### Образец выполнения задания № 3

***Постановка задачи***

Задана плоская фигура, ограниченная двумя кривыми, уравнения которых имеют вид:

.

Требуется разработать программу в среде Mathcad для вычисления площади указанной фигуры.

***Анализ задачи***

Построим графики заданных функций. В Mathcad-документе определим функции y1(x) и y2(x)

Используя меню **Вставка → График → График X-Y**, постоим графики двух заданных функций и определим фигуру, площадь которой следует вычислить. По умолчанию пределы изменения аргумента функций . Учитывая особенности заданных графиков, изменим пределы на .



Кривые пересекаются в двух точках и образуют замкнутую фигуру, площадь которой необходимо вычислить.

В математическом анализе доказана теорема и представлена геометрическая интерпретация понятия определенного интеграла, а именно:

 есть площадь области, ограниченной кривой функции , осью абсцисс и двумя прямыми . Тогда искомая площадь есть разность двух интегралов:

.

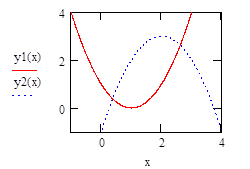
Найдем пределы интегрирования. Из графиков следует, что это абсциссы точек пересечения кривых и . Точки пересечения являются решением уравнения



или после преобразования получим

.

***Решение задачи***

В Mathcad-документе запишем:





Построим графики функций

Приравняем y1 и y2, тогда разность y1 – y2 = 0. Вычислим аналитически (нажав комбинацию клавиш **Ctrl** + **.**) эту разность.



Разделим выражение на 2 и получим уравнение



Подставим сперва в качестве х приближённое значение

x:=0

Найдём корень уравнения:





Теперь подставим в качестве х приближённое значение x2:=2





Получили .

Вычисляем интегралы и площадь:





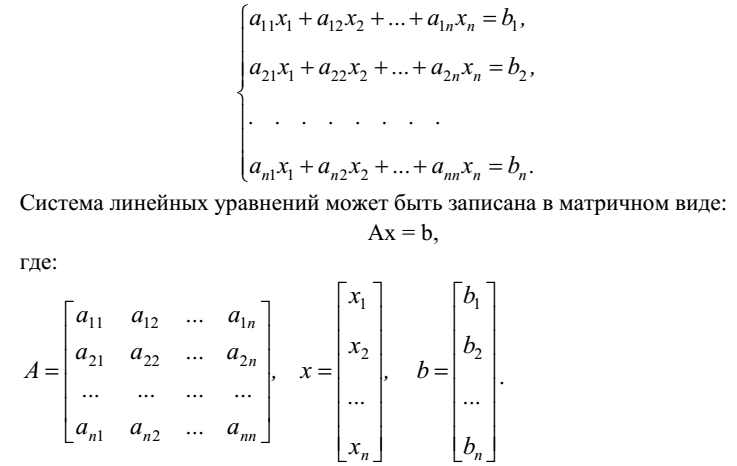


Итак, площадь фигуры равна S = 3,727.

### Образец выполнения задания № 4

*Решение систем уравнений матричным методом*

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных х1, х2, …, хn:



Если det A0 то система или эквивалентное ей матричное уравнение имеет единственное решение.

***Пример 1.*** *Решение систем уравнений с помощью функции Lsolve*

Системы линейных уравнений удобно решать с помощью функции lsolve. Функция lsolve(А, b) - возвращает вектор решения x такой, что Ах = b.

*Пример 1. Решение системы уравнений*





***Пример 2.*** *Решение системы уравнений методом Гаусса*

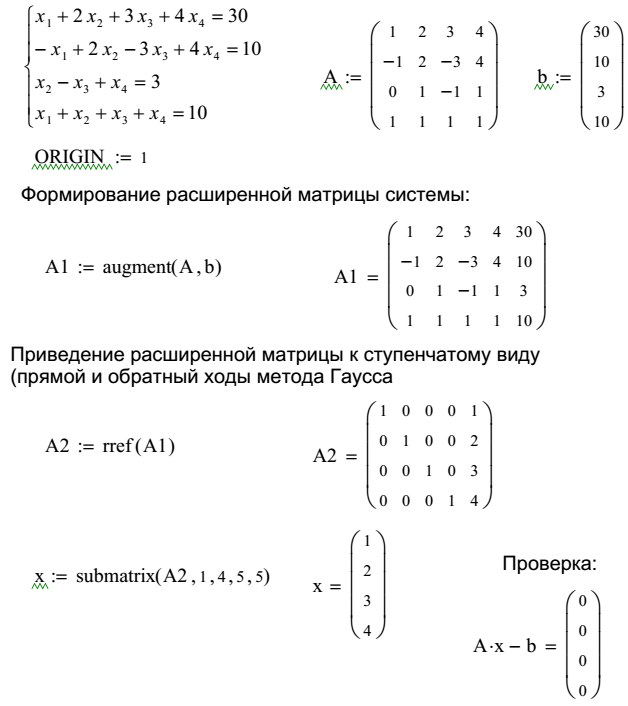
Метод Гаусса, его еще называют методом Гауссовых исключений, состоит в том, что систему уравнений приводят последовательным исключением неизвестных к эквивалентной системе с треугольной матрицей.

