Ткаченко Д.С.

МИФИ, каф. высшей математики

И Л Л Ю С Т Р А Ц И О Н Н Ы Й М А Т Е Р И А Л

1. Применение ряда Фурье для аппроксимации изображений

Ряды, в частности, ряды Фурье, можно использовать для аппроксимации изображений. Пусть, например, дано изображение с маленьким разрешением:

Предположим, мы хотим его увеличить в 2 раза, но так, чтобы гладкие линии остались гладкими, а не ступенчатыми, как при простом удвоении размеров (см. последний рисунок).

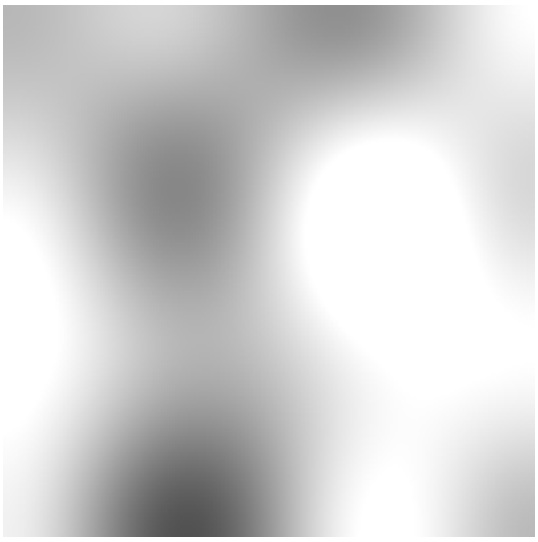
Превратим этот рисунок в сеточную функцию. Каждая точка рисунка имеет яркость, задаваемую в формате .bmp целочисленным значением от 0 до 255. Назовем полученную функцию $f(x,\, y)$ и воспользуемся формулами частичных сумм двойного ряда Фурье по косинусам:

$$S_{mn}(x,y) = \sum_{j=-m}^{j=m} \sum_{k=-n}^{k=n} a_{jk}\ \cos\frac{\pi j x}{l_1}\ \cos\frac{\pi k y}{l_2}, \tag{1.1}$$

где l_1- ширина рисунка, а l_2- его высота (в нашем примере $l_1 = l_2 = 100$ пикселей), а коэффициенты a_{jk} находятся по формулам:

$$a_{jk} = \frac{1}{l_1 l_2} \int\limits_0^{l_1} \int\limits_0^{l_2} f(x,\, y)\ \cos\frac{\pi j x}{l_1}\ \cos\frac{\pi k y}{l_2}\ dydx. \tag{1.2}$$

Приведем найденные частичные суммы ряда для различного числа слагаемых:

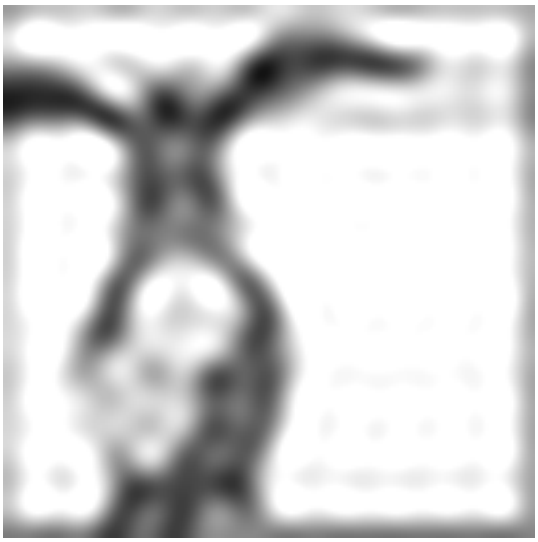


$m = n = 3$

$m = n = 7$

© Д.С. Ткаченко

$m = n = 10$



$m = n = 20$

$m = n = 30$

$m = n = 40$

$m = n = 50$

$m = n = 60$

© Д.С. Ткаченко

$m = n = 70$

$m = n = 80$

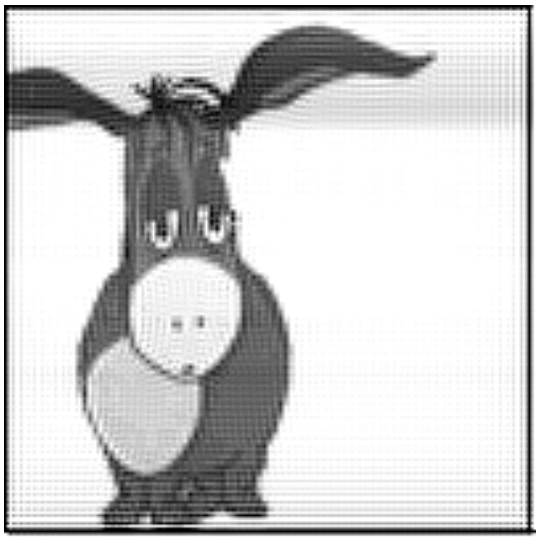
$m = n = 90$

$m = n = 100$

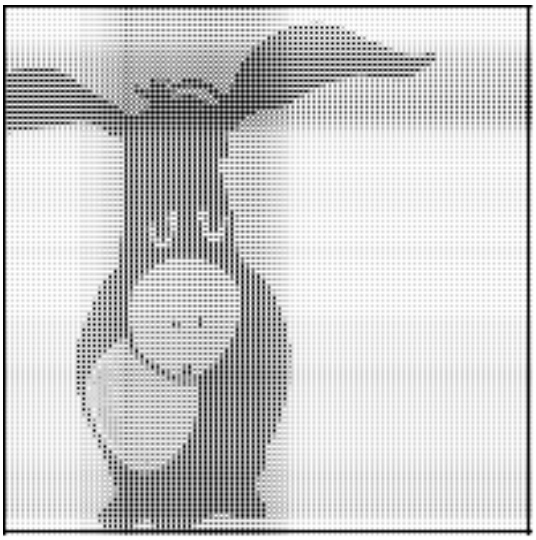
$m = n = 110$

$m = n = 120$

© Д.С. Ткаченко



$m = n = 150$



$m = n = 200$

Легко заметить, что качество изображения после определенного момента начинает портиться, и картинка, полученная при $m = n = 200$ (и даже при $m = n = 150$) уже существенно хуже той, которую мы получили при значениях $m = n = 100$. Это происходит по причине накопления погрешностей округления в процессе счета, которые с какого-то момента начинают превосходить величины слагаемых ряда. Ведь общий член ряда Фурье стремится к нулю, в то время как погрешности округления, как раз напротив, с каждым шагом накапливаются.

Человеческий глаз легко определит, что из всех полученных при помощи ряда Фурье картинок наилучшими по качеству являются те, у которых параметры m и n лежат в пределах от 90 до 110. Какой смысл несут в себе эти параметры? В соответствии с формулой (1.1), числа m и n означают пределы суммирования. Легко посчитать, что количество слагаемых в сумме, стоящих в правой части (1.1), равно $(2m+1)(2n+1)$. В частности, при $m = n = 10$, число слагаемых в сумме равно 441, при $m = n = 100$, оно равно 40401, а при $m = n = 200$, это число становится уже 160801.

1.1. Эффект Гиббса

Любопытно сравнить лучший просчитанный рисунок с исходным, просто увеличенным вдвое:

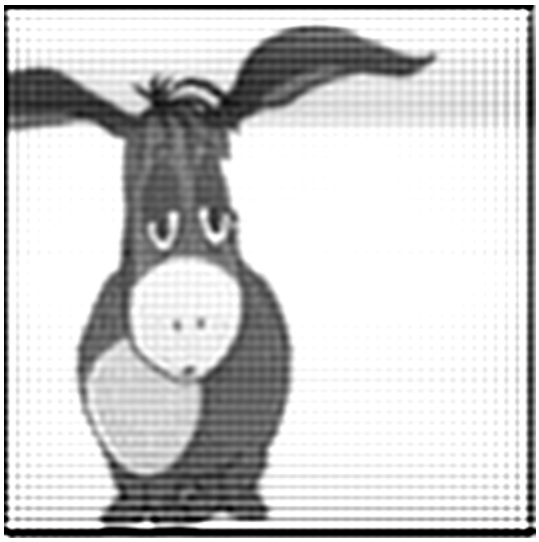
$m = n = 100$

исходное изображение $\times 2$

Легко заметить, насколько более четкой стала, например, форма глаз или «прическа» после применения ряда Фурье. Добавился, правда, периодический «шум», вызванный тем, что мы, вычисляя частичную, а не полную сумму ряда Фурье, пренебрегаем слагаемыми с более высокой частотой, чем последнее слагаемое.

Это так называемый **эффект Гиббса**: в окрестности точек разрыва ряд Фурье сходится неравномерно, и это проявляется в том, что у суммы конечного числа членов ряда Фурье есть характерные всплески в окрестности точки разрыва исходной функции, частота которых увеличивается с увеличением числа слагаемых конечной суммы ряда.[1]

И ещё следует обратить внимание, что значительная доля шума, появившегося в полученных через ряд Фурье картинках, порождена чёрной рамкой у исходного рисунка. Если её сделать белой, то разрыв, а следовательно и эффект Гиббса будут меньше. Вот изображения, полученные как частичная сумма ряда Фурье при $m = n = 100$ при 4х-кратном увеличении, первый – из картинки с чёрной рамкой, второй – с белой рамкой.

