**Алгоритм «приведение матрицы Хаусхолдера к двудиагональной форме».**

Работа выполнена: студент КМБО-01-20, Мустафин Рамиль

Почта: [mrr201702@mail.ru](mailto:mrr201702@mail.ru)

Ник в ТГ: @vinramil

В советской математической литературе метод приведения матрицы Хаусхолдера к двудиагональной форме чаще называется методом отражений. Сама же двудиагональная форма называется так же «бидиагональной».

На вход алгоритма поступает матрица А, числа m и n, такие, что матрица А размера m×n.

На выходе ожидаются матрицы B, V и U такие, что матрица В – верхняя бидиагональная, U и V являются результатом матрицы Хаусхолдера, где А=UBVT.

Опишем пошагово алгоритм:

1. B ← A (Пропустим этот шаг, если А должно быть перезаписано на В)
2. U = Im×n. (Создадим матрицу U размера m×n)
3. V = In×n. (Создадим матрицу V размера n×n)
4. Определим матрицу Хаусхолдера Qk (для k = 1 ,…, n) со следующим свойством: умножение слева столбца на матрицу Qk оставляет компоненты 1 ,…, k – 1 неизменными, причём

Qk =, где s = ±.

1. Переопределим (для k = 1 ,…, n) матрицу В

B ← Qk B.

1. Переопределим (для k = 1 ,…, n) матрицу U

U ← U Qk.

1. Если k ≤ n – 2, то определим матрицу Хаусхолдера ( для k = 1 ,…, n) Pk+1 со следующим свойством: умножение справа строки на матрицу Pk+1 оставляет компоненты 1 ,…, k неизменными, причём

Pk+1 = , где s = ±

1. Переопределим (для k = 1 ,…, n) матрицу В

B ← B Pk+1.

1. Переопределим (для k = 1 ,…, n) матрицу V

V ← Pk+1 V.

**Алгоритм «Голуба-Кахана».**

На вход алгоритма поступает матрица А, числа m и n, такие, что матрица А размера m×n.

На выходе ожидаются матрицы Σ, V и U такие, что матрица Σ – диагональная, U и V имеют ортонормированные столбцы, размеры матриц U=m×n, V=n×n, где А=UΣVT.

Опишем пошагово алгоритм:

1. Применим алгоритм «приведение матрицы Хаусхолдера  к двудиагональной форме» чтобы получить матрицы B, V и U такие, что матрица В – верхняя бидиагональная, U и V являются результатом матрицы Хаусхолдера, где А=UBVT.

Повторять пункты 2) - 4):

1. Если |bi,i+1| ≤ ε (|bi,i| + |bi+1,i+1|) (для i = 1, … , n – 1), то

а) bi,i+1 = 0.

б) Определим наименьшее p и наибольшее q такие, чтобы матрицу B можно было представить в виде

Где матрица диагональная, а матрица не имеет нулевую супердиагонального (диагональ выше главной диагонали) элемента.

1. Если q=n, то матрица Σ = диагональной части матрицы B. Выходим из цикла.
2. Если bi,i = 0 (для i = p+1, … , n – q – 1), то применяем метод вращений Гивенса так, чтобы bi,i+1 = 0 и всё ещё оставалась верхней бидиагональной.

Иначе применим алгоритм «шаг алгоритма Голуба-Кахана» к n, B, U, V, p, q.

**Алгоритм «шаг алгоритма Голуба-Кахана».**

На вход алгоритма поступают, числа n, p, q, матрицы B, Q, P такие, что матрица B размера n×n является верхней бидиагональной, Q и P имеют ортогональные столбцы, а матрица A = QBPT.

На выходе ожидаются матрицы B, Q и P такие, что матрица В – верхняя бидиагональная, Q и P имеют ортогональные столбцы, а недиагональные элементы выходной матрицы B меньше, чем недиагональные элементы входной матрицы. (Матрицы B, Q и P перезаписываются в хранилище)

Опишем пошагово алгоритм:

1. Введём матрицу B2,2, которая является диагональным блоком матрицы В с номерами строки и столбца p + 1, … , n – q.
2. Найдём матрицу .
3. Введём С такой, что C – нижняя правая подматрица размером 2×2 матрицы B2,2.
4. Найдём собственные значения λ1, λ2 подматрицы С.
5. Из чисел λ1, λ2 найдём то, что ближе к элементу с2,2 матрицы С.
6. Введём μ = найденному числу п.5.
7. Введём k = p + 1.
8. Введём α = – μ.
9. Введём β = bk,k bk,k+1.

Все последующие шаги выполнять при k = p + 1, … , n – q – 1.

1. Введём с = cos(θ) и s = sin(θ) такие, что .
2. Введём матрицу Rk,k+1(c,s) – матрица вращение Гивенса, действующая на столбцы k и k + 1 во время умножения справа.
3. Переопределим B ← B Rk,k+1(c,s).
4. Переопределим P ← P Rk,k+1(c,s).
5. Приравняем α = bk,k, β = bk+1,k.
6. Приравняем с = cos(θ) и s = sin(θ) так, что .
7. Введём матрицу Rk,k+1(c,–s) – матрица вращение Гивенса, действующая на столбцы k и k + 1 во время умножения слева.
8. Переопределим B ← Rk,k+1(c,–s) B.
9. Переопределим Q ← Q Rk,k+1(c,s).
10. Если k ≤ n – q – 1, то приравняем

α = bk,k+1, β = bk,k+2.