**Practical Assignment - III**

**Program :** B.Tech. (*Information Technology and Mathematical Innovations)*

**Department :** Cluster Innovation Centre, University of Delhi

**Semester :** Vth

**Title of Paper :** Algorithms for Computational Mathematics: Numerical Methods

**Name of Student :** Ashutosh Jha

**Roll No. of Student :** 11811

**Date of Submission :** Sept 14, 2020

***Ques. 1:*** Write an implementation of the iterative methods i.e. Gauss - Jacobi and Gauss - Seidel.

*10x − 8y= − 6*

*−8x + 10y − z = 9*

*−y + 10z = 28*

and

*2x − y= 1*

*x + 2y − z = 1*

*−2y + 2z − w = 3*

*−z + 2w = 4*

***Sol. 1:***

* ***Gauss - Jacobi*** method for equations:

*10x − 8y= − 6*

*−8x + 10y − z = 9*

*−y + 10z = 28*

When initial guess *(x0 , y0 , z0) = (1, 1, 1)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#define **f1**(x,y,z) (-6 + 8\*y)/10

#define **f2**(x,y,z) (9 + z + 8\*x)/10

#define **f3**(x,y,z) (28 + y)/10

int main()

{

float x0=1, y0=1, z0=1, x1, y1, z1, e1, e2, e3, e;

int count=1;

e = 0.0001;

printf("\nCount\tx\ty\tz\n");

do

{

*/\* Calculation \*/*

x1 = **f1**(x0,y0,z0);

y1 = **f2**(x0,y0,z0);

z1 = **f3**(x0,y0,z0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

*/\* Error \*/*

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

count++;

*/\* Set value for next iteration \*/*

x0 = x1;

y0 = y1;

z0 = z1;

}while(e1>e && e2>e && e3>e);

printf("\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n",x1,y1,z1);

return 0;

}

***Observation Table:***

No. of Iterations done : 33

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | 0.2 | 1.8 | 2.9 |
| 2 | 0.84 | 1.35 | 2.98 |
| 3 | 0.48 | 1.87 | 2.935 |
| 4 | 0.896 | 1.5775 | 2.987 |
| 5 | 0.662 | 1.9155 | 2.9578 |
| 6 | 0.9324 | 1.7254 | 2.9915 |
| 7 | 0.7803 | 1.9451 | 2.9725 |
| 8 | 0.9561 | 1.8215 | 2.9945 |
| 9 | 0.8572 | 1.9643 | 2.9821 |
| 10 | 0.9714 | 1.884 | 2.9964 |
| 11 | 0.9072 | 1.9768 | 2.9884 |
| 12 | 0.9814 | 1.9246 | 2.9977 |
| 13 | 0.9397 | 1.9849 | 2.9925 |
| 14 | 0.9879 | 1.951 | 2.9985 |
| 15 | 0.9608 | 1.9902 | 2.9951 |
| 16 | 0.9922 | 1.9681 | 2.999 |
| 17 | 0.9745 | 1.9936 | 2.9968 |
| 18 | 0.9949 | 1.9793 | 2.9994 |
| 19 | 0.9834 | 1.9959 | 2.9979 |
| 20 | 0.9967 | 1.9865 | 2.9996 |
| 21 | 0.9892 | 1.9973 | 2.9987 |
| 22 | 0.9978 | 1.9912 | 2.9997 |
| 23 | 0.993 | 1.9982 | 2.9991 |
| 24 | 0.9986 | 1.9943 | 2.9998 |
| 25 | 0.9954 | 1.9989 | 2.9994 |
| 26 | 0.9991 | 1.9963 | 2.9999 |
| 27 | 0.997 | 1.9993 | 2.9996 |
| 28 | 0.9994 | 1.9976 | 2.9999 |
| 29 | 0.9981 | 1.9995 | 2.9998 |
| 30 | 0.9996 | 1.9984 | 3 |
| 31 | 0.9988 | 1.9997 | 2.9998 |
| 32 | 0.9998 | 1.999 | 3 |
| 33 | 0.9992 | 1.9998 | 2.9999 |

When initial guess is found using the formula *x0i = bi / (aii)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#define **f1**(x,y,z) (-6 + 8\*y)/10

#define **f2**(x,y,z) (9 + z + 8\*x)/10

#define **f3**(x,y,z) (28 + y)/10

int main()

{

float x0, y0, z0, x1, y1, z1, e1, e2, e3, e;

int count=1;

e = 0.0001;

x0 = **f1**(0, 0 , 0);

y0 = **f2**(0, 0 , 0);

z0 = **f3**(0, 0 , 0);

printf("\nCount\tx\ty\tz\n");

do

{

x1 = **f1**(x0,y0,z0);

y1 = **f2**(x0,y0,z0);

z1 = **f3**(x0,y0,z0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

*/\* Error \*/*

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

count++;

x0 = x1;

y0 = y1;

z0 = z1;

}while(e1>e && e2>e && e3>e);

printf("\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n",x1,y1,z1);

return 0;

}

***Observation Table:***

No. of Iterations done : 31

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | -0.6 | 1.1 | 2.8 |
| 2 | 0.28 | 0.7 | 2.91 |
| 3 | -0.04 | 1.415 | 2.87 |
| 4 | 0.532 | 1.155 | 2.9415 |
| 5 | 0.324 | 1.6198 | 2.9155 |
| 6 | 0.6958 | 1.4507 | 2.962 |
| 7 | 0.5606 | 1.7528 | 2.9451 |
| 8 | 0.8023 | 1.643 | 2.9753 |
| 9 | 0.7144 | 1.8393 | 2.9643 |
| 10 | 0.8715 | 1.7679 | 2.9839 |
| 11 | 0.8144 | 1.8956 | 2.9768 |
| 12 | 0.9165 | 1.8492 | 2.9896 |
| 13 | 0.8793 | 1.9321 | 2.9849 |
| 14 | 0.9457 | 1.902 | 2.9932 |
| 15 | 0.9216 | 1.9559 | 2.9902 |
| 16 | 0.9647 | 1.9363 | 2.9956 |
| 17 | 0.949 | 1.9713 | 2.9936 |
| 18 | 0.9771 | 1.9586 | 2.9971 |
| 19 | 0.9669 | 1.9814 | 2.9959 |
| 20 | 0.9851 | 1.9731 | 2.9981 |
| 21 | 0.9785 | 1.9879 | 2.9973 |
| 22 | 0.9903 | 1.9825 | 2.9988 |
| 23 | 0.986 | 1.9921 | 2.9982 |
| 24 | 0.9937 | 1.9886 | 2.9992 |
| 25 | 0.9909 | 1.9949 | 2.9989 |
| 26 | 0.9959 | 1.9926 | 2.9995 |
| 27 | 0.9941 | 1.9967 | 2.9993 |
| 28 | 0.9973 | 1.9952 | 2.9997 |
| 29 | 0.9962 | 1.9978 | 2.9995 |
| 30 | 0.9983 | 1.9969 | 2.9998 |
| 31 | 0.9975 | 1.9986 | 2.9997 |