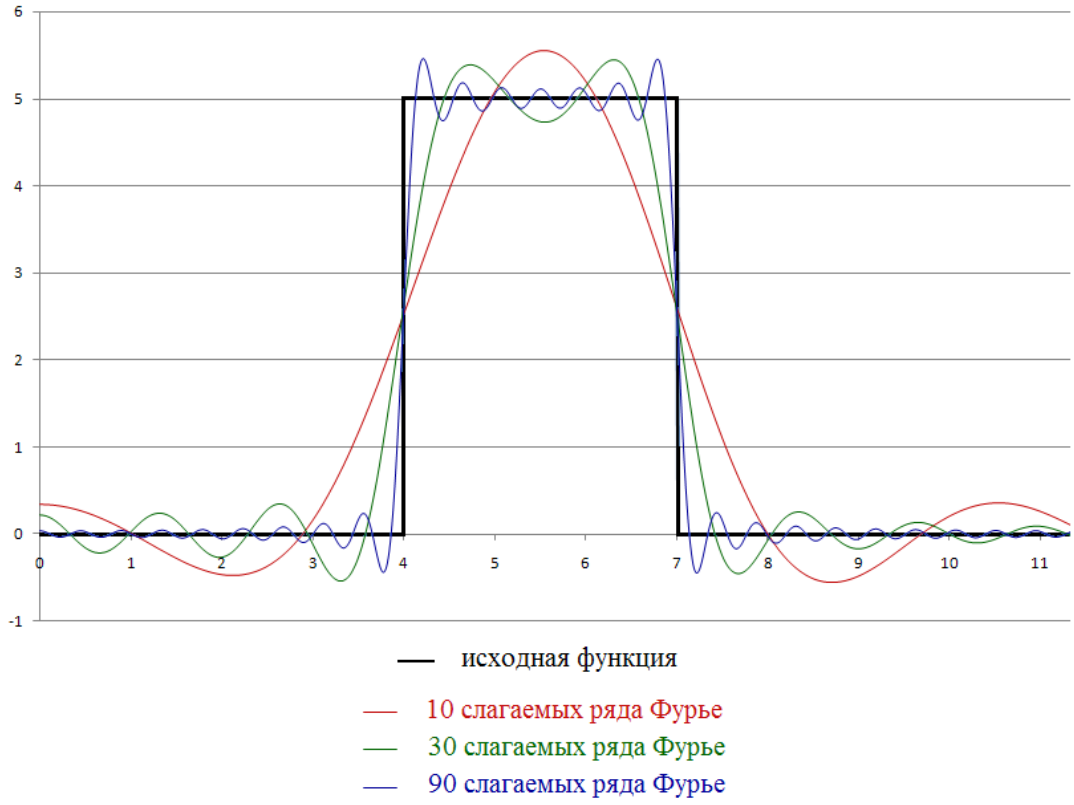


Обратим внимание, что вдали от резких цветовых переходов этот эффект минимален, в то же время он накапливается в областях, где у исходного изображения был максимальный контраст (на уровней ушей и ног Иа) и в окрестности границ картинки.

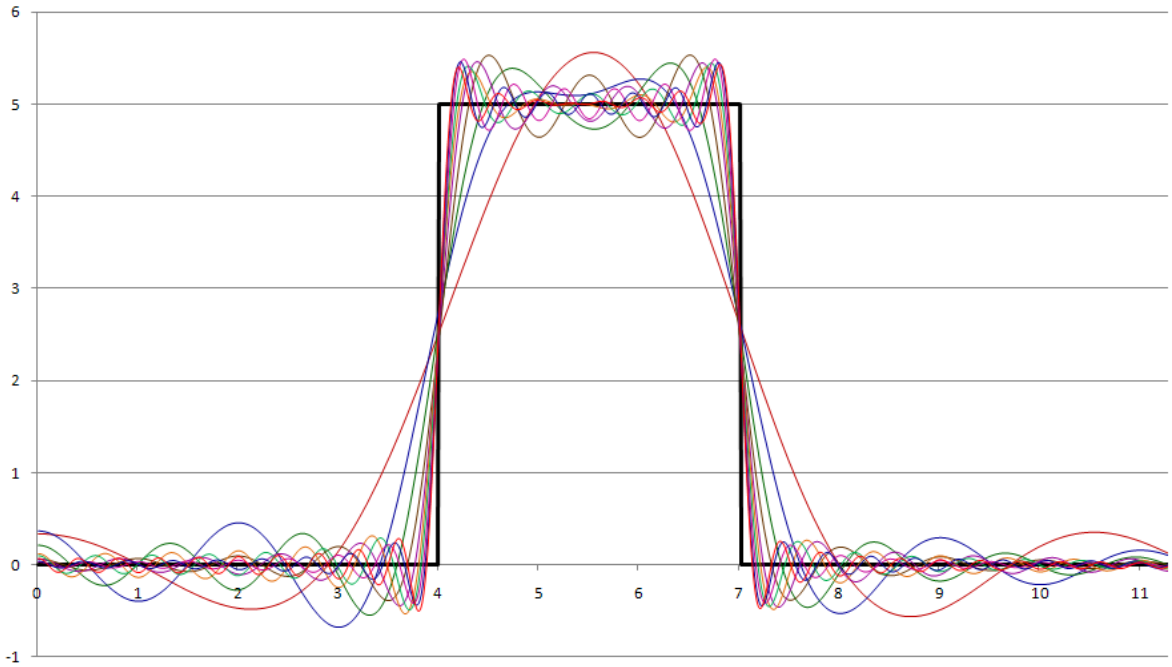
Проиллюстрируем эффект Гиббса, построив частичные суммы ряда Фурье по косинусам для функции с двумя ступеньками:

$$f(x) = \begin{cases} 0, & x \in (0,\ 4); \\ 5, & x \in (4,\ 7); \\ 0, & x \in (7,\ 20). \end{cases} \tag{1.3}$$

[1] Масса иллюстраций этого эффекта, а также свойств преобразования Фурье и интеграла Фурье содержится в электронных пособиях
В.П. Дьяконов *«Имитация, практический спектральный анализ и синтез сигналов»*
http://www.library.vstu.edu.ru/ellib/Exponenta_Ru/soft/Mathemat/dyakonov/nb5/nb5.asp.htm,
Е. Коплович, Д. Коплович, С. Умняшкин *«Теоретические основы цифровой обработки и представления сигналов: практикум»*, адрес http://www.swamp.ru/dsp/links/sem-lab.pdf,
Один из приёмов борьбы с эффектом Гиббса изложен в статье
С. Хонина, В. Баранов, В. Котляр *«Спектральный метод увеличения фрагментов цифровых изображений», Компьютерная оптика. Вып.19. С. 165-173 (1999)*, с которой можно ознакомиться по адресу
http://www.smr.ru/IPSI/research/publication/KO/PDF/KO19/ko19332.pdf.

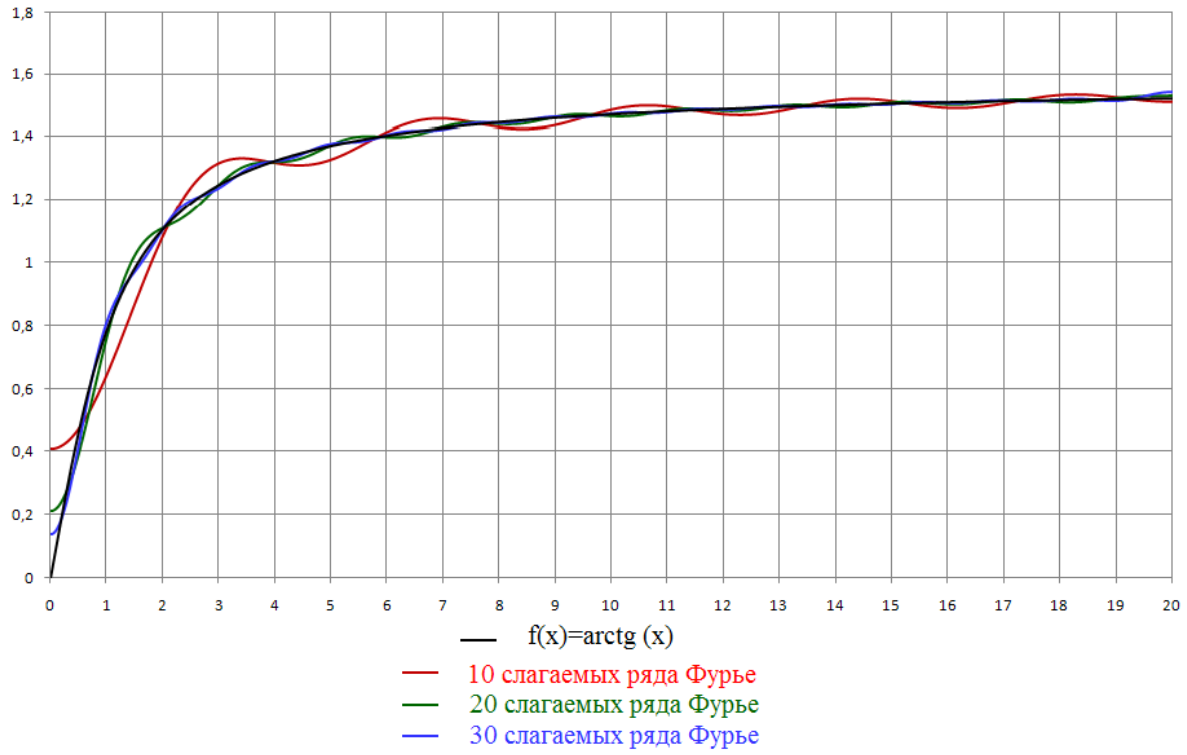


Заметим, что амплитуда всплесков в окрестности точки разрыва слабо меняется от количества слагаемых ряда: на рисунке ниже приведены графики частичных сумм ряда Фурье той же функции с количеством слагаемых 10, 20, 30, 40, 50, 60, 70, 80, 90 и 100.



