ВЕКТОРНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ НА ЯЗЫКЕ MATLAB*

О. В. Просеков

sc2@pisem.net

28 сентября 2004 г.

Будут использоваться следующие обозначения:

$A_n = A_n[0:n-1, 0:n-1]$ — квадратная матрица порядка n, индексы строк и столбцов которой изменяются от 0 до $n-1$;

$I_n = I_n[0:n-1, 0:n-1]$ — единичная матрица порядка n.

1°. Пусть m, n — натуральные числа. Введём квадратную матрицу $L_{mn}^{(n)}$ порядка mn с элементами

$$L_{mn}^{(n)}[i + j\,m, i'\,n + j'] = \begin{cases} 1\,, & \text{если } i' = i \text{ и } j' = j, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Здесь $i, i' \in 0 : m-1$, $j, j' \in 0 : n-1$. Из определения следует, что

$$L_n^{(n)} = I_n\,, \qquad L_m^{(1)} = I_m\,.$$

Нетрудно проверить, что

$$\left(L_{mn}^{(n)}\right)^{-1} = \left(L_{mn}^{(n)}\right)^T = L_{mn}^{(m)}\,.$$

*Работа поддержана грантом Федерального агентства по образованию А04–2.8–415

Семинар по дискретному гармоническому анализу и геометрическому моделированию «DHA & CAGD»: `http://www.dha.spb.ru/`

Умножение матрицы $L_{mn}^{(n)}$ на вектор $x \in \mathbb{C}^{mn}$ порождает вектор $X = L_{mn}^{(n)}\, x$ с компонентами

$$X(i + j\, m) = x(i\, n + j)\,, \qquad i \in 0 : m-1, \quad j \in 0 : n-1\,. \tag{1}$$

По сути матрица $L_{mn}^{(n)}$ является оператором транспонирования двумерной матрицы, представленной (по столбцам) одномерным вектором. Программа соответствующая формуле (1) выглядит следующим образом:

X = **reshape**(x, [n m])
X = X .'
X = **reshape**(X, [m∗n 1])

2°. *Кронекеровым произведением* матриц A_m и B_n называется матрица $D_{mn} = A_m \otimes B_n$ с элементами

$$D_{mn}[i\, n + j, i'\, n + j'] = A_m[i, i']\, B_n[j, j']\,, \qquad i, i' \in 0 : m-1, \quad j, j' \in 0 : n-1\,.$$

Более наглядно

$$D_{mn} = \begin{bmatrix} A_m[0,0]\, B_n & A_m[0,1]\, B_n & \dots & A_m[0,m-1]\, B_n \\ A_m[1,0]\, B_n & A_m[1,1]\, B_n & \dots & A_m[1,m-1]\, B_n \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ A_m[m-1,0]\, B_n & A_m[m-1,1]\, B_n & \dots & A_m[m-1,m-1]\, B_n \end{bmatrix}.$$

Важным свойством кронекерова произведения является соотношение ([1])

$$A_m \otimes B_n = (A_m \otimes I_n)(I_m \otimes B_n) = (I_m \otimes B_n)(A_m \otimes I_n)\,.$$

Матрица $A_m \otimes I_n$ соответствует векторным вычислениям, а матрица $I_m \otimes B_n$ — параллельным.

Векторная форма записи: $X = (A_m \otimes I_n)\, x$:

$$\begin{bmatrix} X(j) \\ X(n+j) \\ \vdots \\ X\big((m-1)\, n + j\big) \end{bmatrix} = \begin{bmatrix} A_m[0,0]\, x(j) + \dots + A_m[0,m-1]\, x\big((m-1)\, n + j\big) \\ A_m[1,0]\, x(j) + \dots + A_m[1,m-1]\, x\big((m-1)\, n + j\big) \\ \vdots \\ A_m[m-1,0]\, x(j) + \dots + A_m[m-1,m-1]\, x\big((m-1)\, n + j\big) \end{bmatrix}.$$

Параллельная форма записи: $X = (I_m \otimes B_n)\, x$:

$$\begin{bmatrix} X(j) \\ X(n+j) \\ \vdots \\ X\big((m-1)\, n + j\big) \end{bmatrix} = \begin{bmatrix} B_n\, x(j) \\ B_n\, x(n+j) \\ \vdots \\ B_n\, x\big((m-1)\, n + j\big) \end{bmatrix}.$$

