Летняя школа 2016

Реализация метода обобщенных минимальных невязок (GMRES) для системы фрагментированного программирования LuNA

Шкорин Михаил КазНу им. Аль-Фараби механико-математический факультет 2 курс

Руководители: Городничев М. А, Киреев С. Е, Перепёлкин В. А, Лебедев Д.В.

15.07.2016

План доклада

- Постановка задачи
- Идея решения
- Реализация
- Заключение
- Дополнительные слайды

Цель работы

Текущий прогресс	Последов.	MPI	LuNA
GMRES	Требуется доработка		
BiCG	Не устойчив	Не устойчив	Не устойчив
BiCGStab			

Есть реализация — Текущая задача — Не разобрано

Метод обобщенных минимальных невязок

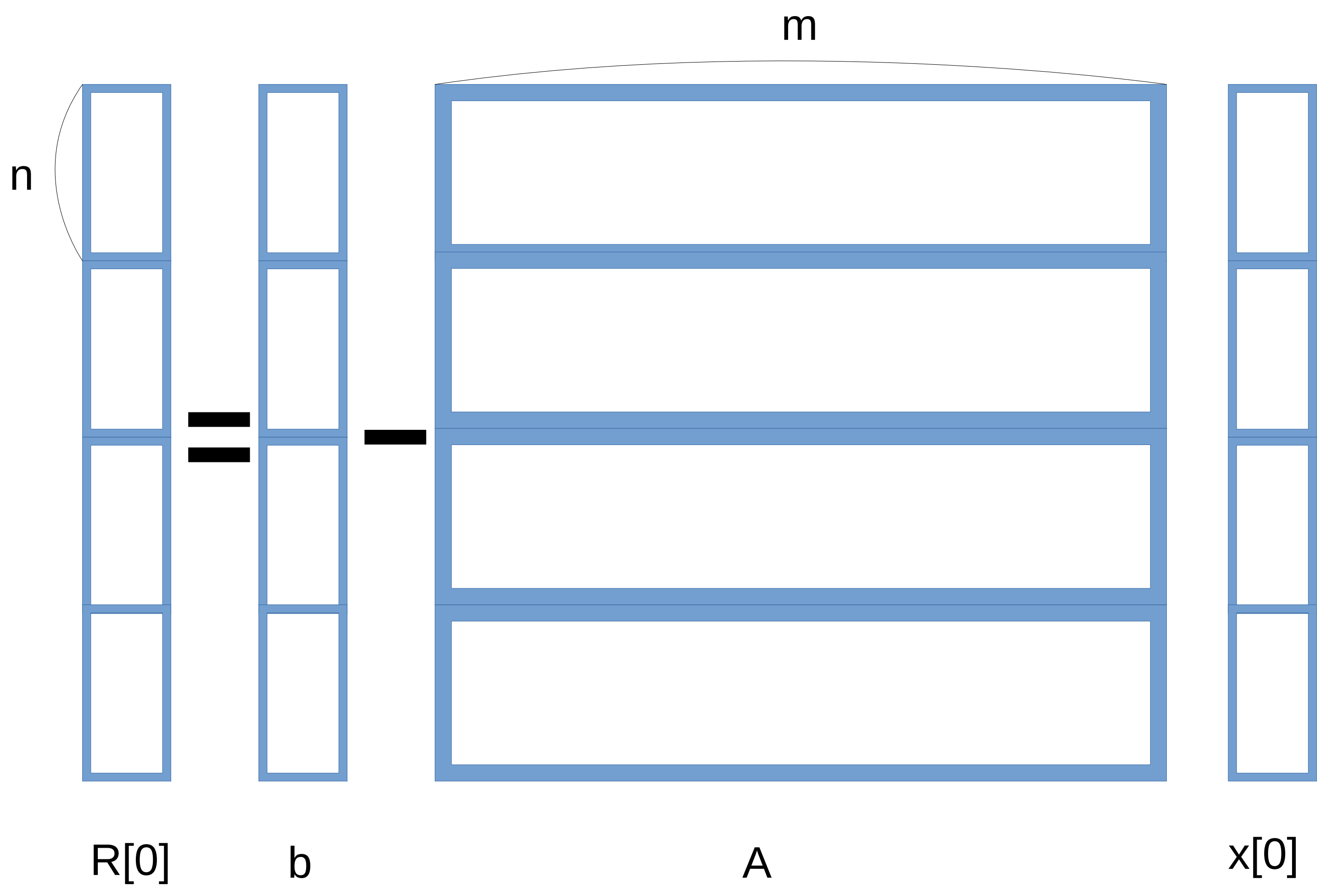
Метод обобщенных минимальных невязок (GMRES - Generalized minimal residual method) является итерационным методом численного решения несимметричных систем линейных уравнений вида.

