

Código:

double[] tiempo = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 };

double[] y = { -2.4684, 56.8039, 4.8394, 13.2878, -0.1931, -19.5771, 55.4147, 74.4829, 62.0632, 37.3616, 14.1877, 72.9088, 81.5206, 120.7811, 156.3607, 141.7361, 110.8989, 129.7615, 192.8516, 253.4779 };

double factor, pivote;

int datos = y.Length;

int inco = 3;

double[,] MJ = new double[datos, inco];

double[,] max = new double[inco,inco+1];

int ren = inco;

int col = ren + 1;

for (int i = 0; i < datos; i++)

{

MJ[i, 0] = Math.Pow(tiempo[i], 2); //ADAPTAR

MJ[i, 1] = Math.Sin(tiempo[i]); //ADAPTAR

MJ[i, 2] = Math.Exp(tiempo[i] / 10); //ADAPTAR

}

for (int i = 0;i<inco; i++)

{

for(int j = 0; j < inco; j++)

{

for (int e =0; e < datos; e++)

{

max[i, j] += MJ[e, j] \* MJ[e, i];

}

}

}

for (int i = 0;i<inco; i++)

{

for(int j=0; j<datos; j++)

{

max[i, inco] += MJ[j,i] \* y[j];

}

}

for (int r = 0; r < ren; r++) //RECORRER RENGLONES r = 0 -> 1 -> 2

{

pivote = max[r, r];

for (int c = 0; c < col; c++) //RECORRER COLUMNAS C = 0

{

// if(matriz[r,c]==0)

max[r, c] = max[r, c] / pivote;

// matriz[r,c] /= pivote;

}

//VOLVER A RECORRER LA MATRIZ PARA HACER LAS CONVERSIONES A CERO

for (int rCero = 0; rCero < ren; rCero++)

{

if (r != rCero) //BRINCAR EL RENGLON DEL PIVOTE

{

factor = max[rCero, r];

for (int cCero = 0; cCero < col; cCero++)

{

//(VALOR ORIGINAL ) – (RENGLON DEL PIVOTE,C)(FACTOR))\

max[rCero, cCero] = max[rCero, cCero] - (factor \* max[r, cCero]);

//matriz[rCero, cCero] -= (factor\* matriz[r,cCero]);

}

}

}

}

Console.WriteLine("Actividad 9");

for (int r = 0; r < ren; r++)

{

Console.WriteLine("Variable " + (r + 1) + ": " + max[r, col - 1]);