* ***Gauss - Jacobi*** method for equations:

*2x − y= 1*

*x + 2y − z = 1*

*−2y + 2z − w = 3*

*−z + 2w = 4*

When initial guess *(x0 , y0 , z0 , w0) = (1, 1, 1, 1)*

***Program (using C):***

#include<cstdio>

#include<cmath>

#define **f1**(x,y,z,w) (1 + y)/2

#define **f2**(x,y,z,w) (1 + z - x)/2

#define **f3**(x,y,z,w) (3 + w + 2\*y)/2

#define **f4**(x,y,z,w) (4+z)/2

int main()

{

float x0=1, y0=1, z0=1, w0 =1, x1, y1, z1, w1, e1, e2, e3, e4, e;

int count=1;

e = 0.0001;

printf("\nCount\tx\ty\tz\n");

do

{

x1 = **f1**(x0, y0, z0, w0);

y1 = **f2**(x0, y0, z0, w0);

z1 = **f3**(x0, y0, z0, w0);

w1 = **f4**(x0, y0, z0, w0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1, w1);

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

e4 = std::fabs(w0-w1);

count++;

x0 = x1;

y0 = y1;

z0 = z1;

w0 = w1;

}while(e1>e || e2>e || e3>e || e4 > e);

printf("\nSolution: x=%0.4f, y=%0.4f and z = %0.4f and w = %0.4f\n",x1,y1,z1,w1);

exit (0);

return 0;

}

***Observation Table***

No. of iterations done : 40

| ***Count*** | ***x*** | ***y*** | ***z*** | ***w*** |
| --- | --- | --- | --- | --- |
| 1 | 1 | 0.5 | 3 | 2.5 |
| 2 | 0.75 | 1.5 | 3.25 | 3.5 |
| 3 | 1.25 | 1.75 | 4.75 | 3.625 |
| 4 | 1.375 | 2.25 | 5.0625 | 4.375 |
| 5 | 1.625 | 2.3438 | 5.9375 | 4.5313 |
| 6 | 1.6719 | 2.6563 | 6.1094 | 4.9688 |
| 7 | 1.8281 | 2.7188 | 6.6406 | 5.0547 |
| 8 | 1.8594 | 2.9063 | 6.7461 | 5.3203 |
| 9 | 1.9531 | 2.9434 | 7.0664 | 5.373 |
| 10 | 1.9717 | 3.0566 | 7.1299 | 5.5332 |
| 11 | 2.0283 | 3.0791 | 7.3232 | 5.5649 |
| 12 | 2.0396 | 3.1475 | 7.3616 | 5.6616 |
| 13 | 2.0737 | 3.161 | 7.4783 | 5.6808 |
| 14 | 2.0805 | 3.2023 | 7.5014 | 5.7391 |
| 15 | 2.1011 | 3.2104 | 7.5718 | 5.7507 |
| 16 | 2.1052 | 3.2354 | 7.5858 | 5.7859 |
| 17 | 2.1177 | 3.2403 | 7.6283 | 5.7929 |
| 18 | 2.1201 | 3.2553 | 7.6367 | 5.8142 |
| 19 | 2.1277 | 3.2583 | 7.6624 | 5.8184 |
| 20 | 2.1291 | 3.2674 | 7.6675 | 5.8312 |
| 21 | 2.1337 | 3.2692 | 7.683 | 5.8337 |
| 22 | 2.1346 | 3.2746 | 7.686 | 5.8415 |
| 23 | 2.1373 | 3.2757 | 7.6954 | 5.843 |
| 24 | 2.1379 | 3.279 | 7.6972 | 5.8477 |
| 25 | 2.1395 | 3.2797 | 7.7029 | 5.8486 |
| 26 | 2.1398 | 3.2817 | 7.704 | 5.8514 |
| 27 | 2.1408 | 3.2821 | 7.7074 | 5.852 |
| 28 | 2.141 | 3.2833 | 7.7081 | 5.8537 |
| 29 | 2.1416 | 3.2835 | 7.7101 | 5.854 |
| 30 | 2.1418 | 3.2842 | 7.7105 | 5.8551 |
| 31 | 2.1421 | 3.2844 | 7.7118 | 5.8553 |
| 32 | 2.1422 | 3.2848 | 7.712 | 5.8559 |
| 33 | 2.1424 | 3.2849 | 7.7128 | 5.856 |
| 34 | 2.1425 | 3.2852 | 7.7129 | 5.8564 |
| 35 | 2.1426 | 3.2852 | 7.7134 | 5.8565 |
| 36 | 2.1426 | 3.2854 | 7.7135 | 5.8567 |
| 37 | 2.1427 | 3.2854 | 7.7137 | 5.8567 |
| 38 | 2.1427 | 3.2855 | 7.7138 | 5.8569 |
| 39 | 2.1428 | 3.2855 | 7.714 | 5.8569 |
| 40 | 2.1428 | 3.2856 | 7.714 | 5.857 |

When initial guess is found using the formula *x0i  = bi / (aii)*

***Program (using C)***

#include<cstdio>

#include<cmath>

#define **f1**(x,y,z,w) (1 + y)/2

#define **f2**(x,y,z,w) (1 + z - x)/2

#define **f3**(x,y,z,w) (3 + w + 2\*y)/2

#define **f4**(x,y,z,w) (4+z)/2

int main()

{

float x0,y0,z0,w0, x1, y1, z1, w1, e1, e2, e3, e4, e;

int count=1;

e = 0.0001;

*//Making an initial guess*

x0 = **f1**(0,0,0,0);

y0 = **f2**(0,0,0,0);

z0 = **f3**(0,0,0,0);

w0 = **f4**(0,0,0,0);

printf("\nCount\tx\ty\tz\n");

do

{

x1 = **f1**(x0, y0, z0, w0);

y1 = **f2**(x0, y0, z0, w0);

z1 = **f3**(x0, y0, z0, w0);

w1 = **f4**(x0, y0, z0, w0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1, w1);