Метод приближает решение вектором в подпространстве Крылова с минимальной невязкой. Чтобы найти этот вектор используется итерация Арнольди.

Algorithm 0: GMRES(m)

ϵ is the tolerance for the residual norm ;
convergence = false ;
choose x_0 ;
until convergence **do**
  $r_0 = b - Ax_0$;
  $\beta = \|r_0\|$;
  * Arnoldi process: construct a basis V_m of the Krylov subspace K
  $v_1 = r_0/\beta$;
  for $j = 1, \cdots, m$ **do**
    $p = Av_j$;
    for $i = 1, \cdots j$ **do**
      $h_{ij} = v_i^T p$;
      $p = p - h_{ij} v_i$;
    endfor ;
    $h_{j+1,j} = \|p\|_2$;
    $v_{j+1} = p/h_{j+1,j}$;
  endfor ;
  * Minimize $\|b - Ax\|$ for $x \in x_0 + K$
  * Solve a least-square problem of size m
  compute y_m solution of $\min_{y \in \mathcal{R}^m} \|\beta e_1 - \overline{H_m} y\|$;
  $x_m = x_0 + V_m y_m$;
  if $\|b - Ax_m\| < \epsilon$ convergence = true ;
  $x_0 = x_m$;
enddo

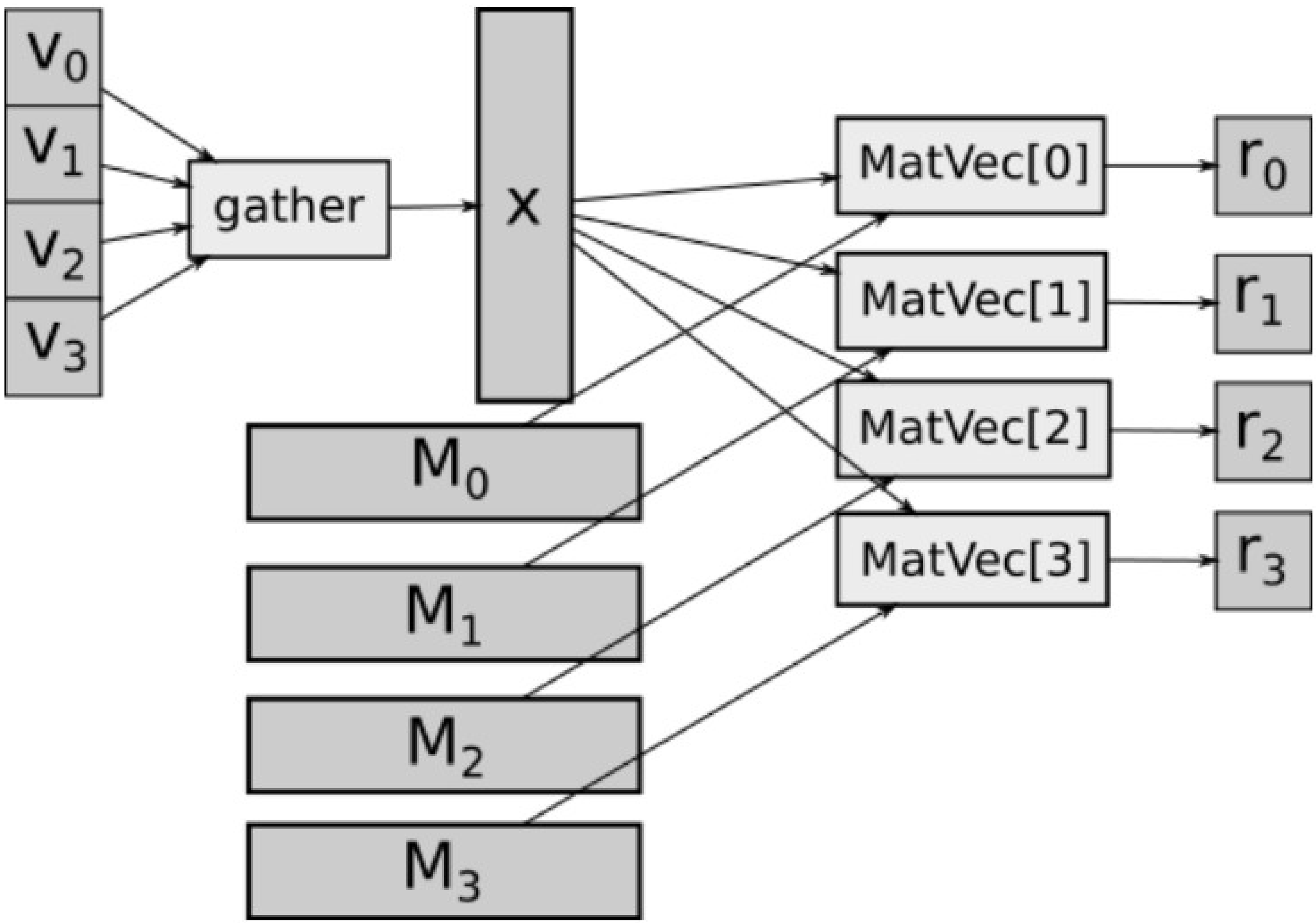
Разделение данных



Структура реализации

- Gmres.fa — основной файл запуска программы
- Imports.inc — файл импортирования атомарных подпрограмм в fa файл.
- Utils.fa — подпрограммы алгоритма
- Ucodes.cpp — последовательные атомарные подпрограммы

Граф вычислений подзадачи умножения матрицы на вектор



Реализация на языке LuNA

Из файла utils.fa :

```
sub mv(int m, int  n , name v, name Mat, name result)
{
    df vGathered;
    gather(v,vGathered,n);
    for rank = 0..$FG_CNT-1{
