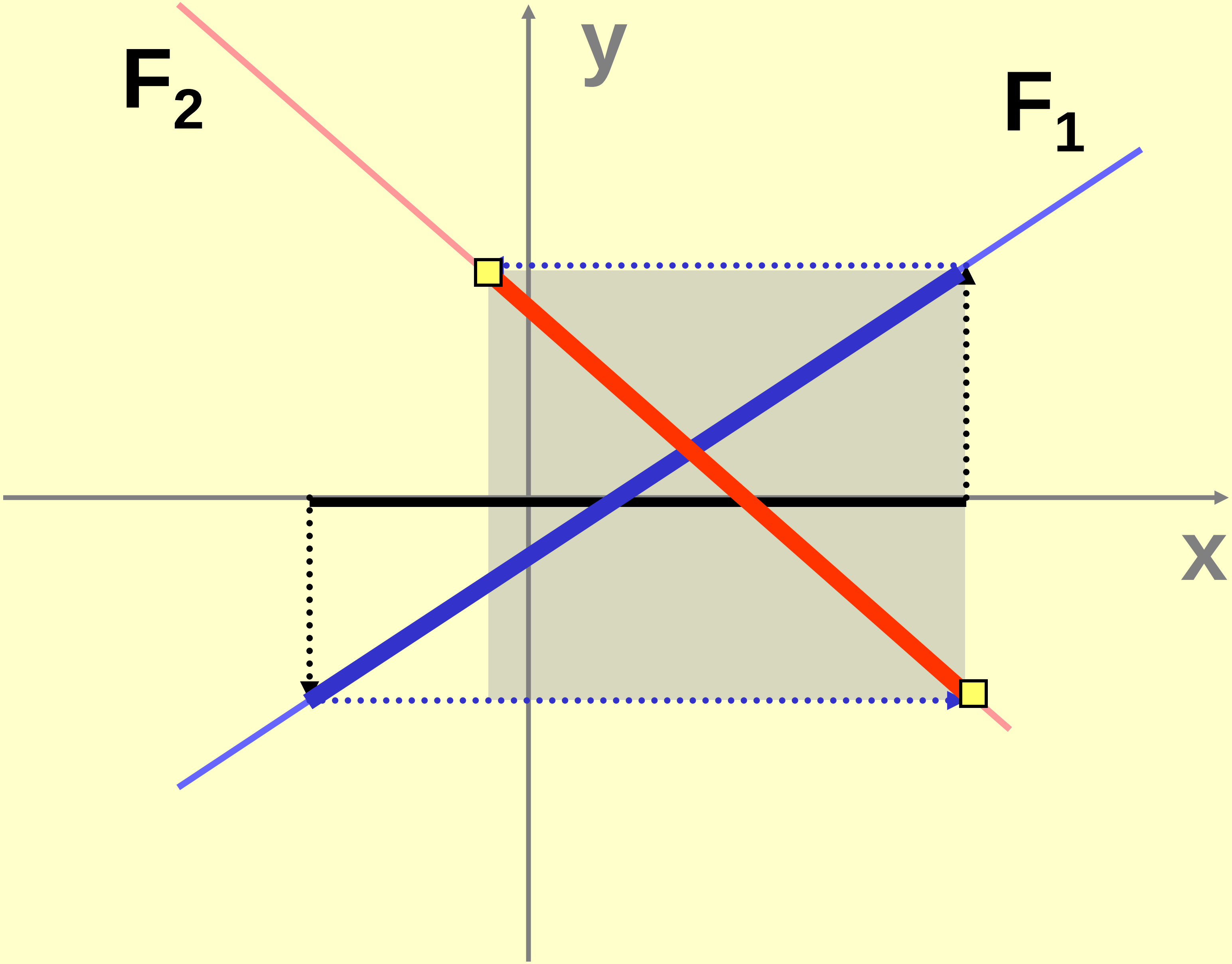
$$\begin{cases} F_1(x, y) \\ F_2(x, y) \\ x = [x_1, x_2] \end{cases}$$