Здесь $j = 0 : n-1$. Программы вычислений выглядят следующим образом:

Векторная форма	Параллельная форма
X = **reshape**(x, [n m])	X = **reshape**(x, [n m])
X = X * A_m .'	X = B_n * X
X = **reshape**(X, [m*n 1])	X = **reshape**(X, [m*n 1])

Программа вычисления $X = (A_m \otimes B_n)\, x$ получается из двух предыдущих:

```
X = reshape(x, [n m])
X = B_n * X * A_m .'
X = reshape(X, [m*n 1])
```

Случай двух множителей в кронекеровом произведении является тривиальным. В следующем пункте представлен общий случай для s сомножителей.

3°. Пусть $N = n_1 n_2 \dots n_s$. Обозначим $\Delta_1 = 1$; $\Delta_\nu = n_1 n_2 \dots n_{\nu-1}$, $\nu = 2, \dots, s+1$; $N_\nu = N/\Delta_{\nu+1}$. Очевидно, что $N_0 = N$, $N_s = 1$ и $N_\nu = n_{\nu+1} \dots n_s$ при $\nu = 1, \dots, s-1$.

Пусть массив n содержит числа n_ν. Определим массивы Delta и N, содержащие числа Δ_ν и N_ν соответственно:

```
Delta(1) = 1
Delta(2 : s+1) = cumprod(n)
Nn = Delta(s+1)
N = Nn ./ Delta(2 : s+1)
```

Здесь переменная Nn равна N. Справедлива формула ([1])

$$A_{n_1} \otimes A_{n_2} \otimes \dots \otimes A_{n_s} = \prod_{\nu=1}^{s} \bigl(I_{\Delta_\nu} \otimes A_{n_\nu} \otimes I_{N_\nu}\bigr). \tag{2}$$

Более того, сомножители в правой части (2) можно переставлять в любом порядке. Пусть массив A содержит матрицы A_{n_ν}.

Справедливы разложения ([1])

$$A_{n_1} \otimes A_{n_2} \otimes \dots \otimes A_{n_s} = \prod_{\nu=1}^{s} \bigl(A_{n_\nu} \otimes I_{N/n_\nu}\bigr) L_N^{(n_\nu)}\,, \tag{3}$$

$$= \prod_{\nu=1}^{s} L_N^{(n_\nu)} \bigl(I_{N/n_\nu} \otimes A_{n_\nu}\bigr)\,. \tag{4}$$

Формула (3) соответствует векторной форме разложения (2), а (4) — параллельной. Приведём программы вычисления вектора $X = (A_{n_1} \otimes A_{n_2} \otimes \dots \otimes A_{n_s})\, x$:

Векторная форма

```
X = x
for nu = s : −1 : 1
    X = reshape(X, [n(nu) Nn/n(nu)])
    X = X .'
    X = X ∗ A{nu} .'
end
X = reshape(X, [Nn 1])
```

Параллельная форма

```
X = x
for nu = s : −1 : 1
    X = reshape(X, [n(nu) Nn/n(nu)])
    X = A{nu} ∗ X
    X = X .'
end
X = reshape(X, [Nn 1])
```

Для примера рассмотрим ещё один вариант из [1]:

$$A_{n_1} \otimes A_{n_2} \otimes \cdots \otimes A_{n_s} = \prod_{\nu=1}^{s} \bigl(I_{\Delta_\nu} \otimes L_{N_{\nu-1}}^{(n_\nu)}\bigr)\bigl(I_{N/n_\nu} \otimes A_{n_\nu}\bigr)\bigl(I_{\Delta_\nu} \otimes L_{N_{\nu-1}}^{(N_\nu)}\bigr). \qquad (5)$$

Соответствующая программа вычисления вектора $X = (A_{n_1} \otimes A_{n_2} \otimes \cdots \otimes A_{n_s})\, x$ записывается следующим образом:

```
X = x
for nu = s : −1 : 1
    X = reshape(X, [N(nu) n(nu) Delta(nu)])
    X = permute(X, [2 1 3])
    X = reshape(X, [n(nu) Nn/n(nu)])
    X = A{nu} ∗ X
    X = reshape(X, [n(nu) N(nu) Delta(nu)])
    X = permute(X, [2 1 3])
end
X = reshape(X, [Nn 1])
```

В этой программе первые две строчки в цикле соответствуют матрице $I_{\Delta_\nu} \otimes L_{N_{\nu-1}}^{(N_\nu)}$, которая представляет собой транспонирование в трёхмерном массиве первых двух размерностей.