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

e4 = std::fabs(w0-w1);

count++;

x0 = x1;

y0 = y1;

z0 = z1;

w0 = w1;

}while(e1>e || e2>e || e3>e || e4 > e);

printf("\nSolution: x=%0.4f, y=%0.4f and z = %0.4f and w = %0.4f\n",x1,y1,z1,w1);

exit (0);

return 0;

}

***Observation Table:***

No. of Iterations done : 39

| ***Count*** | ***x*** | ***y*** | ***z*** | ***w*** |
| --- | --- | --- | --- | --- |
| 1 | 0.5 | 1 | 2.5 | 2.5 |
| 2 | 1 | 1.5 | 3.75 | 3.25 |
| 3 | 1.25 | 1.875 | 4.625 | 3.875 |
| 4 | 1.4375 | 2.1875 | 5.3125 | 4.3125 |
| 5 | 1.5938 | 2.4375 | 5.8438 | 4.6562 |
| 6 | 1.7188 | 2.625 | 6.2656 | 4.9219 |
| 7 | 1.8125 | 2.7734 | 6.5859 | 5.1328 |
| 8 | 1.8867 | 2.8867 | 6.8398 | 5.293 |
| 9 | 1.9434 | 2.9766 | 7.0332 | 5.4199 |
| 10 | 1.9883 | 3.0449 | 7.1865 | 5.5166 |
| 11 | 2.0225 | 3.0991 | 7.3032 | 5.5933 |
| 12 | 2.0496 | 3.1404 | 7.3958 | 5.6516 |
| 13 | 2.0702 | 3.1731 | 7.4662 | 5.6979 |
| 14 | 2.0865 | 3.198 | 7.522 | 5.7331 |
| 15 | 2.099 | 3.2177 | 7.5645 | 5.761 |
| 16 | 2.1089 | 3.2328 | 7.5983 | 5.7823 |
| 17 | 2.1164 | 3.2447 | 7.6239 | 5.7991 |
| 18 | 2.1223 | 3.2538 | 7.6443 | 5.812 |
| 19 | 2.1269 | 3.261 | 7.6597 | 5.8221 |
| 20 | 2.1305 | 3.2664 | 7.672 | 5.8299 |
| 21 | 2.1332 | 3.2708 | 7.6814 | 5.836 |
| 22 | 2.1354 | 3.2741 | 7.6888 | 5.8407 |
| 23 | 2.137 | 3.2767 | 7.6944 | 5.8444 |
| 24 | 2.1383 | 3.2787 | 7.6989 | 5.8472 |
| 25 | 2.1393 | 3.2803 | 7.7023 | 5.8494 |
| 26 | 2.1401 | 3.2815 | 7.705 | 5.8511 |
| 27 | 2.1407 | 3.2824 | 7.707 | 5.8525 |
| 28 | 2.1412 | 3.2832 | 7.7087 | 5.8535 |
| 29 | 2.1416 | 3.2837 | 7.7099 | 5.8543 |
| 30 | 2.1419 | 3.2842 | 7.7109 | 5.855 |
| 31 | 2.1421 | 3.2845 | 7.7116 | 5.8555 |
| 32 | 2.1423 | 3.2848 | 7.7122 | 5.8558 |
| 33 | 2.1424 | 3.285 | 7.7127 | 5.8561 |
| 34 | 2.1425 | 3.2852 | 7.7131 | 5.8563 |
| 35 | 2.1426 | 3.2853 | 7.7133 | 5.8565 |
| 36 | 2.1426 | 3.2854 | 7.7135 | 5.8567 |
| 37 | 2.1427 | 3.2855 | 7.7137 | 5.8568 |
| 38 | 2.1427 | 3.2855 | 7.7138 | 5.8569 |
| 39 | 2.1428 | 3.2856 | 7.7139 | 5.8569 |

* ***Gauss - Seidel*** method for equations:

*10x − 8y= − 6*

*−8x + 10y − z = 9*

*−y + 10z = 28*

When initial guess *(x0 , y0 , z0) = (1, 1, 1)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#include<cmath>

#define **f1**(x,y,z) (-6 + 8\*y)/10

#define **f2**(x,y,z) (9 + z + 8\*x)/10

#define **f3**(x,y,z) (28 + y)/10

int main()

{

float x0=1, y0=1, z0=1, x1, y1, z1, e1, e2, e3, e;

int count=1;

e = 0.0001;

printf("\nCount\tx\ty\tz\n");

do

{

*/\* Calculation \*/*

x1 = **f1**(x0,y0,z0);

y1 = **f2**(x1,y0,z0);

z1 = **f3**(x1,y1,z0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

*/\* Error \*/*

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

count++;

*/\* Set value for next iteration \*/*

x0 = x1;

y0 = y1;

z0 = z1;

}while(e1>e && e2>e && e3>e);

printf("\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n",x1,y1,z1);

return 0;

}

***Observation Table:***

No. of iterations = 16

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | 0.2 | 1.16 | 2.916 |
| 2 | 0.328 | 1.454 | 2.9454 |
| 3 | 0.5632 | 1.6451 | 2.9645 |
| 4 | 0.7161 | 1.7693 | 2.9769 |
| 5 | 0.8155 | 1.8501 | 2.985 |
| 6 | 0.88 | 1.9025 | 2.9903 |
| 7 | 0.922 | 1.9366 | 2.9937 |
| 8 | 0.9493 | 1.9588 | 2.9959 |
| 9 | 0.9671 | 1.9732 | 2.9973 |
| 10 | 0.9786 | 1.9826 | 2.9983 |
| 11 | 0.9861 | 1.9887 | 2.9989 |
| 12 | 0.991 | 1.9926 | 2.9993 |
| 13 | 0.9941 | 1.9952 | 2.9995 |
| 14 | 0.9962 | 1.9969 | 2.9997 |
| 15 | 0.9975 | 1.998 | 2.9998 |
| 16 | 0.9984 | 1.9987 | 2.9999 |

When initial guess is found using the formula *x0i  = bi / (aii)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#define **f1**(x,y,z) (-6 + 8\*y)/10

#define **f2**(x,y,z) (9 + z + 8\*x)/10

#define **f3**(x,y,z) (28 + y)/10

*/\* Main function \*/*

int main()

{

float x0, y0, z0, x1, y1, z1, e1, e2, e3, e;

int count=1;

e = 0.0001;

x0 = **f1**(0,0,0);

y0 = **f2**(0,0,0);

z0 = **f3**(0,0,0);

printf("\nCount\tx\ty\tz\n");

do

{

*/\* Calculation \*/*

x1 = **f1**(x0,y0,z0);

y1 = **f2**(x1,y0,z0);

z1 = **f3**(x1,y1,z0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

*/\* Error \*/*

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

count++;