При этом на отрезках, где нет точек разрыва, сходимость ряда Фурье равномерна, и на графике это отражается в гораздо меньшем отклонении частичной суммы даже с небольшим

числом слагаемых от исходной функции:



1.2. Формат *jpeg* и эффект Гиббса

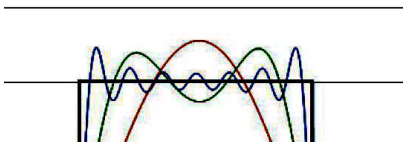
Методы, родственные разложению в ряд Фурье, используются при *jpeg*-сжатии изображений и *mpeg*-сжатии видео. Поэтому всем хорошо знакомы проявления эффекта Гиббса, – волнообразные искажения, появляющиеся в окрестности резких цветовых переходов.

Для *jpeg*-сжатия изображений, например, используется дискретное косинус-преобразование Фурье для фрагментов 8×8 пикселей, которое практически совпадает (см. формулы (1.4) – (1.7)) с частичной суммой ряда Фурье по косинусам, использованной нами, и может быть посчитано по численному алгоритму под названием «быстрое преобразование Фурье»[2]. Чем выше степень компрессии, тем заметнее искажения. Проиллюстрируем это на примере одного и того же фрагмента фотографии, погвергнутого разной степени *jpeg*-сжатия:

[2]Подробное описание алгоритма jpeg-сжатия, а также быстрого преобразования Фурье можно найти, например, в пособии **Е. Коплович, Д. Коплович, С. Умняшкин** *«Теоретические основы цифровой обработки и представления сигналов: практикум»*, адрес http://www.swamp.ru/dsp/links/sem-lab.pdf.

А вот фрагмент текста со страницы 5, с которым были проделаны те же дейтсвия:

$$f(x) = \begin{cases} 0, & x \in (0,\ 4); \\ 5, & x \in (4,\ 7); \\ 0, & x \in (7,\ 20). \end{cases}$$



$$f(x) = \begin{cases} 0, & x \in (0,\ 4); \\ 5, & x \in (4,\ 7); \\ 0, & x \in (7,\ 20). \end{cases}$$

$$f(x) = \begin{cases} 0, & x \in (0,\ 4); \\ 5, & x \in (4,\ 7); \\ 0, & x \in (7,\ 20). \end{cases}$$

Очень хорошо видно, что «грязь» – волнообразные искажения из-за эффекта Гиббса – располагается квадратами, а в квадратах, где на исходной картинке не было цветовых переходов, всё идеально «чисто».

Это показывает, что кодирование фрагментов 8×8 позволяет избежать распространения искажений за пределы каждого из маленьких квадратов. И в отличие от восстановленной нами картинки с Иа, где каждая волна повторялась от края до края, метод *jpeg*-компрессии оставляет волны лишь в непосредственной окрестности того цветового перехода, который их породил. В то же время, алгоритм *jpeg*-сжатия не предполагает увеличения разрешения, и сохраняет картинку в виде специальным образом обработанного (с потерей информации) образа Фурье с абсолютно тем же количеством точек, которое было у исходной картинки. Обратный алгоритм распаковки *jpeg*-файла проводит обратное дискретное косинус-преобразование, также не меняя разрешения. Эти алгоритмы действуют по формулам:

$$F(k,\ l) = \frac{2}{\sqrt{MN}}\, c(k)\, c(l) \sum_{j=0}^{N-1} \cos\left(\frac{\pi k}{N}\left(j + \frac{1}{2}\right)\right) \sum_{m=0}^{M-1} f(j,\ m) \cos\left(\frac{\pi l}{M}\left(m + \frac{1}{2}\right)\right), \tag{1.4}$$

$$k,\ j \in \overline{0,\ N-1}, \quad l,\ m \in \overline{0,\ M-1}, \tag{1.5}$$

$$\check{f}(j,\ m) = \frac{2}{\sqrt{MN}} \sum_{k=0}^{N-1} c(k)\, \cos\left(\frac{\pi k}{N}\left(j + \frac{1}{2}\right)\right) \sum_{l=0}^{M-1} c(l)\, F(k,\ l) \cos\left(\frac{\pi l}{M}\left(m + \frac{1}{2}\right)\right), \tag{1.6}$$

$$c(n) = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0; \\ 1, & n > 0. \end{cases} \tag{1.7}$$

Формулы написаны в общем виде, хотя при *jpeg*-компрессии рассматриваются квадраты 8×8, поэтому реально в (1.4) – (1.6) параметры $M = N = 8$.