4°. Рассмотрим вопрос о вычислении дискретного преобразования Фурье (ДПФ)

$$X = F_N\, x\,.$$

Здесь F_N — матрица ДПФ с элементами

$$F_N[k,l] = \omega_N^{-kl}\,, \qquad k,l \in 0 : N-1\,,$$

где $\omega_N = \exp(2\pi i/N)$ — корень N-й степени из единицы.

Нам потребуются две специальные матрицы — матрица перестановок $L_{mn}^{(n)}$ (определённая раньше) и диагональная матрица вращений $T_{mn}^{(m)}$. Матрица $T_{mn}^{(m)}$

определяется так:

$$T_{mn}^{(m)}[i + j\,m, i' + j'\,m] = \begin{cases} \omega_{mn}^{-ij}\,, & \text{если } i' = i \text{ и } j' = j, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Здесь $i, i' \in 0 : m-1$, $j, j' \in 0 : n-1$. Из определения следует, что

$$T_n^{(1)} = I_n\,, \qquad T_m^{(m)} = I_m\,.$$

Справедлива формула ([2])

$$F_{mn} = \big(F_m \otimes I_n\big) T_{mn}^{(n)} \big(I_m \otimes F_n\big) L_{mn}^{(m)}\,.$$

Программа вычисления вектора $X = F_{mn}\,x$ записывается так:

```
X = reshape(x, [m n])
X = X .'
X = F_n * X
X = X .* T_n_m
X = X * F_m
X = reshape(X, [m*n 1])
```

Здесь матрицы F_n и F_m равны F_n и F_m соответсвенно, а массив T_n_m соответствует матрице вращений $T_{mn}^{(n)}$ и вычисляется следующим образом:

```
omega_mn = exp(2i * pi / (m*n))
i  = [0 : m−1]
j  = [0 : n−1]
T_n_m = omega_mn .^ (− j' * i)
```

5°. Рассмотрим общий случай. Пусть $N = n_1 n_2 \ldots n_s$. Тогда ([2])

$$F_N = \left(\prod_{\nu=1}^{s} \Big(I_{\Delta_\nu} \otimes F_{n_\nu} \otimes I_{N_\nu}\Big) \Big(I_{\Delta_\nu} \otimes T_{N_{\nu-1}}^{(N_\nu)}\Big) \right) R_N^T\,,$$

где

$$R_N = \prod_{\nu=1}^{s-1} \Big(I_{\Delta_\nu} \otimes L_{N_{\nu-1}}^{(N_\nu)}\Big)\,.$$

Пусть массив F содержит матрицы ДПФ F_{n_ν}. Он вычисляется так:

```
for nu = 1 : s
        kl  = [0 : n(nu)−1]
        omega = exp(2i * pi / n(nu))
        F{nu} = omega .^ (− kl' * kl)
end
```

Массив T содержит матрицы вращений $T_{N_{\nu-1}}^{(N_\nu)}$. Он вычисляется так:

```
omega = exp(2i * pi / Nn)
for nu = 1 : s
    i = [0 : n(nu)−1]
    j = [0 : N(nu)−1]
    T{nu} = omega .^ (− j' * i)
    omega = exp(2i * pi / N(nu))
end
```

Теперь вычислим массив индексов r, равный $R_N \cdot \begin{bmatrix} 1 & 2 & \dots & N \end{bmatrix}^T$:

```
r = [1 : Nn]
for nu = s−1 : −1 : 1
    r = reshape(r, [N(nu) n(nu) Delta(nu)])
    r = permute(r, [2 1 3])
end
r = reshape(r, [Nn 1])
```