*/\* Set value for next iteration \*/*

x0 = x1;

y0 = y1;

z0 = z1;

}while(e1>e && e2>e && e3>e);

printf("\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n",x1,y1,z1);

return 0;

}

***Observation Table***

No. of iterations = 17

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | -0.6 | 0.62 | 2.862 |
| 2 | -0.104 | 1.103 | 2.9103 |
| 3 | 0.2824 | 1.4169 | 2.9417 |
| 4 | 0.5336 | 1.621 | 2.9621 |
| 5 | 0.6968 | 1.7537 | 2.9754 |
| 6 | 0.8029 | 1.8399 | 2.984 |
| 7 | 0.8719 | 1.8959 | 2.9896 |
| 8 | 0.9167 | 1.9323 | 2.9932 |
| 9 | 0.9459 | 1.956 | 2.9956 |
| 10 | 0.9648 | 1.9714 | 2.9971 |
| 11 | 0.9771 | 1.9814 | 2.9981 |
| 12 | 0.9851 | 1.9879 | 2.9988 |
| 13 | 0.9903 | 1.9922 | 2.9992 |
| 14 | 0.9937 | 1.9949 | 2.9995 |
| 15 | 0.9959 | 1.9967 | 2.9997 |
| 16 | 0.9973 | 1.9978 | 2.9998 |
| 17 | 0.9983 | 1.9986 | 2.9999 |

* ***Gauss - Seidel*** method for equations:

*2x − y= 1*

*x + 2y − z = 1*

*−2y + 2z − w = 3*

*−z + 2w = 4*

When initial guess *(x0 , y0 , z0 , w0) = (1, 1, 1, 1)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#define **f1**(x,y,z,w) (1 + y)/2

#define **f2**(x,y,z,w) (1 + z - x)/2

#define **f3**(x,y,z,w) (3 + w + 2\*y)/2

#define **f4**(x,y,z,w) (4+z)/2

*/\* Main function \*/*

int main()

{

float x0=1, y0=1, z0=1, w0 =1, x1, y1, z1, w1,e1, e2, e3, e4, e;

int count=1;

e = 0.0001;

printf("\nCount\tx\ty\tz\n");

do

{

x1 = **f1**(x0, y0, z0, w0);

y1 = **f2**(x1, y0, z0, w0);

z1 = **f3**(x1, y1, z0, w0);

w1 = **f4**(x1, y1, z1, w0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

e4 = std::fabs(w0-w1);

count++;

x0 = x1;

y0 = y1;

z0 = z1;

w0 = w1;

}while(e1>e || e2>e || e3>e || e4>e);

printf("\nSolution: x=%0.4f, y=%0.4f and z = %0.4f and w = %0.4f\n",x1,y1,z1,w1);

return 0;

}

***Observation Table:***

No. of iterations : 22

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | 1 | 0.5 | 2.5 |
| 2 | 0.75 | 1.375 | 4.5 |
| 3 | 1.1875 | 2.1562 | 5.7812 |
| 4 | 1.5781 | 2.6016 | 6.5469 |
| 5 | 1.8008 | 2.873 | 7.0098 |
| 6 | 1.9365 | 3.0366 | 7.2891 |
| 7 | 2.0183 | 3.1354 | 7.4576 |
| 8 | 2.0677 | 3.195 | 7.5594 |
| 9 | 2.0975 | 3.2309 | 7.6208 |
| 10 | 2.1155 | 3.2527 | 7.6579 |
| 11 | 2.1263 | 3.2658 | 7.6802 |
| 12 | 2.1329 | 3.2737 | 7.6937 |
| 13 | 2.1368 | 3.2784 | 7.7019 |
| 14 | 2.1392 | 3.2813 | 7.7068 |
| 15 | 2.1407 | 3.2831 | 7.7098 |
| 16 | 2.1415 | 3.2841 | 7.7116 |
| 17 | 2.1421 | 3.2847 | 7.7126 |
| 18 | 2.1424 | 3.2851 | 7.7133 |
| 19 | 2.1426 | 3.2854 | 7.7137 |
| 20 | 2.1427 | 3.2855 | 7.7139 |
| 21 | 2.1428 | 3.2856 | 7.7141 |
| 22 | 2.1428 | 3.2856 | 7.7142 |

When initial guess is found using the formula *x0i  = bi / (aii)*

***Program (using C):***

#include <cmath>

#include<cstdio>

#define **f1**(x,y,z,w) (1 + y)/2

#define **f2**(x,y,z,w) (1 + z - x)/2

#define **f3**(x,y,z,w) (3 + w + 2\*y)/2

#define **f4**(x,y,z,w) (4+z)/2

int main()

{

float x0, y0, z0, w0 , x1, y1, z1, w1,e1, e2, e3, e4, e;

int count=1;

e = 0.0001;

x0 = **f1**(0,0,0,0);

y0 = **f2**(0,0,0,0);

z0 = **f3**(0,0,0,0);

w0 = **f4**(0,0,0,0);

printf("\nCount\tx\ty\tz\n");

do

{

*/\* Calculation \*/*

x1 = **f1**(x0, y0, z0, w0);

y1 = **f2**(x1, y0, z0, w0);

z1 = **f3**(x1, y1, z0, w0);

w1 = **f4**(x1, y1, z1, w0);

printf("%d\t%0.4f\t%0.4f\t%0.4f\n",count, x1,y1,z1);

e1 = std::fabs(x0-x1);

e2 = std::fabs(y0-y1);

e3 = std::fabs(z0-z1);

e4 = std::fabs(w0-w1);

count++;

x0 = x1;

y0 = y1;

z0 = z1;

w0 = w1;

}while(e1>e || e2>e || e3>e || e4>e);

printf("\nSolution: x=%0.4f, y=%0.4f and z = %0.4f and w = %0.4f\n",x1,y1,z1,w1);

return 0;