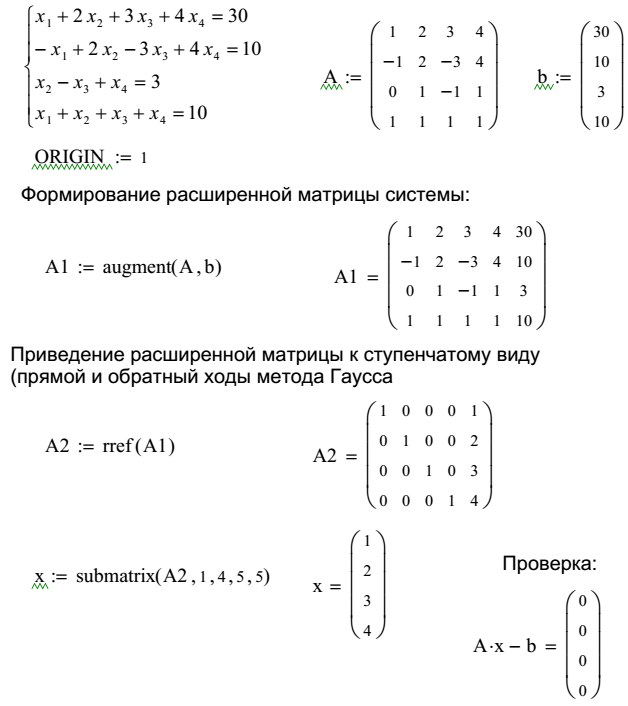
В матричной записи это означает, что сначала (прямой ход метода Гаусса) элементарными операциями над строками приводят расширенную матрицу системы к ступенчатому виду, а затем (обратный ход метода Гаусса) эту ступенчатую матрицу преобразуют так, чтобы в первых n столбцах получилась единичная матрица. Последний, (n + 1) столбец этой матрицы содержит решение системы.

В MathCAD прямой и обратный ходы метода Гаусса выполняет функция rref(A).

Решим систему уравнений методом Гаусса в MathCad

0 





***Решение систем уравнений с помощью функций Find или Minner***

Для решения системы уравнений с помощью функции Find необходимо выполнить следующее:

1. Задать начальное приближение для всех неизвестных, входящих в систему уравнений. MathCAD решает систему с помощью итерационных методов;
2. Напечатать ключевое слово Given. Оно указывает MathCAD, что далее следует система уравнений;
3. Введите уравнения и неравенства в любом порядке. Используйте [Ctrl]= для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >,≤  и ≥**;**
4. Введите любое выражение, которое включает функцию Find, например: х:= Find(х, у).
5. Ключевое слово Given, уравнения и неравенства, которые следуют за ним, и какое - либо выражение, содержащее функцию Find, называют блоком решения уравнений.

Функция *Minner* очень похожа на функцию *Find* (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minner* возвращает это приближение. Функция *Find* в этом случае возвращает сообщение об ошибке. Правила использования функции *Minner* такие же, как и функции *Find*.

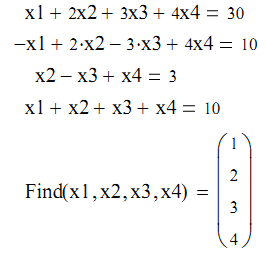
Функция **Minerr(x1, x2, . . .) -** возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое - либо выражение, содержащее функцию *Find*, называют **блоком решения уравнений**.

***Пример 3.***  *Решение системы уравнений с помощью функции Find*

x1 := 0 x2 := 0 x3 := 0 x4 := 0 Начальные приближения

Given



***Пример 4.***  *Решение системы уравнений с помощью функции Minner*

Функция Minner очень похожа на функцию Find (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, Minner возвращает это приближение. Функция Find в этом случае возвращает сообщение об ошибке.

Правила использования функции Minner такие же, как и функции Find.

Функция Minerr(x1, x2, . . .) возвращает приближенное решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