Справедлива формула ([2])

$$F_N = \left(\prod_{\nu=1}^{s} \left[\left(F_{n_\nu} \otimes I_{N/n_\nu} \right) \left(T_{N_{\nu-1}}^{(N_\nu)} \otimes I_{\Delta_\nu} \right) L_N^{(n_\nu)} \right] \right) R_N^T .$$

Эта формула соответствует векторной форме вычислений. Запишем программу вычисления вектора $X = F_N\, x$:

```
X(r) = x
for nu = s : −1 : 1
    X = reshape(X, [n(nu) Nn/n(nu)])
    X = X .'
    X = X .* TI{nu}
    X = X * F{nu}
end
X = reshape(X, [Nn 1])
```

Здесь массив TI соответствует матрице $T_{N_{\nu-1}}^{(N_\nu)} \otimes I_{\Delta_\nu}$ и вычисляется следующим образом:

```
for nu = 1 : s
    TI{nu} = reshape(T{nu}, [1 N(nu)*n(nu)])
    TI{nu} = repmat(TI{nu}, [Delta(nu) 1])
    TI{nu} = reshape(TI{nu}, [Nn/n(nu) n(nu)])
end
```

Для примера рассмотрим альтернативный вариант факторизации матрицы ДПФ без предварительной перестановки R_N^T [2]:

$$F_N = \prod_{\nu=1}^{s} \Big(F_{n_\nu} \otimes I_{N/n_\nu}\Big)\Big(T_{N_{\nu-1}}^{(N_\nu)} \otimes I_{\Delta_\nu}\Big)\Big(L_{N_{\nu-1}}^{(n_\nu)} \otimes I_{\Delta_\nu}\Big).$$

Запишем программу вычисления вектора $X = F_N\, x$ по этой формуле:

```
X = x
for nu = s : −1 : 1
    X = reshape(X, [Delta(nu) n(nu) N(nu)])
    X = permute(X, [1 3 2])
    X = reshape(X, [N/n(nu) n(nu)]);
    X = X .∗ TI{nu}
    X = X ∗ F{nu}
end
X = reshape(X, [Nn 1])
```

6°. Рассмотрим случай, когда $N = n^s$. Справедливо разложение ([2])

$$F_{n^s} = \left(\prod_{\nu=1}^{s} \Big[L_N^{(n)}\big(I_{N/n} \otimes F_n\big)G_N^{(\nu)}\Big]\right) \mathrm{Rev}_{n^s}\,, \tag{6}$$

где

$$G_N^{(\nu)}[i + j\,n, i + j\,n] = \omega_{N_{\nu-1}}^{i\lfloor j/\Delta_\nu\rfloor}\,, \qquad i \in 0 : n-1\,, \quad j \in 0 : n^{s-1}-1\,.$$

Здесь $\Delta_\nu = n^{\nu-1}$, $N_\nu = N/n^\nu$ и $\mathrm{Rev}_{n^s} = R_N^T = R_N$.

Пусть переменная n равна n, переменная Nn равна N и переменная N_n равна N/n. Пусть массивы Delta и N содержат числа Δ_ν и N_ν соответственно.

```
Nn = n ^ s
N_n = N / n
Delta = n .^ [0 : s]
N = Delta(s : −1 : 1)
```

Введём массив F_n, равный F_n:

```
kl = [0 : n−1]
omega = exp(2i ∗ pi / n)
F_n = omega .^ (− kl' ∗ kl)
```

Отметим тот факт, что в формуле (6) под знаком произведения от ν зависит только матрица $G_N^{(\nu)}$. Это позволяет реализовать универсальную программу вычисления вектора $X = L_N^{(n)}\big(I_{N/n} \otimes F_n\big)\, x$ для любого s.