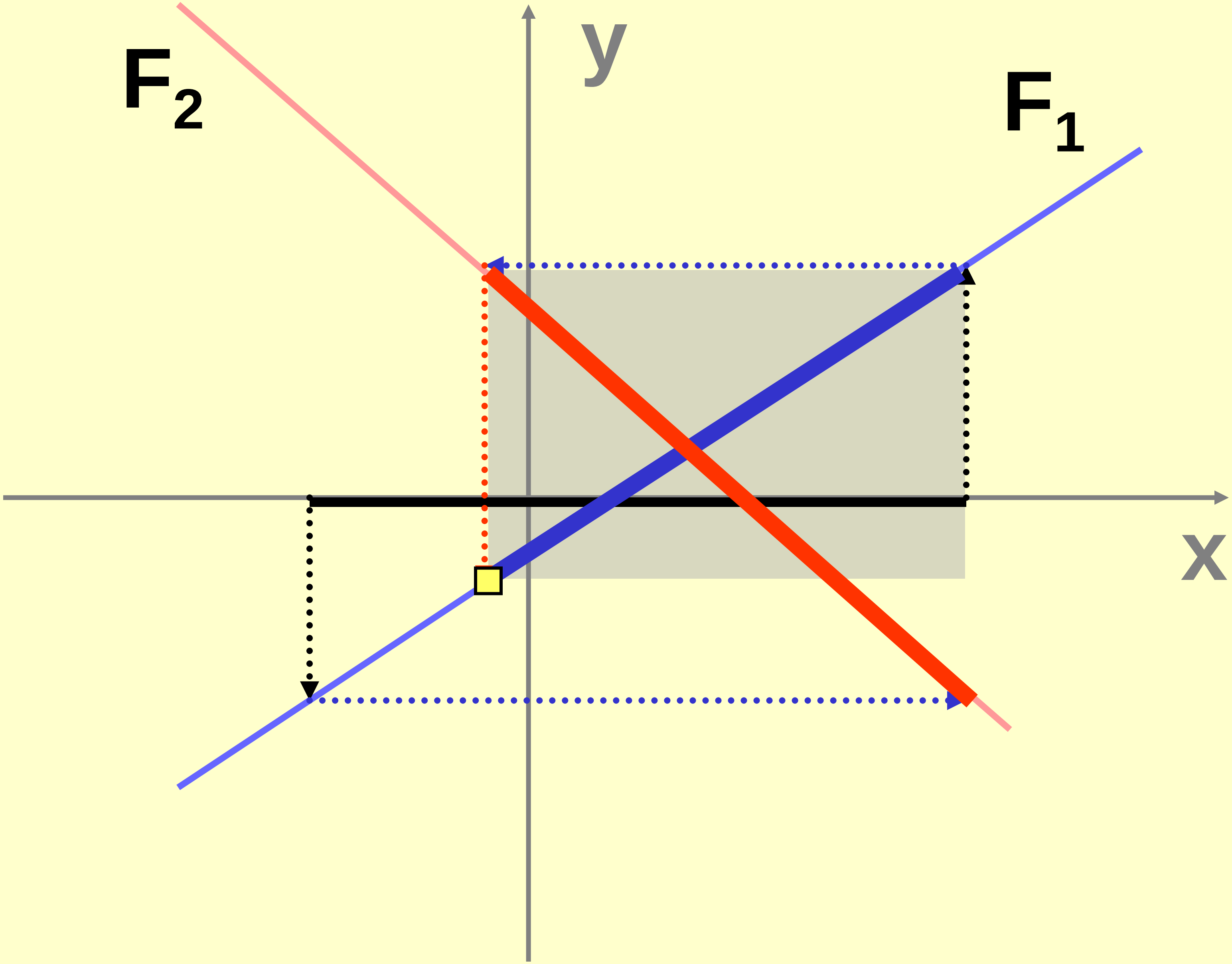


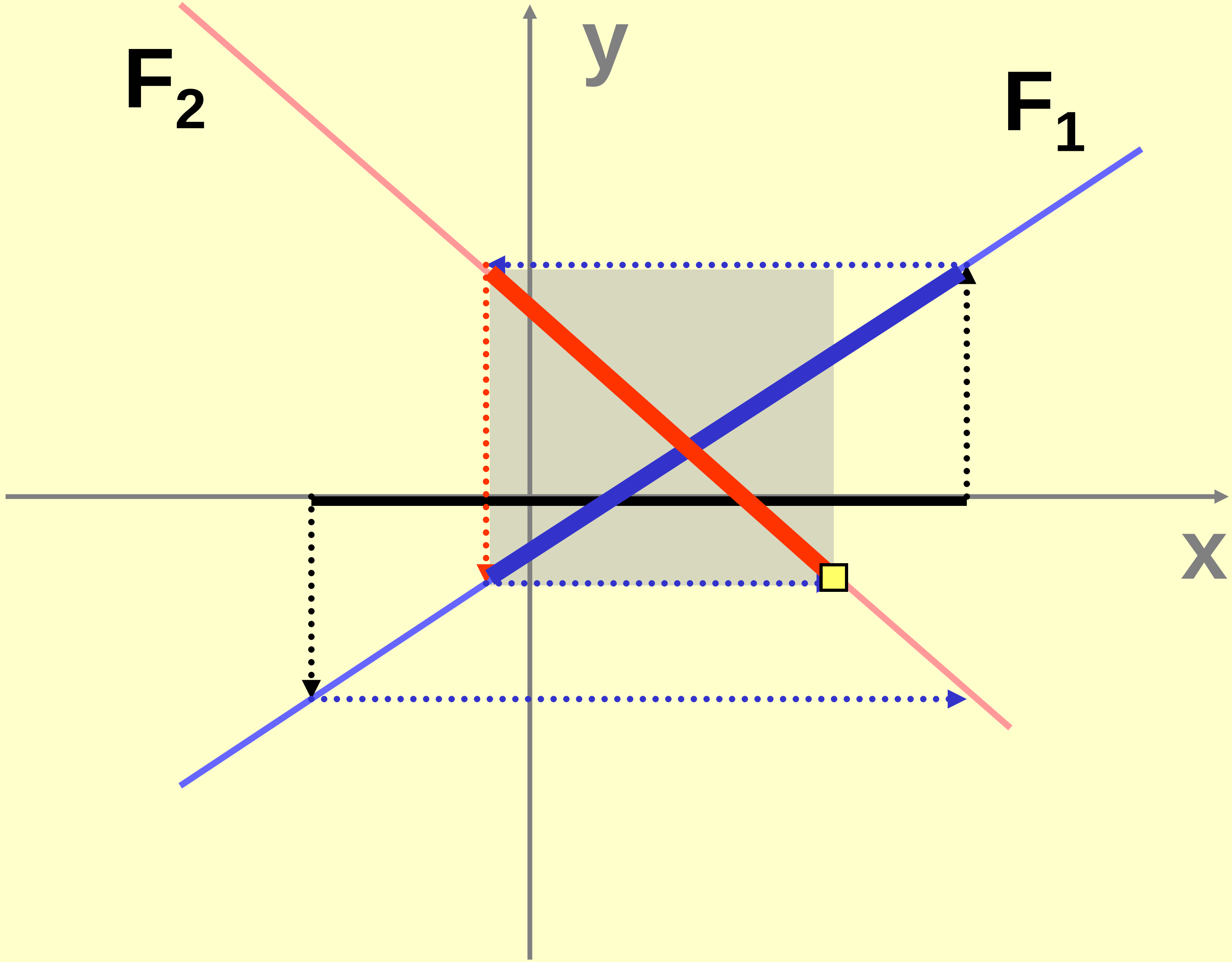
$F_1(x, y)$
$F_2(x, y)$
$x = [x_1, x_2]$