        matVec(Mat[rank],vGathered,m,n,result[rank]);
    }
}
```

c_matVec из ucodes.cpp

```
void c_matVec(InputDF &A_in, InputDF &v_in, int  m ,int  n, OutputDF &result_out){
    result_out.create(n*sizeof(double));
    double * result = result_out.getData<double>();
    const double * MA = A_in.getData<double>();
    const double * v = v_in.getData<double>();
    for (int i=0; i<n; i++){
        result[i]=0;
        for (int k=0; k<m; k++){
            result[i]+= MA(i,k)*v[k];
        }
    }
}
```

Заключение

За время летней школы :

- Был разработан поход к фрагментированной реализации алгоритма GMRES.
- Были разработаны алгоритмы и реализации подзадач: gather, умножение матрицы на вектор, reduceSum, вычисление нормы вектора

Дальнейшие цели

- Завершить реализацию фрагментированной программы GMRES(m).
- Разработать реализацию параллельного GMRES(m) используя MPI
- Провести сравнение двух реализаций
- Изучить модификации алгоритма GMRES избегающие глобальные коммуникации (NewtonGMRES)

Gmres.fa

```
##include "imports.inc"
##include "utils.fa"
##const FG_CNT 4
##const TOL 0.0000001
##const MAX_ITER 10000
##const FILENAME "16.txt"
sub main(){
    df A, b, x, m, n, R,normR;
    for rank=0..$FG_CNT-1    {
        cf initData[rank]: readData($FILENAME,$FG_CNT,rank,m,n,A[rank],b[rank],x[0][rank]);
    }
    v_minus_Mv(m,n,b,A,x[0],R[0]);
    norm(n,R[0],normR[0]);
    ...
}
```