Программа вычисления массива G записывается так:

```
i = [0 : n−1]
j = [0 : N_n − 1]
omega = exp(2i * pi / Nn)
for nu = 1 : s
    k = floor(j / Delta(nu))
    G{nu} = omega .^ (− i' * k)
    omega = exp(2i * pi / N(nu))
end
```

Запишем программу вычисления вектора $X = F_{n^s}\, x$, где матрица F_{n^s} определяется формулой (6) :

```
X = x(r)
for nu = s : −1 : 1
    X = reshape(X, [n N_n])
    X = X .* G{nu}
    X = F_n * X
    X = X .'
end
X = reshape(X, [Nn 1])
```

Для случая $N = n^s$ индексы r соответствуют классической перестановке reverse [2]. В этом случае их можно вычислить проще, воспользовавшись функцией digitrevorder из Signal Processing Toolbox:

```
r = [1 : Nn]
r = digitrevorder(r, n)
```

7°. В [3] с современных позиций рассматривается вопрос о факторизации матрицы Фурье, порядок которой равен произведению попарно взаимно простых чисел. Для этого случая вычисления можно произвести более эффективным образом. Используется обозначение

$$\langle k\rangle_n = k - \lfloor k/n\rfloor\, n\,, \qquad k \in \mathbb{Z}\,.$$

Пусть m, n — взаимно простые натуральные числа. Введём две квадратные матрицы порядка mn:

$$Q_{mn}^{(n)}[k, k_1\, n + k_2] = \begin{cases} 1\,, & \text{если } k_1 = \langle k\rangle_m \text{ и } k_2 = \langle k\rangle_n, \\ 0 & \text{в остальных случаях;} \end{cases}$$

$$P_{mn}^{(n)}[l, l_1\, n + l_2] = \begin{cases} 1\,, & \text{если } l = \langle l_1\, n + l_2\, m\rangle_{mn}, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Здесь $k, l \in 0 : mn-1$, $k_1, l_1 \in 0 : m-1$, $k_2, l_2 \in 0 : n-1$. Матрица $Q_{mn}^{(n)}$ заполняется по строкам, а матрица $P_{mn}^{(n)}$ — по столбцам.

Умножение матрицы $Q_{mn}^{(n)}$ на вектор $x \in \mathbb{C}^{mn}$ порождает вектор $X = Q_{mn}^{(n)}\, x$ с компонентами

$$\begin{gathered} X(k) = x(k_1\, n + k_2)\,, \\ k \in 0 : mn-1\,, \quad k_1 = \langle k \rangle_m\,, \quad k_2 = \langle k \rangle_n\,. \end{gathered} \tag{7}$$

Программа, соответствующая формуле (7), выглядит следующим образом:

```
k = [0 : mn−1]
k1 = mod(k, m)
k2 = mod(k, n)
q = k1 * n + k2 + 1
X = x(q)
```

Умножение матрицы $P_{mn}^{(n)}$ на вектор $x \in \mathbb{C}^{mn}$ порождает вектор $X = P_{mn}^{(n)}\, x$ с компонентами

$$\begin{gathered} X(l) = x(l_1\, n + l_2)\,, \\ l_1 \in 0 : m-1\,, \quad l_2 \in 0 : n-1\,, \quad l = \langle l_1\, n + l_2\, m \rangle_{mn}\,. \end{gathered} \tag{8}$$

Программа, соответствующая формуле (8), выглядит следующим образом:

```
l1 = [0 : m−1]
l1 = repmat(l1, n, 1)
l1 = reshape(l1, [1 m*n])
l2 = [0 : n−1]
l2 = repmat(l2, 1, m)
l = mod(l1 * n + l2 * m, m*n)
p = l + 1
X(p) = x
```

Справедлива формула ([3])

$$F_{mn} = Q_{mn}^{(n)} \big(F_m \otimes F_n\big) \big(P_{mn}^{(n)}\big)^T\,.$$

Программа вычисления вектора $X = F_{mn}\, x$ записывается так:

```
X(p) = x
X = reshape(x, [m n])
X = X .'
X = F_n * X * F_m
X = X(q)
X = reshape(X, [m*n 1])
```

где индексы q и p вычисляются по программам, приведённым выше.