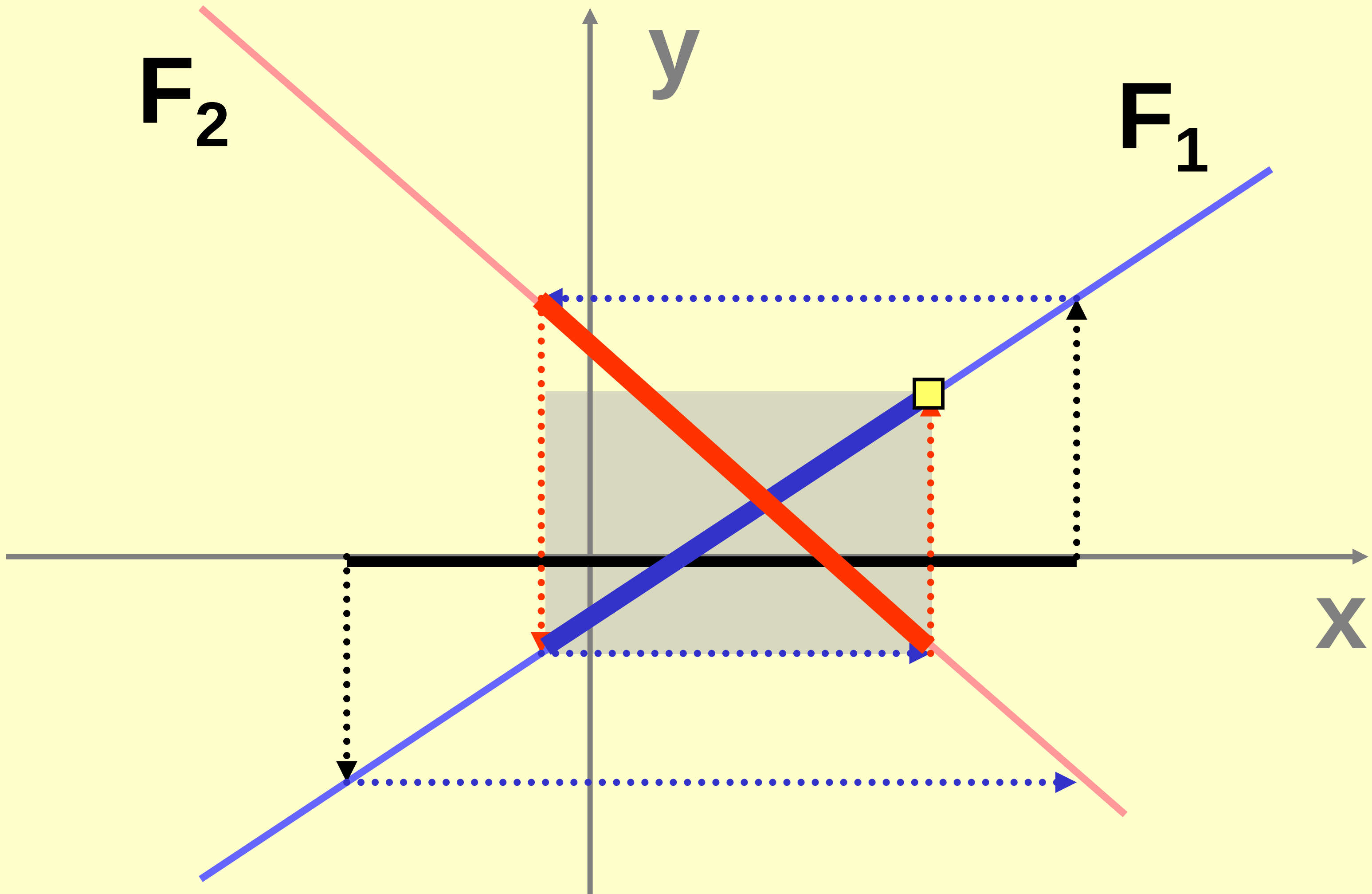
y
F_2
F_1
x


F2
y
F1
x

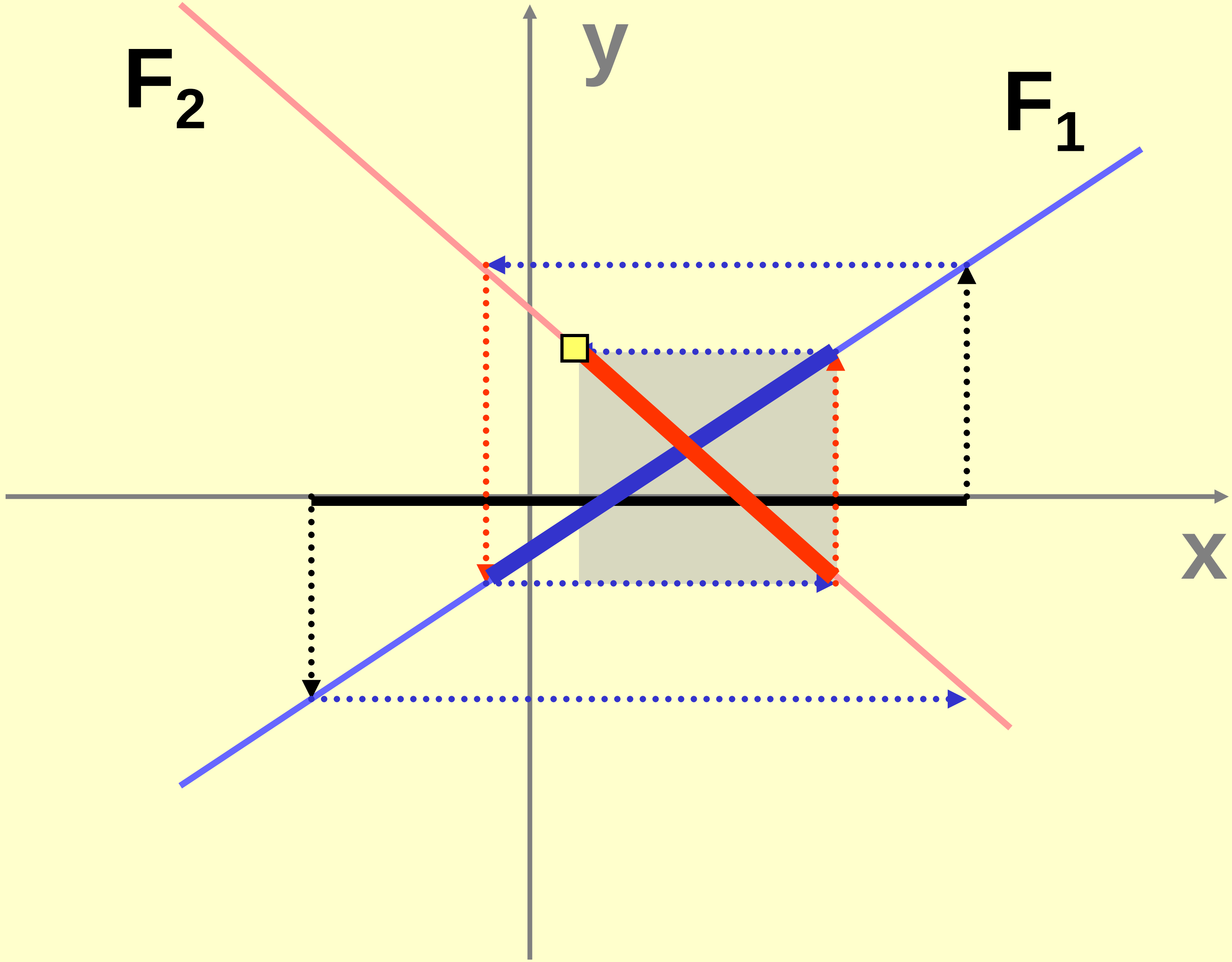

F2
y
F1
x



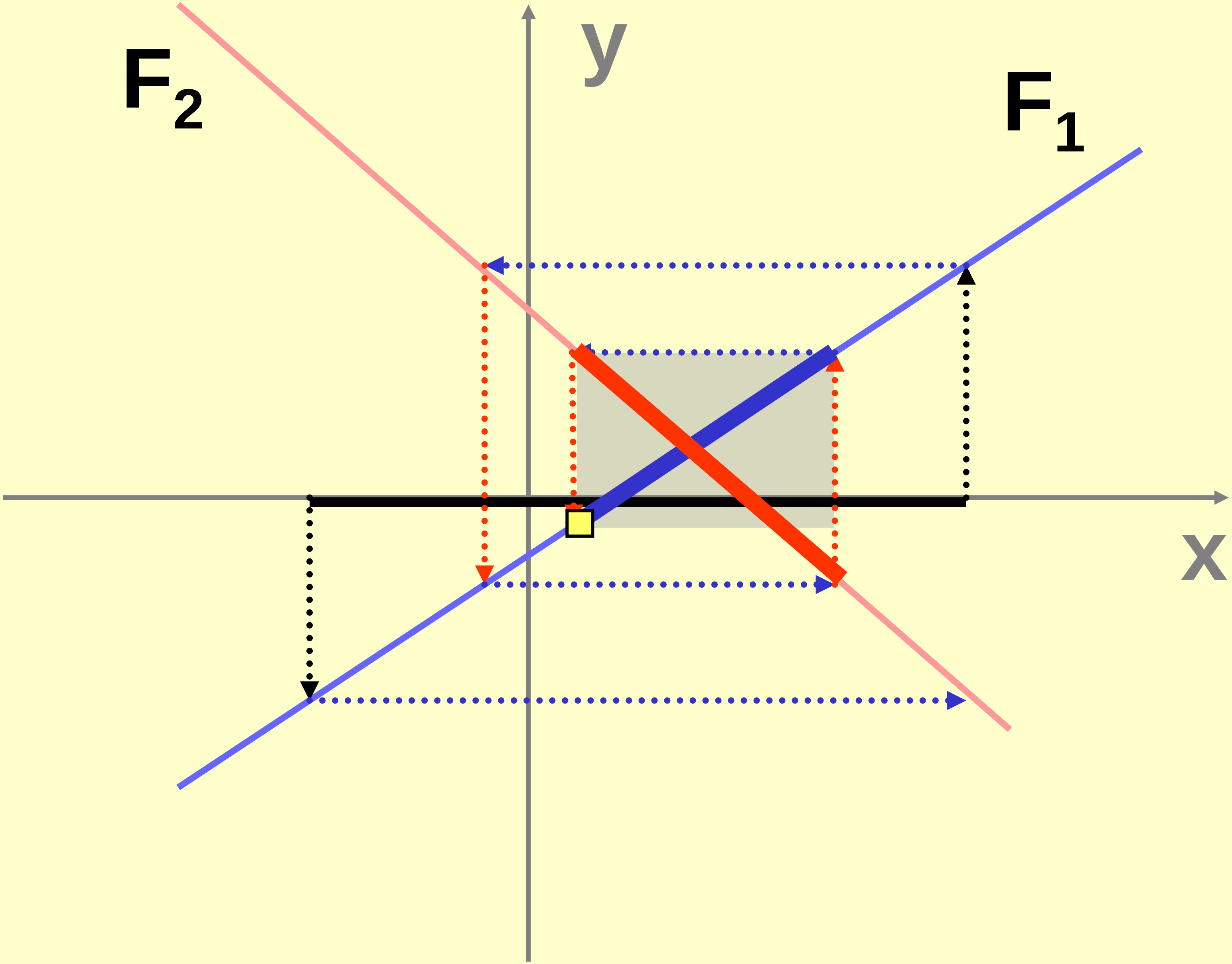
y
F_2
F_1
x

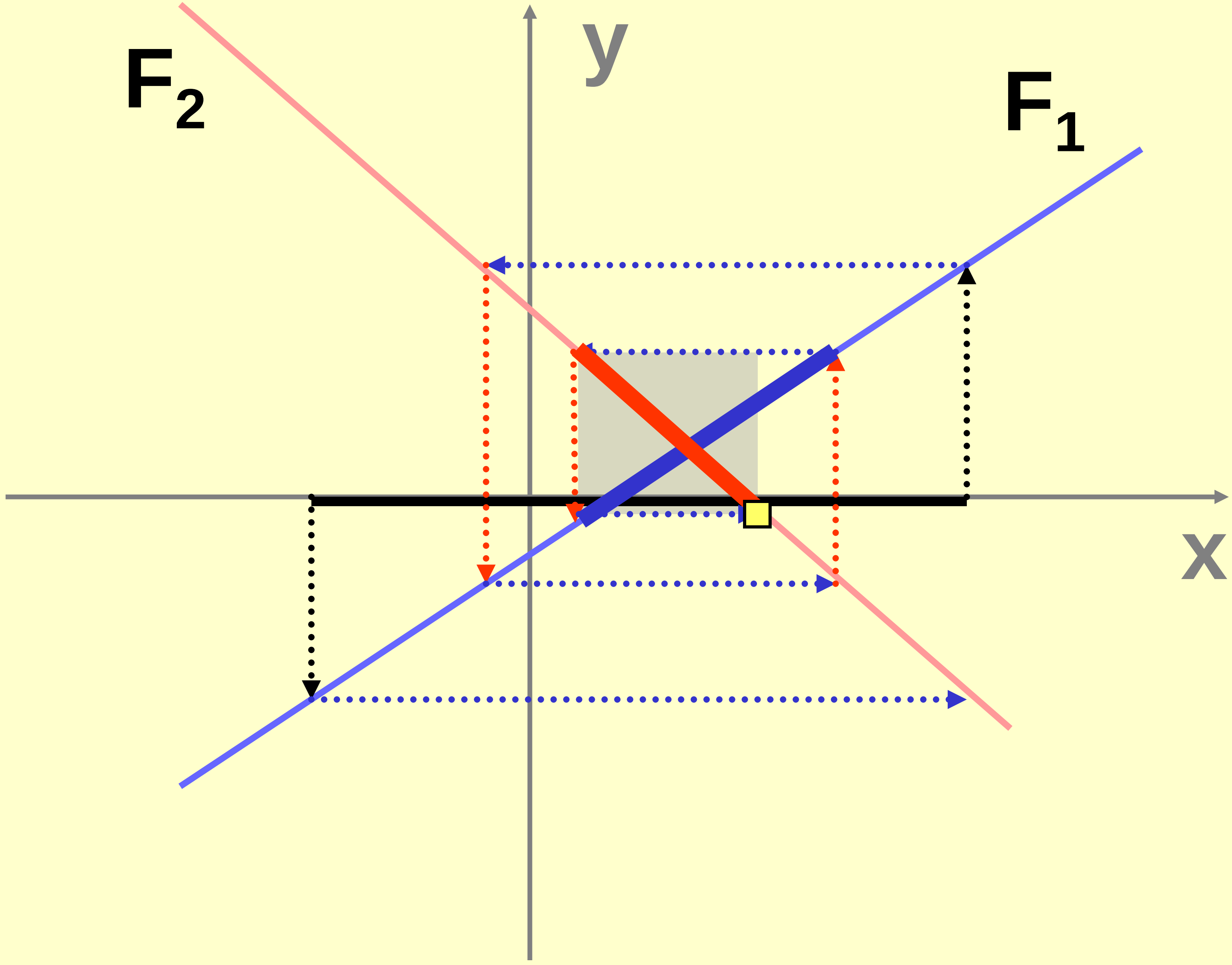


F2
y
F1
x

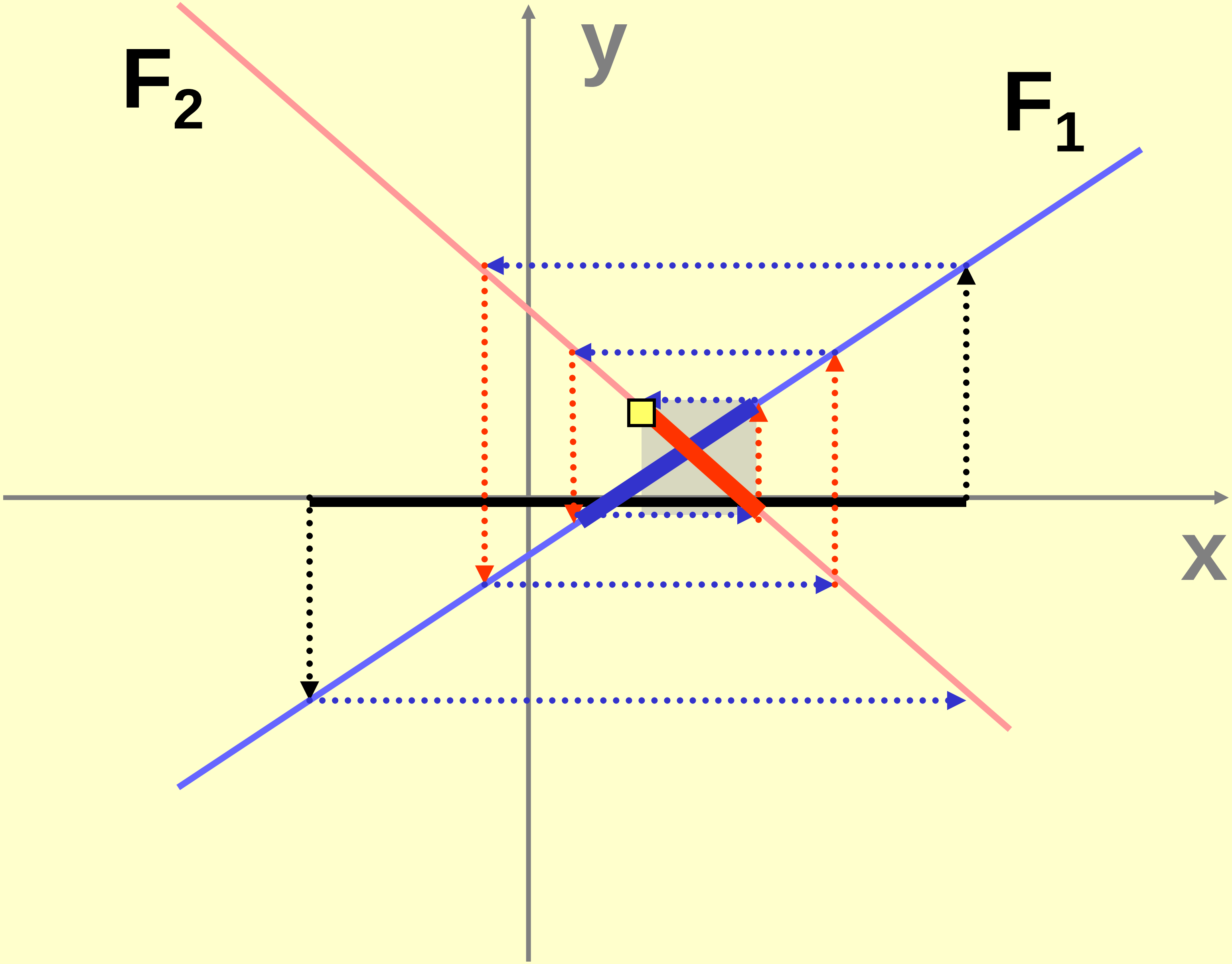


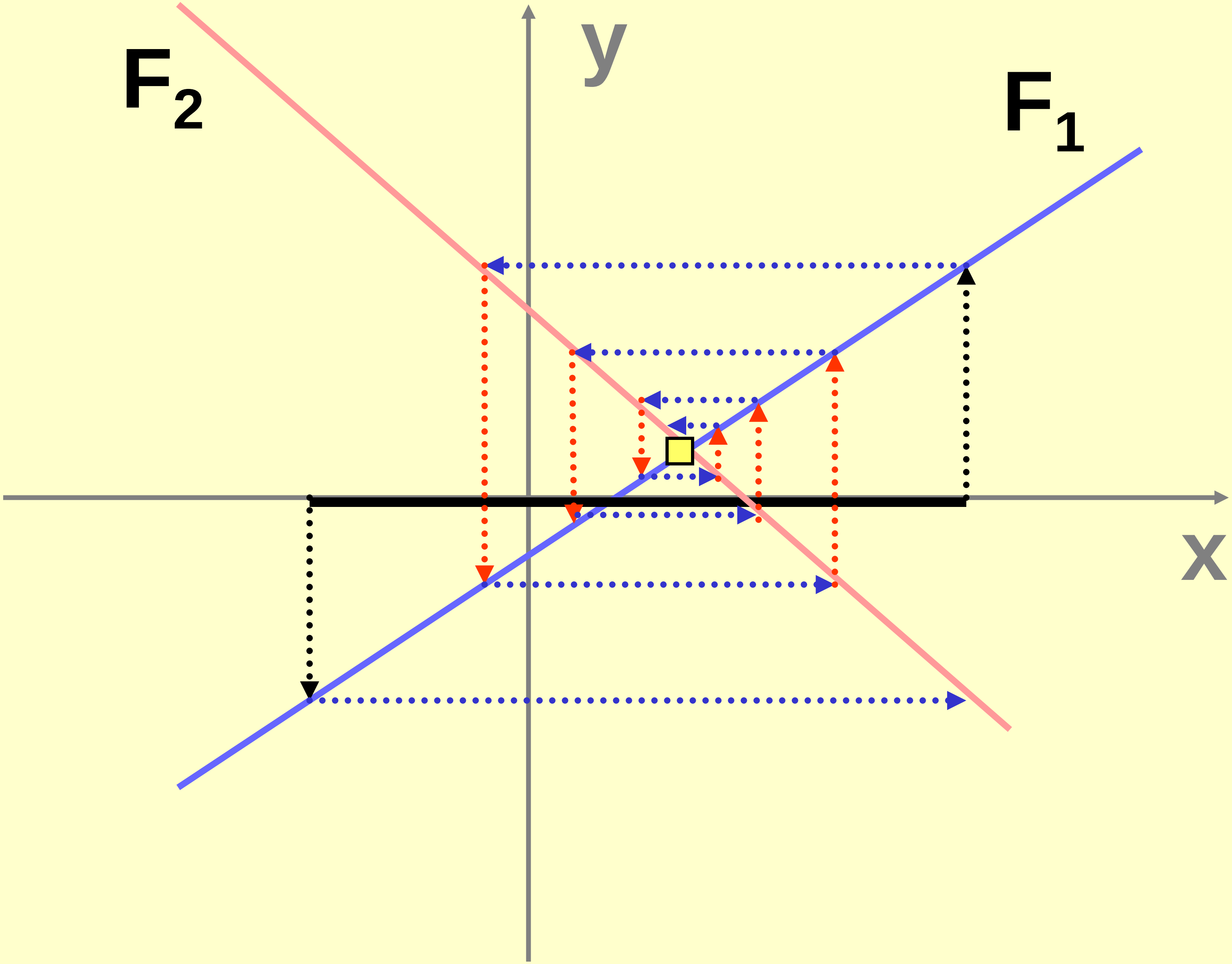
y
F_2
F_1
x


y
F2
F1
x



y
F_2
F_1