Подпространство Крылова

В линейной алгебре подпространством Крылова размерности m, порождённым вектором v ∈ C и матрицей $A \in C^{n \times n}$, называется линейное пространство

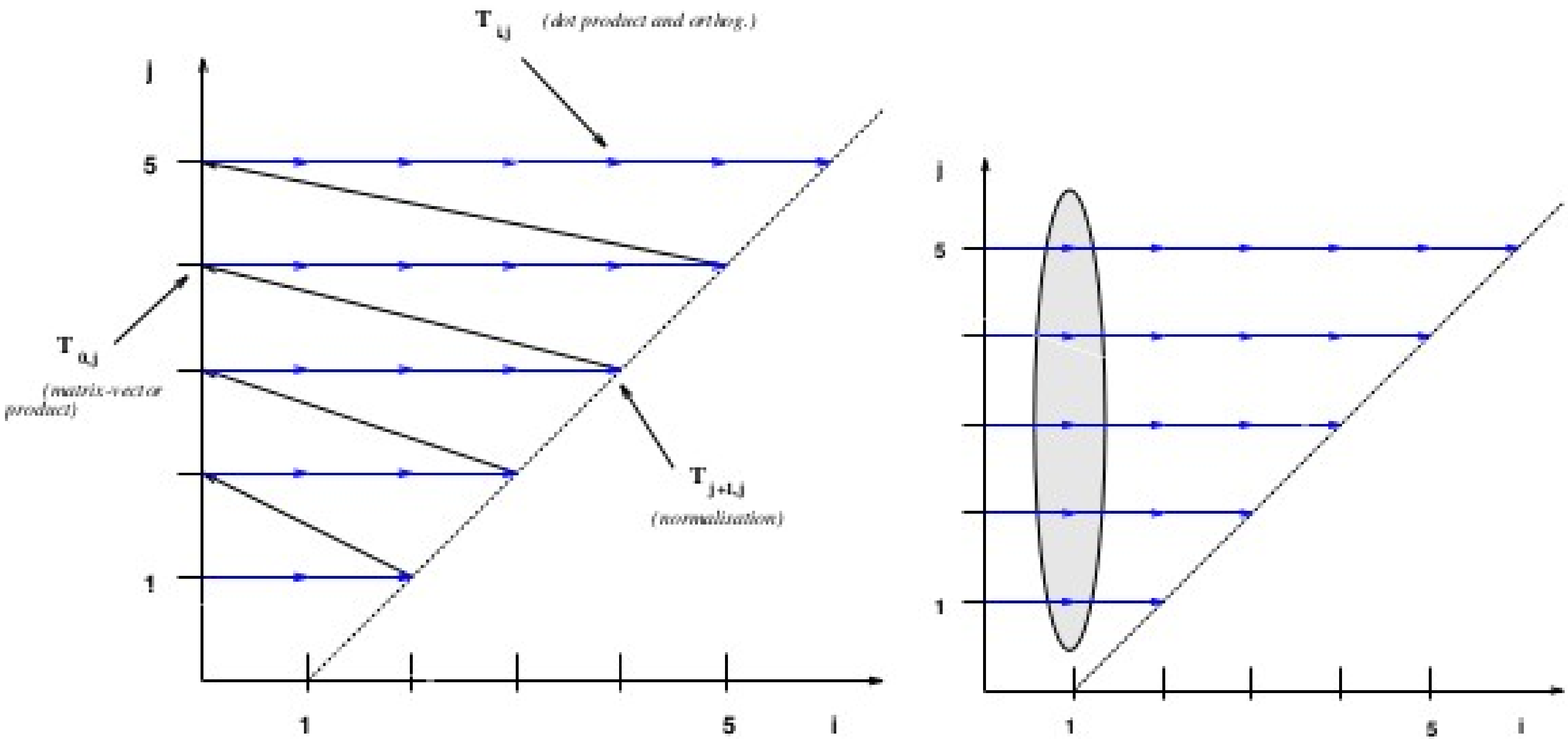
$$K_m = K_m(A, v) = span\left\{v, Av, A^2v, \ldots, A^{m-1}v\right\}.$$

