Лабораторная работа №2

Тема: «Итерационные методы решения СЛАУ»

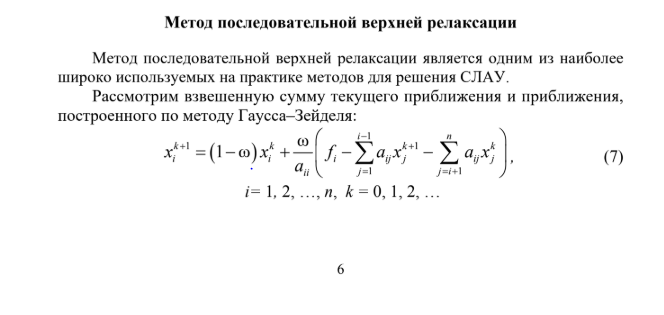
Самоховец Давид

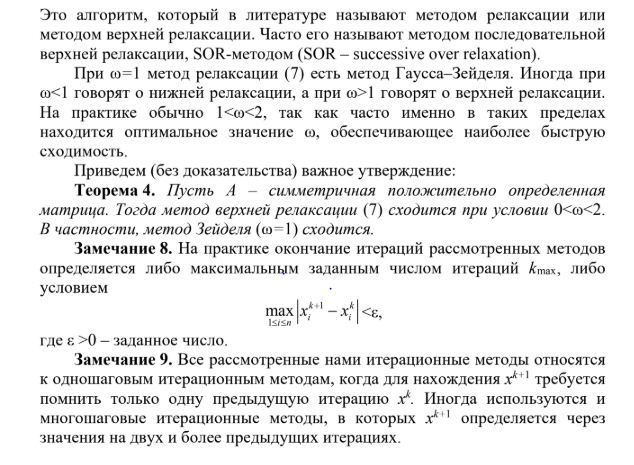
2 курс,1 группа

Постановка задачи

Разработать программу численного решения СЛАУ f = Ax методом релаксации, обеспечив сходимость итерационного процесса. В качестве критерия остановки итерационного процесса использовать ( 1) ||x(k+1) – x(k)|| , где  = 10-5 . Для проведения вычислительного эксперимента необходимо решить систему размерности n 10 . Матрицу A и вектор точного решения x заполнить случайными числами (сгенерировать) с двумя знаками после запятой из диапазона от -10 до 10. Правую часть задать умножением матрицы A на вектор x : f = Ax . В результатах выполнения вычислительного эксперимента необходимо привести следующую информацию: Матрицу A (построчно), вектор f , точное решение x ,  . Исследовать сходимость метода релаксации в зависимости от параметра релаксации .

Краткие теоретические сведения





Листинг программы

Генерируем матрицу А, вектор X. Вычисляем точное решение.

double GenNumb()

{

return (double) ((rand() % 2000) - 1000) / 100.;

}

void Gen( double A[N][N+1], double X[N] )

{

srand(time(NULL));

double B[N][N+1];

int i, j;

for (i=0; i<N; i++){

for (j = 0; j<N; j++){

B[i][j] = GenNumb() / (double)N;

A[i][j] = 0.;

}

X[i] = GenNumb();

}

for (i=0; i<N; i++)

for (j=0; j<N; j++)

for (int k=0; k<N; k++)

A[i][j] += B[i][k] \* B[j][k];

for (i=0; i<N; i++){

A[i][N] = 0.;

for ( j=0; j<N; j++ )

A[i][N] += A[i][j] \* X[j];

}

}

Метод релаксации:

void Solve(double A[N][N+1], double X[N], double Y[N], double w, ofstream& out)

{

int i;

int j;

double Yprev[N];

for (i=0; i<N; i++)

Y[i] = Yprev[i] = 1.;

int cnt = 0;

do

{

for (i=0; i<N; i++)

Yprev[i] = Y[i];

for (i=0; i<N; i++){

Y[i] = (1. - w) \* Yprev[i];

double sum = A[i][N];

for (j=0; j<i; j++)

sum -= A[i][j] \* Y[j];

for (j=i+1; j<N; j++)

sum -= A[i][j] \* Yprev[j];

Y[i] += sum \* (w / A[i][i]);

}

cnt ++;

}

while (MaxNorm(Y, Yprev) > Eps);

out.setf(ios::fixed);

out.precision(1);

out << w << " ";

out << cnt << " ";

out.precision(9);

out << MaxNorm(Y, Yprev) << " ";

out << MaxNorm(Y, X) << endl;

return;

}

Максимум норма:

double MaxNorm( double X[N], double Y[N] )

{

int i;

double ans = 0.;