x


y
F2
F1
x


y
F2
F1
x

Ограничения метода недоопределенных вычислительных моделей

Зачастую, исходных ограничений оказывается недостаточно, чтобы получить точное решение поставленной задачи. В этом случае для получения точного решения необходимо вводить дополнительные ограничения.

Например, в качестве ограничения мы имеем уравнение:

$$x^2 = 4.$$

МНВ даст нам ответ в виде интервала $x = [-2,2]$.

Чтобы получить точное решение нужно ввести дополнительные ограничения.

Вводя $x > 0$, получаем $x = 2$. А вводя $x < 0$, получаем $x = -2$.

В решатели, построенные на основе МНВ, для поиска точных решений (или корней) часто вводят метод дихотомии, т.е. деление интервала значений искомой переменной на два интервала. В нашем случае интервал $[-2,2]$, делится на интервалы $[-2,0]$ и $[0,2]$

Применение метода недоопределенных вычислительных моделей

Н-модели применяются для решения широкого класса задач. В частности, они были использованы для задач типа САПР, задач составления и ведения сетевых графиков, некоторых классов вычислительных задач и задач оптимизации, построения интерактивных экономических моделей, арифметико-буквенных головоломок и др.

Важным достоинством данного подхода является то, что в одной Н-модели могут одновременно присутствовать разнородные отношения (линейные и нелинейные уравнения и неравенства, табличные отношения, множественные и логические соотношения).

