

eps=1E-03		QR-разложение			
		с формой Хессенберга		без формы Хессенберга	
Сдвиги	+	всего итераций:	8	всего итераций:	8
		всего вращений:	20	всего вращений:	34
		точность:	4.25E-09	точность:	4.25E-09
	-	всего итераций:	14	всего итераций:	8
		всего вращений:	29	всего вращений:	40
		точность:	0.0017	точность:	0.0017

eps = 1E-04		QR-разложение			
		с формой Хессенберга		без формы Хессенберга	
Сдвиги	+	всего итераций:	9	всего итераций:	8
		всего вращений:	22	всего вращений:	34
		точность:	4.25E-09	точность:	4.25E-09
	-	всего итераций:	35	всего итераций:	35
		всего вращений:	102	всего вращений:	202
		точность:	7.76E-05	точность:	7.76E-05

eps = 1E-06		QR-разложение			
		с формой Хессенберга		без формы Хессенберга	
Сдвиги	+	всего итераций:	10	всего итераций:	9
		всего вращений:	25	всего вращений:	35
		точность:	4.25E-09	точность:	4.25E-09
	-	всего итераций:	43	всего итераций:	43
		всего вращений:	126	всего вращений:	250
		точность:	6.97E-07	точность:	6.97E-07

Замечания

QR-алгоритм выполнен с редукцией размерности матрицы. Переход к редукции осуществляется по достижению стабилизации элемента $a(i,i)$ матрицы A с требуемой точностью, где $a(i,i)$ - первый диагональный элемент в случае алгоритма без сдвигов ($i=0$) и последний диагональный элемент в случае алгоритма с произвольными сдвигами ($i=k$). В качестве коэффициентов сдвигов в таком случае как раз берем элементы $a(k,k)$.

В данных таблицах отсутствуют сведения о количестве сдвигов ввиду того, что их число в нашей программной реализации всегда равно числу итераций ($2 \cdot \text{число итераций}$, если считать сдвиг и обратный сдвиг на одной итерации отдельно - как два сдвига).

QR-алгоритм с приведением к форме Хессенберга, как и следовало ожидать, значительно сокращает число вращений (почти в 2 раза), ведь в матрице Хессенберга элементы ниже поддиагональных уже не требуется приводить к нулю.

QR-алгоритм со сдвигами быстро (8-10 итераций) достигает решения приближенного к точному, даже если установленная точность была слабее. В случае отсутствия сдвигов наблюдаем закономерный рост числа итераций при увеличении точности.

Сходимость алгоритма обратных итераций напрямую зависит от точности вычисленных ранее собственных значений. При достаточно точном определении собственных значений сходимость алгоритма высокая (1-2 итерации). Однако если погрешность собственных значений значительна, то алгоритм не сойдется при использовании предложенного критерия останова, так как он напрямую зависит от значения собственного числа. То есть даже если собственный вектор уже найден достаточно точно, критерий останова не выполнится, ввиду значительной погрешности собственного числа. Имеет смысл использовать альтернативный критерий останова для устранения подобных проблем: оценка стабилизации вычисляемого собственного вектора (например, сферическая норма от разности предыдущего и текущего приближений к собственному вектору). Реализованный алгоритм обратных итераций, как предполагалось теоретически, сходится при любых начальных приближениях.

Комбинированный метод обратных итераций с использованием соотношения Релея также подтвердил теоретические предположения.

К примеру, взяв в качестве первого приближения вектор $(-0.6, 0.3, -0.1, -0.2)$, мы получили собственное значение 0.997313 и соответствующий ему собственный вектор $(0.86446047, 0.00339321, 0.24459138, 0.43917153)$. То есть в данном случае первое приближение наиболее соответствовало именно этому собственному вектору, нежели остальным. Отклонив первое приближение до $(-0.4, 0.4, 0.1, -0.2)$, получим собственное значение 2.00425 и соответствующий ему собственный вектор $(-0.01209917, 0.70977426, -0.60747029, 0.35644628)$.