Заинтересованному читателю предоставляется самостоятельно выяснить, в чём совпадают и в чём различаются формулы (1.4) – (1.7) и численная реализация формул (1.1) – (1.2).

Отметим, однако, что знание функции $F(k,\ l)$ – численного образа Фурье – позволяет восстанавливать исходную функцию f в любом количестве точек: достаточно только в формуле (1.6) заменить N на N_1, а M – на M_1, и разрешить индексам j и m пробегать значения в пределах

$$j \in \overline{0,\ N_1 - 1}, \quad m \in \overline{0,\ M_1 - 1},$$

то полученная по (1.6) функция $\check{f}(j,\ m)$ будет задавать картинку с разрешением $M_1 \times N_1$ вместо $M \times N$.

1.3. Быстрое преобразование Фурье

Приведём для полноты картины краткое описание этого алгоритма.

Быстрое преобразование Фурье – численный метод, позволяющий сократить число операций при вычислениии преобразования Фурье или ряда Фурье с величины порядка N^2,

где N – количество сеточных значений функции $f(k)$ до величины порядка $N\log_2 N$, что при объёмных вычислениях даёт значительную экономию. В оригинальном виде, как его создали его авторы – J.W. Cooley и J.W. Tukey в 1967 году, этот метод рассчитан на случай $N = 2^n$ и состоит в том, что сумма

$$F(k) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} f(j) e^{-\frac{2\pi i}{2^n} kj}, \tag{1.8}$$

может быть разбита на половины, вычисляемые идентичным образом, а так как половина от 2^n – тоже степень двойки, то и каждая из половин суммы (1.8) может быть разбита пополам и так далее до 2^0, когда сумма состоит уже из одного единственного слагаемого.

Разберём метод Cooley–Tukey для случая **прореживания по времени**[3]. В этом случае сумма (1.8) разбивается на суммы отдельно слагаемых с **чётным** и **нечётным** j.

У каждого шага с номером $k \in \overline{1,\ n}$ будет
на входе: 2^{n-k+1} векторов $\vec{F}_{k-1,l}$, $\quad l \in \overline{0,\ 2^{n-k+1}-1}$, в каждом векторе по 2^{k-1} элементов;
на выходе: 2^{n-k} векторов $\vec{F}_{k,m}$, $\quad m \in \overline{0,\ 2^{n-k}-1}$, в каждом векторе по 2^k элементов.

Алгоритм

<u>Шаг 0.</u> Для каждого l от 0 до 2^n-1 определим «вектор» $\vec{F}_{0,l}$ с одним элементом по закону:

$$\vec{F}_{0,l} = f(l), \qquad l \in \overline{0,\ 2^n-1}. \tag{1.9}$$

<u>Шаг k.</u>

1) Определим вектор

$$\vec{W} = \left\{1,\ e^{-\frac{2\pi i}{2^k}\cdot 1},\ e^{-\frac{2\pi i}{2^k}\cdot 2},\ e^{-\frac{2\pi i}{2^k}\cdot 3},\ \ldots,\ e^{-\frac{2\pi i}{2^k}\cdot\left(2^{k-1}-1\right)}\right\}. \tag{1.10}$$

2) Для каждого m от 0 до $2^{n-k}-1$ умножим вектор $\vec{F}_{k-1,m+2^{n-k}}$ скалярно на вектор $\vec{W}$:

$$\vec{G}_{k-1,m} = \left(\vec{F}_{k-1,m+2^{n-k}},\ \vec{W}\right) \qquad m \in \overline{0,\ 2^{n-k}-1}. \tag{1.11}$$

3) Для каждого m от 0 до $2^{n-k}-1$ определим вектор $\vec{F}_{k,m}$ по закону:

$$\vec{F}_{k,m} = \left\{ \begin{array}{c} \vec{F}_{k-1,m} + \vec{G}_{k-1,m} \\ \\ \vec{F}_{k-1,m} - \vec{G}_{k-1,m} \end{array} \right\}, \qquad m \in \overline{0,\ 2^{n-k}-1}. \tag{1.12}$$

Эту запись следует понимать так, что первая половина элементов нового вектора состоит из элементов $\vec{F}_{k-1,m} + \vec{G}_{k-1,m}$, а вторая половина – из элементов $\vec{F}_{k-1,m} - \vec{G}_{k-1,m}$.