Любой вектор x из подпространства $x_0 + K_m$ может быть представлен в форме

$$x = x_0 + V_m y,$$

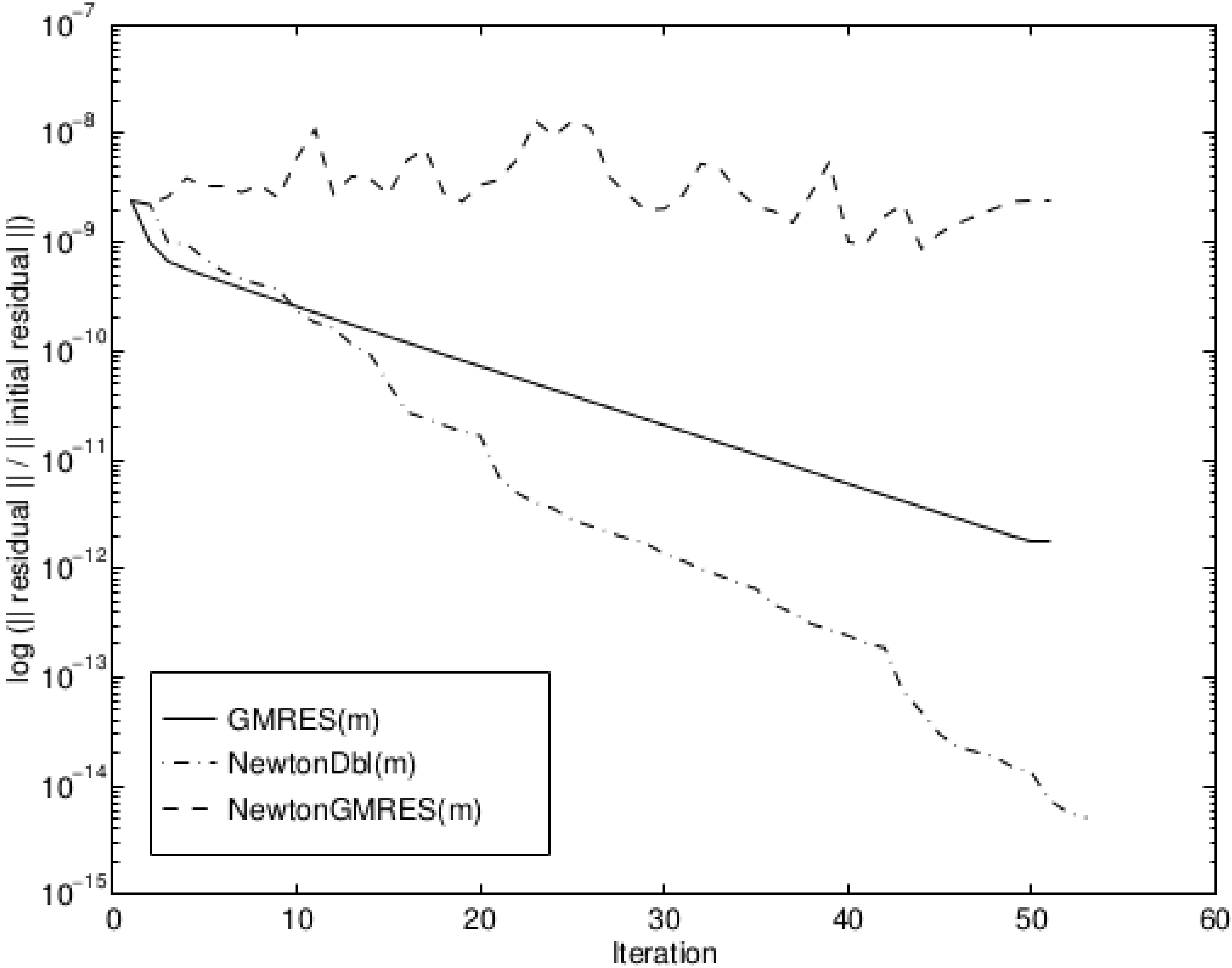
где V_m – матрица, составленная из векторов ортонормированного базиса подпространства Крылова K_m, а y – вектор размерности m.