Пусть N равно произведению попарно взаимно простых чисел $n_1, n_2, \ldots$ $\ldots, n_s$, отличных от единицы. Тогда ([4])

$$F_N = Q_N \bigl(F_{n_1} \otimes F_{n_2} \otimes \cdots \otimes F_{n_s}\bigr) P_N^T\,,$$

где

$$Q_N = \prod_{\nu=1}^{s-1} \Bigl(I_{\Delta_\nu} \otimes Q_{N_{\nu-1}}^{(N_\nu)}\Bigr)\,, \qquad P_N = \prod_{\nu=1}^{s-1} \Bigl(I_{\Delta_\nu} \otimes P_{N_{\nu-1}}^{(N_\nu)}\Bigr)\,.$$

Программы вычисления вектора $X = (F_{n_1} \otimes F_{n_2} \otimes \cdots \otimes F_{n_s})\, x$ записываются аналогичным образом с учётом формул (3)–(5). Для примера, запишем программу вычисления вектора $X = F_N\, x$, воспользовавшись формулой (3):

```
X(p) = x
for nu = s : −1 : 1
    X = reshape(X, [n(nu) Nn/n(nu)])
    X = X .'
    X = X ∗ F{nu}
end
X = X(q)
X = reshape(X, [Nn 1])
```

Здесь массивы индексов q и p соответсвуют перестановкам, определяемые матрицами Q и P. Матрица Q называется матрицей *китайских перестановок.* Программа вычисления индексов q записывается следующим образом:

```
q = [1 : Nn]
for nu = s−1 : −1 : 1
    q = reshape(q, [N(nu)∗n(nu) Delta(nu)])
    k = [0 : N(nu)∗n(nu)−1]
    k1 = mod(k, n(nu))
    k2 = mod(k, N(nu))
    q = q(k1 ∗ N(nu) + k2 + 1, :)
end
q = reshape(q, [Nn 1])
```

Матрица P называется матрицей *руританских перестановок.* Программа вычисления индексов p записывается так:

```
p = [1 : Nn]
for nu = s−1 : −1 : 1
    p = reshape(p, [N(nu)∗n(nu) Delta(nu)])
    l1 = [0 : n(nu)−1]
    l1 = repmat(l1, N(nu), 1)
```

```
        l1 = reshape(l1, [1 N(nu)*n(nu)])
        l2 = [0 : N(nu)−1]
        l2 = repmat(l2, 1, n(nu))
        l = mod(l1 * N(nu) + l2 * n(nu), N(nu)*n(nu))
        p(l+1, :) = p
    end
    p = reshape(p, [Nn 1])
```

8°. В заключение заметим, что эффективность вычислений вектора $X = (F_{n_\nu} \otimes I_{N/n_\nu})\,x$ в приведённых программах можно повысить, если воспользоваться алгоритмами БПФ малого порядка [4]. Для примера приведём подпрограммы БПФ для $n_\nu = 3$ и $n_\nu = 4$:

```
function X = fft_3(x)
                                T2 = x(2, :) + x(3, :)      T3 = x(2, :) − x(3, :)
T2 = T2 * −1.5                  X(1, :) = x(1, :) + T2
T3 = T3 * −1i * sqrt(3)/2       T2 = X(1, :) + T2
                                X(2, :) = T2 + T3           X(3, :) = T2 − T3

function X = fft_4(x)
                                T1 = x(1, :) + x(3, :)      T3 = x(1, :) − x(3, :)
                                T2 = x(2, :) + x(4, :)      T4 = x(2, :) − x(4, :)
T4 = T4 * −1i                   X(1, :) = T1 + T2           X(3, :) = T1 − T2
                                X(2, :) = T3 + T4           X(4, :) = T3 − T4
```

ЛИТЕРАТУРА

1. Малозёмов В. Н., Просеков О. В. *Перестановки и кронекерово произведение матриц* // Семинар «DHA & CAGD». Избранные доклады. 31 марта 2004 г. (`http://dha.spb.ru/reps04.shtml#0331`).

2. Малозёмов В. Н., Просеков О. В. *Факторизация Кули-Тьюки матрицы Фурье* // Семинар «DHA & CAGD». Избранные доклады. 7 апреля 2004 г. (`http://dha.spb.ru/reps04.shtml#0407`).

3. Малозёмов В. Н., Просеков О. В. *Факторизация Гуда матрицы Фурье* // Семинар «DHA & CAGD». Избранные доклады. 5 мая 2004 г. (`http://dha.spb.ru/reps04.shtml#0505`).

4. Малозёмов В. Н., Просеков О. В. *О быстром преобразовании Фурье малых порядков* // Вестник СПбГУ. Сер. 1. 2003. Вып. 1 (№ 1). С. 36–45.