}

***Observation Table:***

No. of iterations : 22

| ***Count*** | ***x*** | ***y*** | ***z*** |
| --- | --- | --- | --- |
| 1 | 0.5 | 0.75 | 3.25 |
| 2 | 0.875 | 1.6875 | 5 |
| 3 | 1.3438 | 2.3281 | 6.0781 |
| 4 | 1.6641 | 2.707 | 6.7266 |
| 5 | 1.8535 | 2.9365 | 7.1182 |
| 6 | 1.9683 | 3.075 | 7.3545 |
| 7 | 2.0375 | 3.1585 | 7.4971 |
| 8 | 2.0793 | 3.2089 | 7.5832 |
| 9 | 2.1045 | 3.2394 | 7.6352 |
| 10 | 2.1197 | 3.2577 | 7.6665 |
| 11 | 2.1289 | 3.2688 | 7.6855 |
| 12 | 2.1344 | 3.2755 | 7.6969 |
| 13 | 2.1378 | 3.2796 | 7.7038 |
| 14 | 2.1398 | 3.282 | 7.708 |
| 15 | 2.141 | 3.2835 | 7.7105 |
| 16 | 2.1417 | 3.2844 | 7.712 |
| 17 | 2.1422 | 3.2849 | 7.7129 |
| 18 | 2.1424 | 3.2852 | 7.7134 |
| 19 | 2.1426 | 3.2854 | 7.7138 |
| 20 | 2.1427 | 3.2855 | 7.714 |
| 21 | 2.1428 | 3.2856 | 7.7141 |
| 22 | 2.1428 | 3.2856 | 7.7142 |

***Ques. 2:*** Compare the convergence of the methods of Jacobi and Gauss-Seidel. Make a table with three columns: Computer time, number of iterations needed by Jacobi, and number of iterations needed by Gauss-Seidel. (*tolerance= .0001*)

***Sol. 2:*** As we can infer from the no. of iterations table from each corresponding program, Gauss Seidel converges way faster than Jacobi for these equations. In Jacobi, we calculate *x11, x21, x31, ---- xn1* from initial values *x1, x2, x3------x4* and so on we calculate as long as the stopping criteria is met. In seidal, we don’t calculate all the *x11, x21, x31, ---- xn1*. Instead we calculate *x11*and when finding *x21*, we substitute *x11*, so this method is better than jacobi and faster in most cases as it uses updated values.

| ***S.No.*** | ***Operation*** | ***Equation No.*** | ***Guess*** | ***Jacobi*** | ***Seidal*** |
| --- | --- | --- | --- | --- | --- |
| 1 | Iterations | I | (1,1,1) | 33 | 16 |
| *x0i  = bi / aii* | 31 | 17 |
| II | (1,1,1,1) | 40 | 22 |
| *x0i  = bi / ai*i | 39 | 22 |
| 2 | Program Build Time (in ms) | I | (1,1,1) | 713ms | 243ms |
| *x0i  = bi / aii* | 632ms | 287ms |
| II | (1,1,1,1) | 849ms | 349ms |
| *x0i  = bi / aii* | 828ms | 352ms |

***Ques. 3:*** Draw a graph between the number of iterations and error arises in both these methods at each iteration.

***Sol. 3:***

***Observation Table for Errors:***

| ***Iterations*** | ***Gauss - Jacobi*** | |  | | ***Gauss - Seidel*** | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Equations I* | | *Equations II* | | *Equations I* | | *Equations II* | |
|  | *(1, 1, 1)* | *x0i  = bi / ai* | *(1, 1, 1, 1)* | *x0i  = bi / ai* | *(1, 1, 1)* | *x0i  = bi / ai* | *(1, 1, 1, 1)* | *x0i  = bi / ai* |
| 1 | 1.9 | 1.1 | 2 | 1.5 | 1.916 | 0.862 | 2.25 | 2.25 |
| 2 | 0.64 | 0.88 | 1 | 1.25 | 0.294 | 0.496 | 2 | 1.75 |
| 3 | 0.52 | 0.715 | 1.5 | 0.875 | 0.2352 | 0.3864 | 1.281 | 1.078 |
| 4 | 0.416 | 0.572 | 0.75 | 0.6875 | 0.1529 | 0.2512 | 0.766 | 0.648 |
| 5 | 0.338 | 0.4648 | 0.875 | 0.5312 | 0.0994 | 0.1633 | 0.463 | 0.392 |
| 6 | 0.2704 | 0.3718 | 0.4375 | 0.4219 | 0.0646 | 0.1061 | 0.279 | 0.236 |
| 7 | 0.2197 | 0.3021 | 0.5312 | 0.3203 | 0.042 | 0.069 | 0.169 | 0.143 |
| 8 | 0.1758 | 0.2417 | 0.2656 | 0.2539 | 0.0273 | 0.0448 | 0.102 | 0.086 |
| 9 | 0.1428 | 0.1964 | 0.3203 | 0.1934 | 0.0177 | 0.0291 | 0.061 | 0.052 |
| 10 | 0.1142 | 0.1571 | 0.1602 | 0.1533 | 0.0115 | 0.0189 | 0.037 | 0.031 |
| 11 | 0.0928 | 0.1276 | 0.1934 | 0.1167 | 0.0075 | 0.0123 | 0.022 | 0.019 |
| 12 | 0.0743 | 0.1021 | 0.0967 | 0.0925 | 0.0049 | 0.008 | 0.014 | 0.011 |
| 13 | 0.0603 | 0.083 | 0.1167 | 0.0704 | 0.0032 | 0.0052 | 0.008 | 0.007 |
| 14 | 0.0483 | 0.0664 | 0.0583 | 0.0558 | 0.0021 | 0.0034 | 0.005 | 0.004 |
| 15 | 0.0392 | 0.0539 | 0.0704 | 0.0425 | 0.0013 | 0.0022 | 0.003 | 0.003 |
| 16 | 0.0314 | 0.0431 | 0.0352 | 0.0337 | 0.0009 | 0.0014 | 0.002 | 0.002 |
| 17 | 0.0255 | 0.0351 | 0.0425 | 0.0257 |  | 0.0009 | 0.001 | 0.001 |
| 18 | 0.0204 | 0.028 | 0.0213 | 0.0203 |  |  | 0.001 | 0.001 |
| 19 | 0.0166 | 0.0228 | 0.0257 | 0.0155 |  |  |  |  |
| 20 | 0.0133 | 0.0182 | 0.0128 | 0.0123 |  |  |  |  |
| 21 | 0.0108 | 0.0148 | 0.0155 | 0.0093 |  |  |  |  |
| 22 | 0.0086 | 0.0118 | 0.0077 | 0.0074 |  |  |  |  |
| 23 | 0.007 | 0.0096 | 0.0093 | 0.0056 |  |  |  |  |
| 24 | 0.0056 | 0.0077 | 0.0047 | 0.0045 |  |  |  |  |
| 25 | 0.0046 | 0.0063 | 0.0056 | 0.0034 |  |  |  |  |
| 26 | 0.0036 | 0.005 | 0.0028 | 0.0027 |  |  |  |  |
| 27 | 0.003 | 0.0041 | 0.0034 | 0.0021 |  |  |  |  |
| 28 | 0.0024 | 0.0033 | 0.0017 | 0.0016 |  |  |  |  |
| 29 | 0.0019 | 0.0026 | 0.0021 | 0.0012 |  |  |  |  |
| 30 | 0.0015 | 0.0021 | 0.001 | 0.001 |  |  |  |  |
| 31 | 0.0012 | 0.0017 | 0.0012 | 0.0007 |  |  |  |  |
| 32 | 0.001 |  | 0.0006 | 0.0006 |  |  |  |  |
| 33 | 0.0008 |  | 0.0007 | 0.0005 |  |  |  |  |
| 34 |  |  | 0.0004 | 0.0004 |  |  |  |  |
| 35 |  |  | 0.0005 | 0.0003 |  |  |  |  |
| 36 |  |  | 0.0002 | 0.0002 |  |  |  |  |
| 37 |  |  | 0.0003 | 0.0002 |  |  |  |  |
| 38 |  |  | 0.0001 | 0.0001 |  |  |  |  |
| 39 |  |  | 0.0002 | 0.0001 |  |  |  |  |
| 40 |  |  | 0.0001 |  |  |  |  |  |