График зависимостей итераций Арнольди

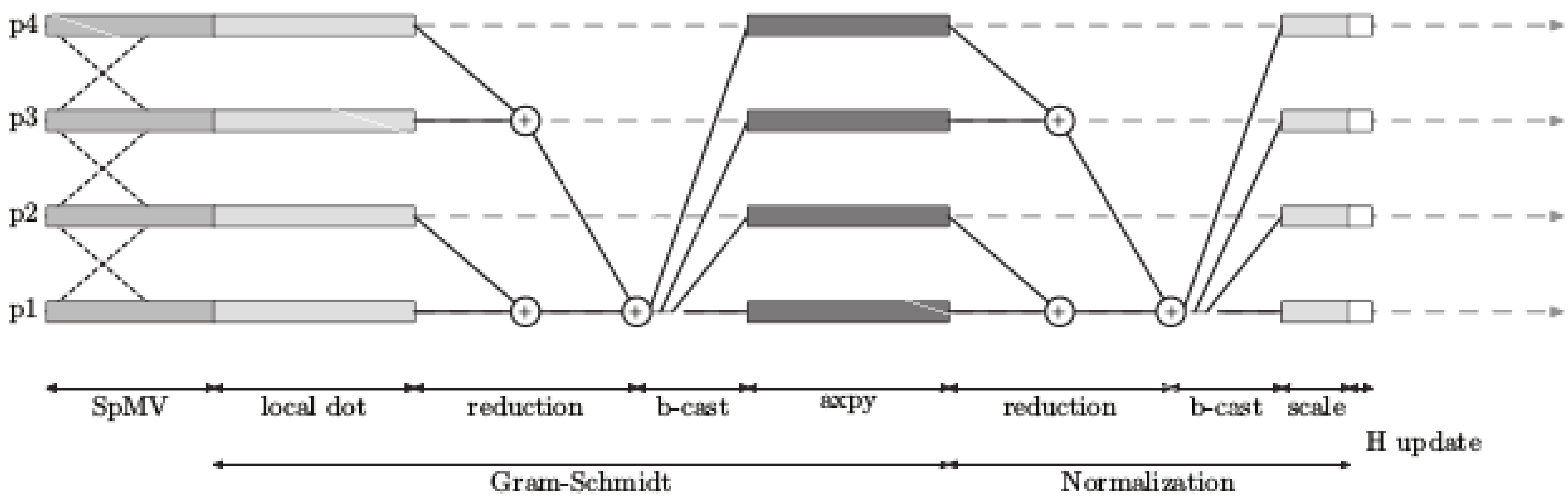


Arnoldi-MGS Newton-Arnoldi

Сходимость на матрице WATT



Схематическое представление одной итерации стандартного GMRES алгоритма



start:

x_0= initial guess; $r_0 = b - Ax_0$;

$v_1 = r_0/\|r_0\|_2$;

iterate:

for $j = 1, 2, \ldots, m$ do

$\hat{v}_{j+1} = Av_j$;

for $i = 1, 2, \ldots, j$ do

$h_{i,j} = (\hat{v}_{j+1}, v_i)$;

$\hat{v}_{j+1} = \hat{v}_{j+1} - h_{i,j}v_i$;

end;

$h_{j+1,j} = \|\hat{v}_{j+1}\|_2$;

$v_{j+1} = \hat{v}_{j+1}/h_{j+1,j}$;

end

form the approximate solution:

$x_m = x_0 + V_m y_m$; where

y_m minimizes $\| \|r_0\|_2 e_1 - \bar{H}_m y\|_2$, $y \in I\!R^m$

restart:

compute $r_m = b - Ax_m$; if satisfied then

stop, else $x_0 = x_m$; $v_1 = r_m/\|r_m\|_2$;

goto *iterate*

Fig. 2. The GMRES(m) algorithm.