На основе МНВ было построено несколько рещателей - Unicalc, NeMo+, система финансового планирования ФинПлан.

UniCalc: универсальный решатель задач

Решатель UniCalc предназначен для решения прямых и обратных задач, представленных системами алгебраических уравнений, неравенств и логических выражений.

Решаемая система может быть переопределенной или недоопределенной, а параметры уравнений и неравенств могут быть заданы неточно, в виде интервалов допустимых значений.

UniCalc позволяет проводить вычисления как с целыми, так и с вещественными переменными, причем они могут входить в систему одновременно.

http://uniserv.iis.nsk.su/unicalc/

UniCalc: универсальный решатель задач

$x^2 + 6*x = y - 2^k;$

$k*x + 7.7*y = 2.4;$

$(k-1)^2 < 4;$

ln$(y+2*x+12) < k+5$ **or** $y>k^2$

->

$x < -0.1$ **and** $y < 1.0;$

Модель

ядро UniCalc

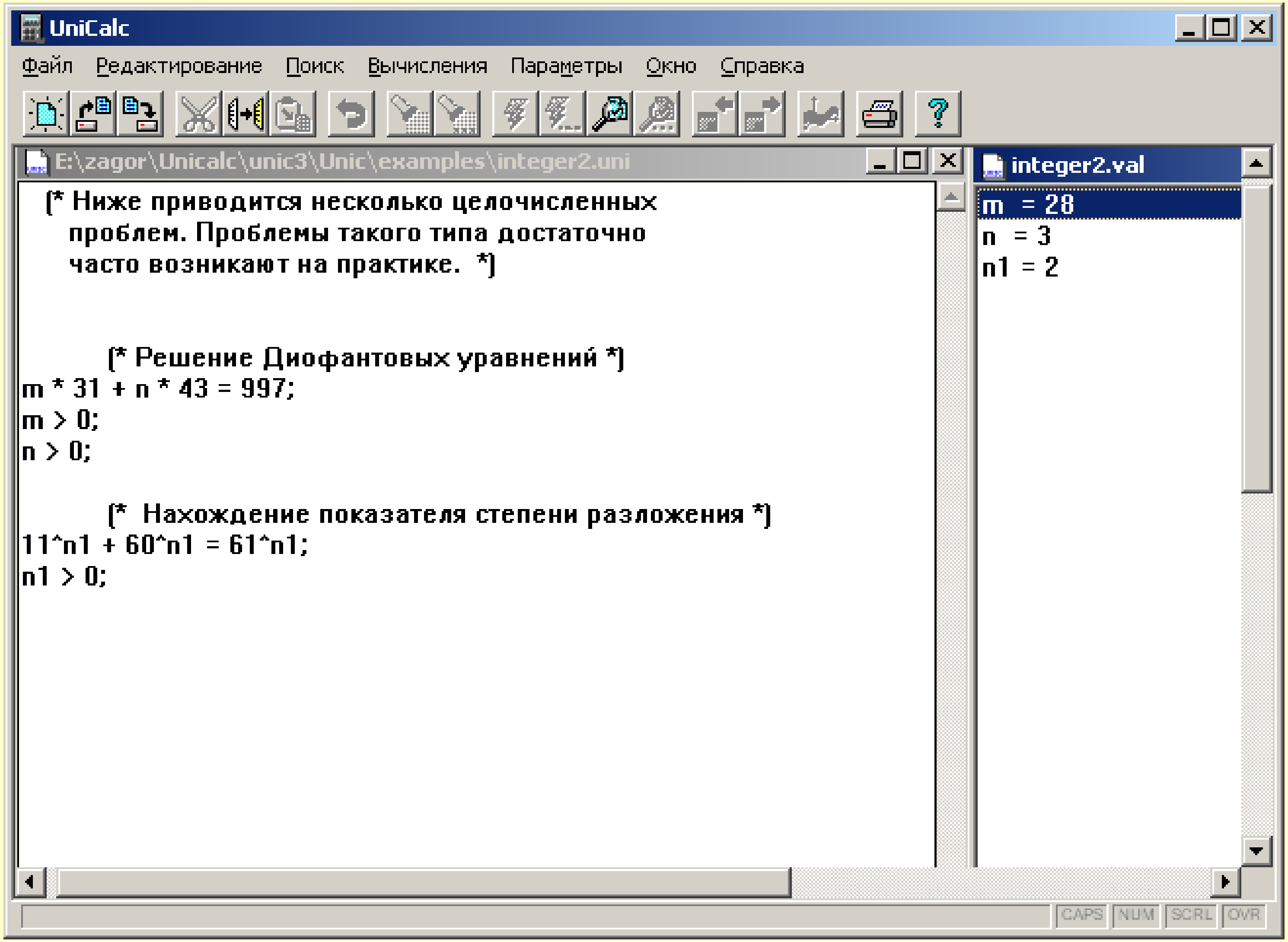
k = [0, 2]
x = [-6, -0.1]
y = [0.311688, 1]

Пространство решений

k = 0
x = [-5.88301, -5.88299]
y = 0.311688

Корни

UniCalc: универсальный решатель задач



ФинПлан: технология для решения сложных финансовых и экономических задач на основе интервальных электронных таблиц