<u>Шаг $n+1$.</u> Функцию $F(j)$, $\ j \in \overline{0,\ 2^n-1}$, из равенства (1.8) получим из равенства

$$F(j) = \frac{1}{\sqrt{2^n}}\, F_{n,0,j}, \tag{1.13}$$

где $F_{n,0,j}$ – элемент с номером j вектора $\vec{F}_{n,0}$, полученного на шаге с номером n.

Аналогичные алогритмы применимы к синус-преобразованию, косинус преобразованию, а также к кратным преобразованиям Фурье, – в частности к двойному косинус-преобразованию, используемому при *jpeg*-сжатии изображений.

[3] У этого метода есть другая разновидность – **прореживание по частоте**. Кроме того, необходимо заметить, что существуют другие методы быстрого преобразования Фурье, о них можно узнать, например в книге **Блейхут** «Быстрые алгоритмы цифровой обработки сигналов».

1.4. Подавление эффекта Гиббса

Разработке методов подавления этого эффекта уделяется много внимания. Один из методов состоит в том, чтобы воспользоваться **«сглаживающим свойством уравнения теплопроводности»**[4] Функция, которую надо аппроксимировать (приблизить) рядом Фурье, ставится в качестве начального условия $\varphi(x)$ (в двумерном случае $\varphi(x,\ y)$) в задаче для уравнения теплопроводности. Эту задачу можно решить, например, методом разделения переменных (он также называется методом Фурье), или по формуле Пуассона и рассмотреть вместо исходной функции, равной $u\big|_{t=0}$, функцию u в момент времени $t_1 > 0$. Функция $u\big|_{t=t_1}$ будет уже бесконечно гладкой, поэтому эффект Гиббса при разложении её в ряд Фурье будет минимизирован. А если нам нужно, чтобы резкие цветовые переходы были действительно резкими, в их достаточно малой окрестности t_1 надо «устремить к нулю» – т.е. взять достаточно t_1 малым.

Проиллюстрируем использование сглаживающего свойства уравнения теплопроводности. Воспользумся решением задачи распространения тепла в квадратной мембране с теплоизолированными краями, без источников тепла и с начальным распределением температуры $\varphi(x,\ y)$:

$$\begin{cases} u_t - a^2(u_{xx} + u_{yy}) = 0, & x \in (0,\ p),\ \ y \in (0,\ s),\ \ t > 0; \\ u(x,\ y,\ 0) = \varphi(x,\ y), & x \in [0,\ p],\ \ y \in [0,\ s]; \\ u(0,\ y,\ t) = u(p,\ y,\ t) = 0, & y \in [0,\ s],\ \ t \geqslant 0; \\ u(x,\ 0,\ t) = u(x,\ s,\ t) = 0, & x \in [0,\ p],\ \ t \geqslant 0. \end{cases} \tag{1.14}$$

Решение этой задачи, полученное методом Фурье, имеет вид

$$u = \sum_{n,k=-\infty}^{+\infty} \varphi_{kn} \cos\left(\frac{\pi n x}{p}\right) \cos\left(\frac{\pi k y}{s}\right) e^{-a^2 \lambda_{kn} t}, \tag{1.15}$$

где

$$\lambda_{kn} = \left(\frac{\pi n}{p}\right)^2 + \left(\frac{\pi k}{s}\right)^2, \tag{1.16}$$

$$\varphi_{kn} = \frac{1}{ps} \int\limits_0^s \int\limits_0^p \varphi(x,\ y) \cos\left(\frac{\pi n x}{p}\right) \cos\left(\frac{\pi k y}{s}\right)\, dx\, dy. \tag{1.17}$$

Если теперь взять в качестве функции начального распределения температуры функцию яркости точки (с соответствующими координатами) картинки

в квадрате $x,\ y \in [0,\ 1]$, мы получим при разных $t > 0$ в разной степени «размазанную» картинку, соответствующую распределению температуры после того, как тепло растекалось по мембране.

[4] О нём можно прочитать, например в семинаре по уравнеиям математической физики http://tkachenko-mephi.narod.ru/pdfs/SemK16.rar.

