Программа для рассчета схемы Ричардсона.

Схема Ричардсона является одной из хорошо известных явных двухслойных итерационных схем построения решения системы линейных алгебраических уравнений с самосопряженым знакоопределенным оператором. Преимуществом схемы является её простота и возможность эффективного распараллеливания. При этом применение чебышевского ускорителя существенно повышает скорость сходимости. Однако в классической формулировке схема Ричардсона предполагает знание границ спектра матрицы. Оценка верхней границы спектра не представляет трудностей, т.к. в этом случае применяется теорема о кругах Гершгорина.

В данной программе предлогаетя алгоритм эффективной настройки на нижнюю границу спектра задачи без привязки к типу расчетной области или используемой сетки. Алгоритм построен на одновременной работе двух конкурирующих итерационных процессов, эффективность которых постоянно анализируется.

1.1 Расчет итогового значения вектора y для еденичной матрицы и без конкурирующих процессов.

```
computeResultVectorForE (vector<double> &y, RichardsonSLAU *SLAU, vector<double> f,int fold)
```

y — нуевое значение вектора y (тип vector <double>)
SLAU — матрица A. Тип SLAU. Тип слау можно описать 2 способами:1) обычным формате (vectot <vector <double> >); 2) Йельский формат (вектор pointer <int> - сколько цыфр в строке, вектор cols <int> - номера столбцов, вектор values <double> - сами числа)
f — вектор f(тип vector <double>)
fold — кратность графиков итерации, которые нужно выводить, fold=0-ни одного, fold=1-(iterationNomber-1)-графики кратные fold, fold=iterationNomber -только последнюю итерацию;

1.2 Расчет итогового значенияя вектора с канкурирующими процессами для единичной матрицы.

```
computeResultVectorForEWithRivalProcess(vector<double> &y, RichardsonSLAU *SLAU,vector<double> f,int fold)
```

y — нуевое значение вектора y (тип vector <double>)
SLAU — матрица A. Тип SLAU. Тип слау можно описать 2 способами:1) обычным формате (vectot <vector <double> >); 2) Йельский формат (вектор pointer <int> - сколько цыфр в строке, вектор cols <int> - номера столбцов, вектор values <double> - сами числа)
f — вектор f(тип vector <double>)
fold — кратность графиков итерации, которые нужно выводить, fold=0-ни одного, fold=1-(iterationNomber-1)-графики кратные fold, fold=iterationNomber -только последнюю итерацию;

1.3 Расчета итогового значенияя вектора с канкурирующими процессами для не единичной матрицы

```
computeResultVectorForNotEWithRivalProcess(vector<double> &y, RichardsonSLAU *SLAU,vector<double> f,int fold)
```

y — нуевое значение вектора y (тип vector <double>)
SLAU — матрица A. Тип SLAU. Тип слау можно описать 2 способами:1) обычным формате (vectot <vector <double> >); 2) Йельский формат (вектор pointer <int> - сколько цыфр в строке, вектор cols <int> - номера столбцов, вектор values <double> - сами числа)

f — вектор f(тип vector <double>)
fold — кратность графиков итерации, которые нужно выводить, fold=0-ни одного, fold=1-(iterationNomber-1)-графики кратные fold, fold=iterationNomber -только последнюю итерацию;