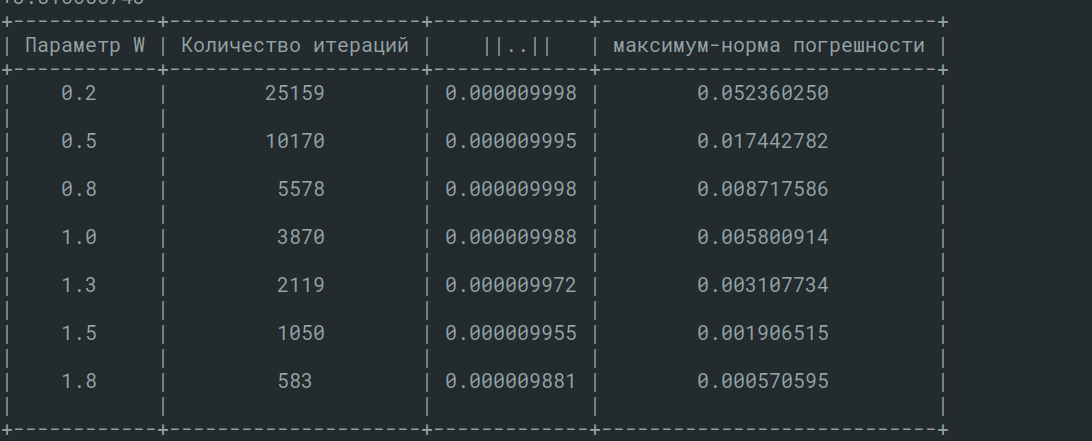
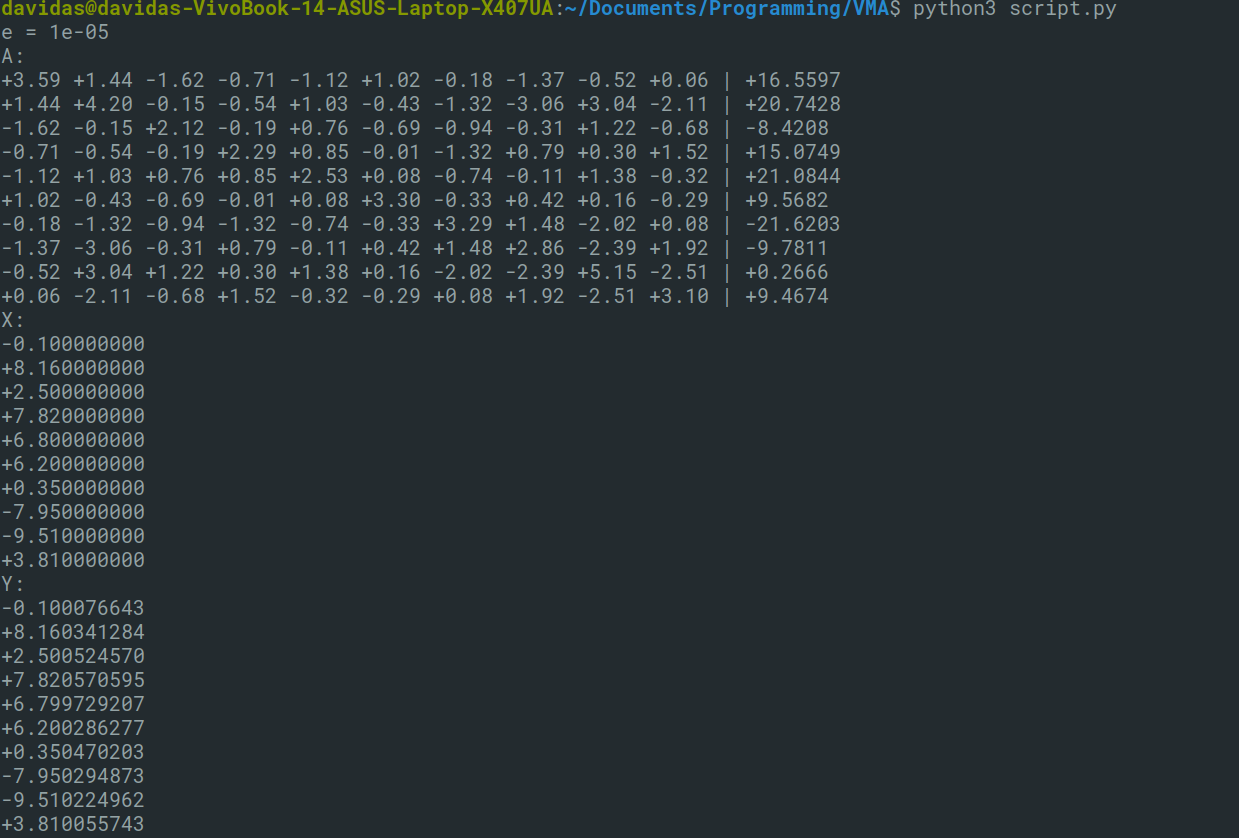
for ( i=0; i<N; i++ )

ans = max( ans, fabs(X[i] - Y[i]) );

return ans;

}

Результаты



|  |  |  |
| --- | --- | --- |
| Параметр  | Количество итераций k 1 , за которое была достигнута требуемая точность | ||x(k+1) – x(k)|| |
| 0.2 | 25159 | 9.99e-06 |
| 0.5 | 10170 | 9.99e-06 |
| 0.8 | 5578 | 9.99e-06 |
| 1 | 3870 | 9.98e-06 |
| 1.3 | 2119 | 9.97e-06 |
| 1.5 | 1050 | 9.95e-06 |
| 1.8 | 583 | 9.88e-06 |

Вывод:

Метод последовательной релаксации является одним из наиболее эффективных и широко используемых итерационных методов для решения систем линейных алгебраических уравнений с симметричными положительно определенными матрицами А. Плюсом метода, безусловно можно назвать его простоту в понимании и реализации. Минусом можно назвать его нестабильность, т.к. для разных систем линейных уравнений метод будет работать за разное количество итераций. Также при различных значениях параметра w метод сходится за разное количество итераций.