***Graphs for Errors:***

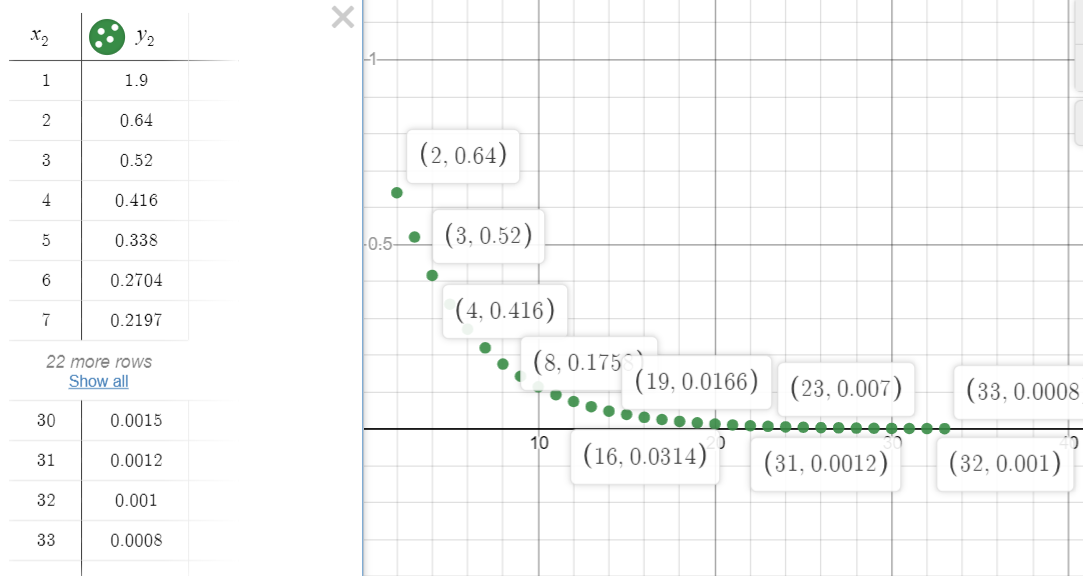
* ***Gauss - Jacobi*** method for equations:

*10x − 8y= − 6*

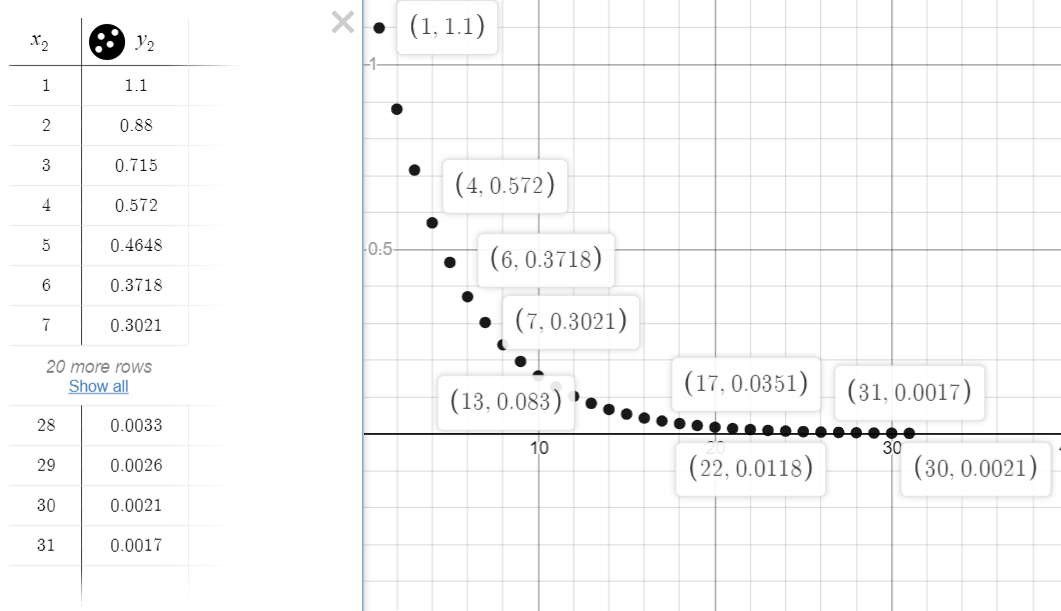
*−8x + 10y − z = 9*

*−y + 10z = 28*

When initial guess *(x0 , y0 , z0) = (1, 1, 1)*



When initial guess is found using the formula *x0i = bi / (aii)*



* ***Gauss - Jacobi*** method for equations:

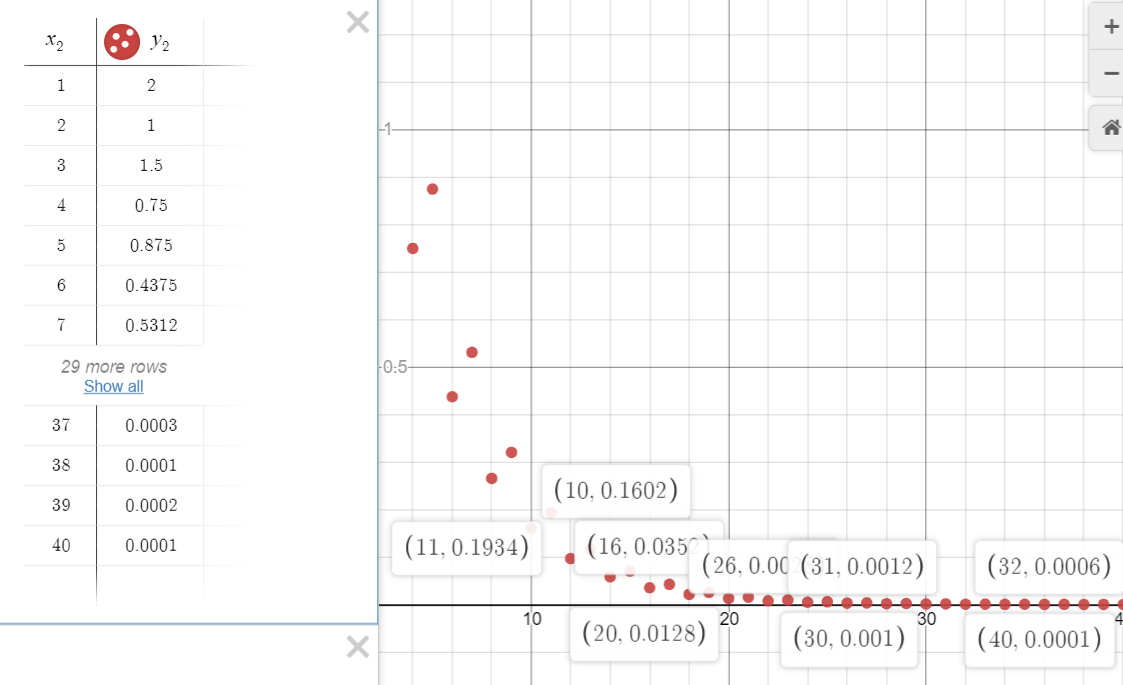
*2x − y= 1*

*x + 2y − z = 1*

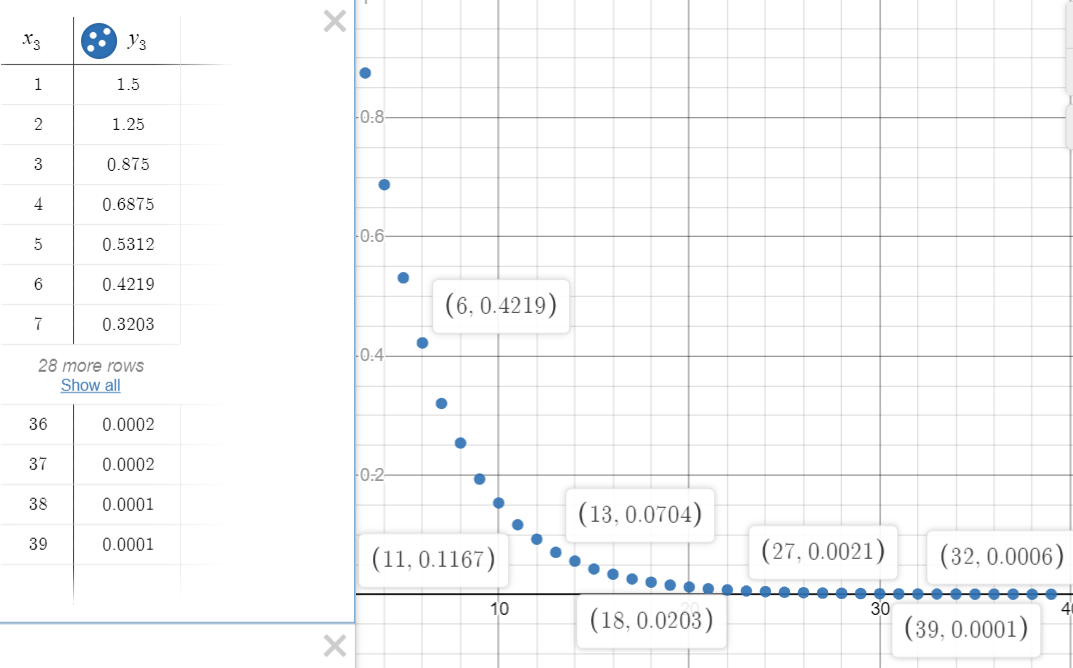
*−2y + 2z − w = 3*

*−z + 2w = 4*

When initial guess *(x0 , y0 , z0 , w0) = (1, 1, 1, 1)*



When initial guess is found using the formula *x0i = bi / (aii)*



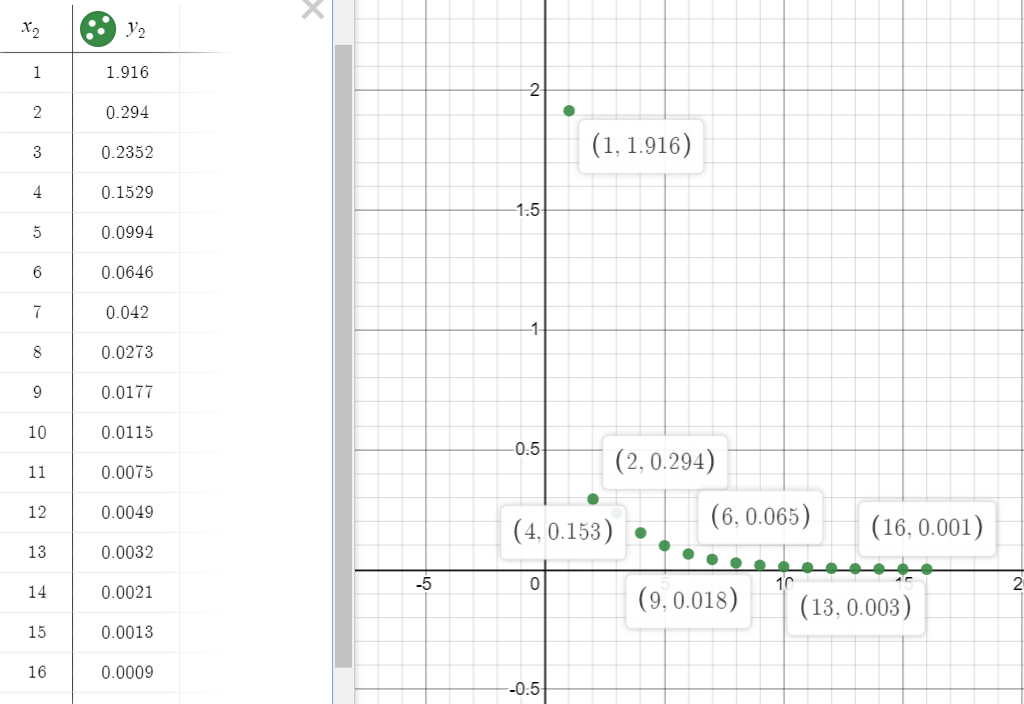
* ***Gauss - Seidel*** method for equations:

*10x − 8y= − 6*

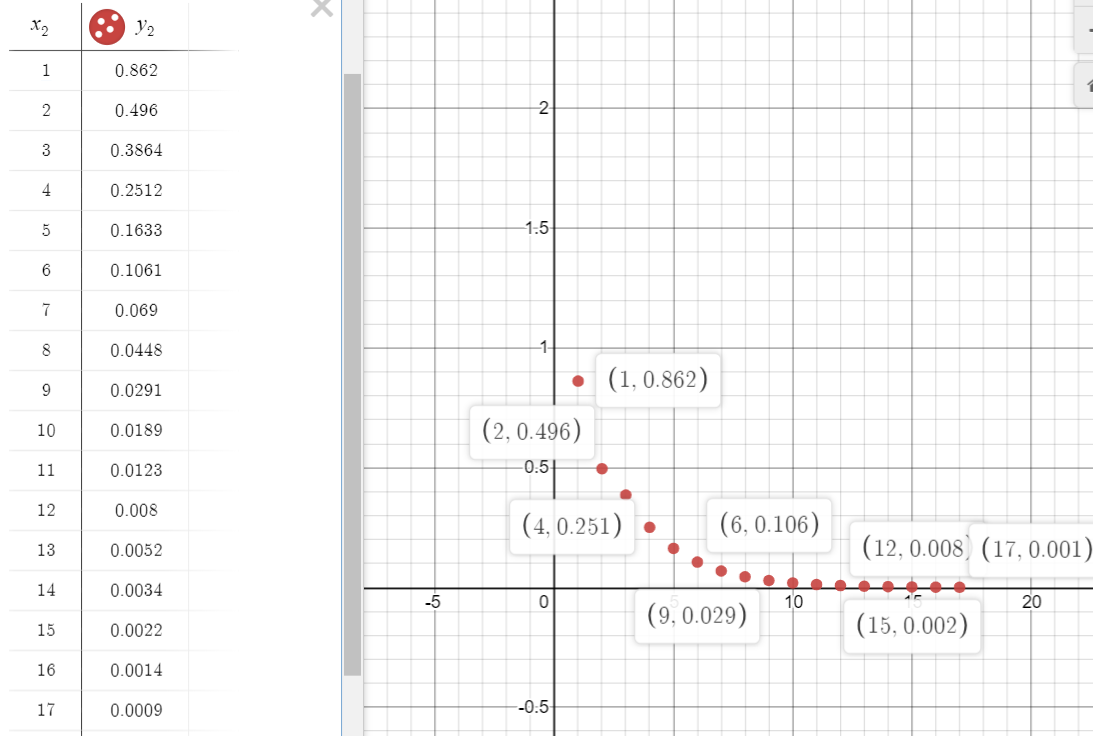
*−8x + 10y − z = 9*

*−y + 10z = 28*

When initial guess *(x0 , y0 , z0) = (1, 1, 1)*



When initial guess is found using the formula *x0i = bi / (aii)*



* ***Gauss - Jacobi*** method for equations:

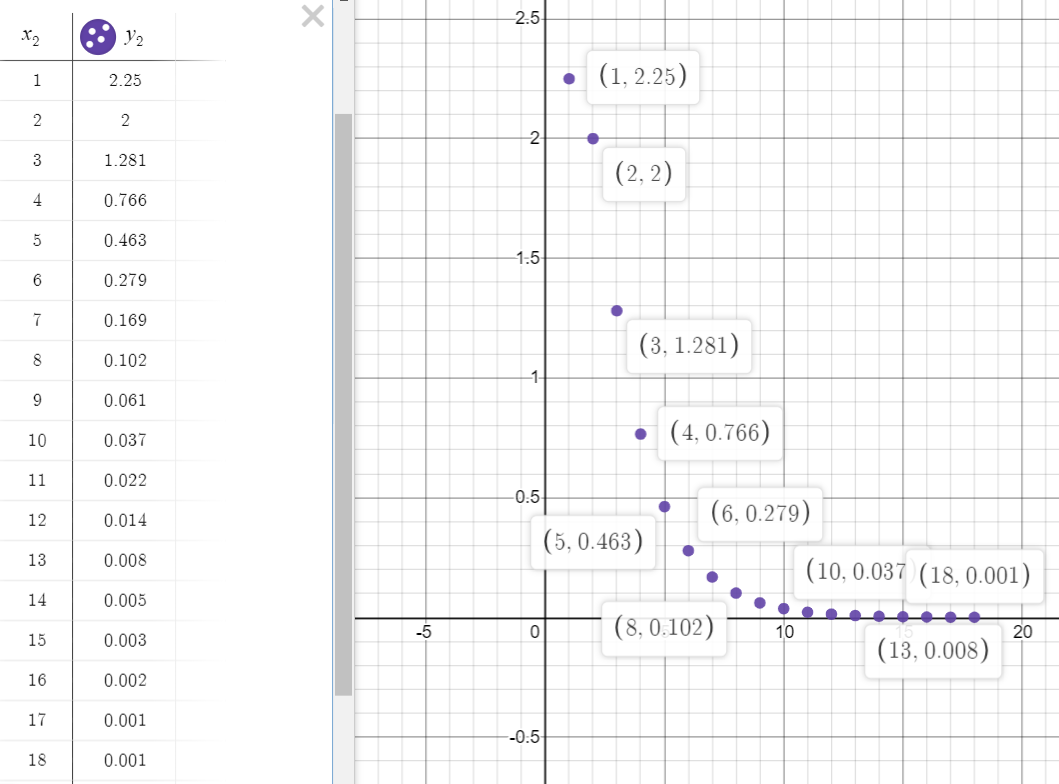
*2x − y= 1*

*x + 2y − z = 1*

*−2y + 2z − w = 3*

*−z + 2w = 4*

When initial guess *(x0 , y0 , z0 , w0) = (1, 1, 1, 1)*



When initial guess is found using the formula *x0i = bi / (aii)*