Вот полученные при различных t результаты $\left(\text{величина } a^2 = \frac{1}{500000}\right)$:

$t = 5$

$t = 7$

$t = 30.$

Если полученные изображения обработать в простейшем графическом редакторе, увеличив яркость и контраст, мы получим

$t = 5$

$t = 7$

$t = 30.$

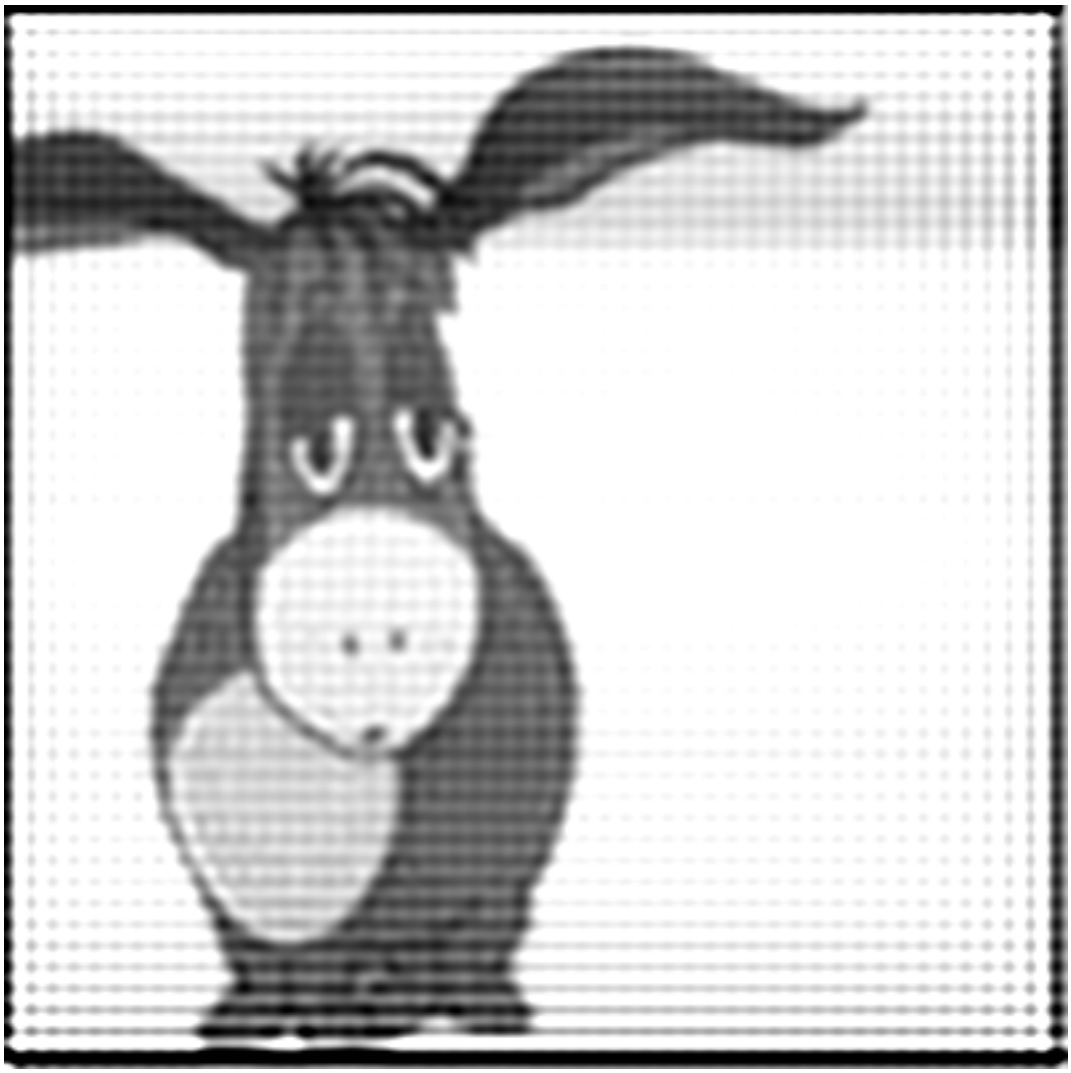
Осталось сравнить исходное изображение, увеличенное вдвое, с обработанным сглаженным изображением при $t = 7$:

исходное изображение $\times 2$

результат обработки.

При этом необходимо отметить, что, посчитав коэффициенты Фурье для $m = n = 100$, мы можем получить картинку сколь угодно высокого разрешения, и единственным фактором, снижающим ее качество, будет **только** этот самый периодический шум. Вот, например,

рисунок 400×400 пикселей, полученный из исходной картинки 100×100 частичной суммой ряда Фурье при $m = n = 100$.



$m = n = 100$

Приведем для сравнения часть исходного изображения (с чёрной рамкой), увеличенного в 4 раза, и такую же часть изображения, полученного через ряд Фурье, без сглаживания и обработки и часть изображения, полученного через ряд Фурье с использованием сглаживающего свойства уравнения теплопроводности, и увеличенными контрастом и яркостью:

исходное изображение $\times 4$ ряд Фурье $m = n = 100$ сглаженное и обработанное

Наконец, приведём целиком изображение, увеличенное в 4 раза с помощью ряда Фурье и с использованием сглаживающего свойства уравнения теплопроводности, после увеличения

контраста и яркости:

исходное изображение $\times 4$, $m = n = 100$
после сглаживания и обработки.