

Ссылка на документацию Intel:

[Developer Reference for Intel® oneAPI Math Kernel Library - C](#)

Интерполяция данных

Data Fitting Functions

Процедуры этого раздела библиотеки Intel MKL выполняют сплайн-интерполяцию данных с последующим вычислением значений функции, ее производных и интегралов от функции.

Приведенная ниже информация содержит описание подмножества процедур из пакета интерполяции библиотеки Intel MKL, информации о которых достаточно для выполнения лабораторной работы. Полный список процедур пакета с описанием можно найти в документации библиотеки Intel MKL.

Процедуры выполняют сплайн-интерполяцию для разных типов разбиения (partition) отрезка интерполяции $[a, b]$:

Неравномерное разбиение задается в виде массива узлов интерполяции (breakpoints) $\mathbf{x} = \{x_j\}$, $j = 1, 2, \dots, n$, которые удовлетворяют условию

$$a = x_1 < x_2 < \dots < x_n = b.$$

Число отрезков (ячеек) интерполяции (cells) равно $n - 1$:

$$\Delta_j = x_{j+1} - x_j, j = 1, 2, \dots, n - 1$$

В **равномерном** разбиении задаются концы отрезка интерполяции и число узлов. Концы отрезка передаются в процедуры как элементы массива с двумя элементами.

Типы сплайнов, реализованные в библиотеке MKL:

- левый кусочно-постоянный;
- правый кусочно-постоянный;
- линейный;
- классический (natural) параболический;
- параболический сплайн Субботина;
- классический (natural) кубический сплайн;
- кубический сплайн Эрмита;
- кубический сплайн Бесселя;
- кубический сплайн Акимы.

Работа пакета интерполяции библиотеки Intel MKL организована следующим образом:

- создается задача (task); при создании задачи передаются данные, для которых выполняется сплайн-интерполяция – разбиение отрезка интерполяции и значения функции в узлах интерполяции; возвращается дескриптор задачи (task descriptor);
- вызываются процедуры настройки параметров задачи, в частности, тип сплайна и граничные условия; эти процедуры называются процедурами конфигурации;
- вызываются процедуры, которые выполняют построение сплайна, интерполяцию, вычисление производных и интегралов; эти процедуры называются вычислительными процедурами;
- вызывается деструктор задачи, который освобождает ресурсы.

Пакет интерполяции содержит отдельные процедуры для интерполяции данных с одинарной (**float**) и двойной (**double**) точностью. Имена процедур, которые работают с разными типами данных, отличаются только одним символом (**s / d**). В документации Intel MKL при описании процедур, прототипы которых отличаются только типом элементов данных, для этого поля используется символ **'?'**.

Ниже перечислены основные процедуры, которые необходимо вызвать, чтобы построить сплайн и вычислить его значения.

df?NewTask1D	Создание задачи (task).
df?EditPPSpline1D	Настройка параметров задачи – выбор тип сплайна и граничных условий для сплайна.
df?Construct1D	Создание сплайна. Процедура вычисляет коэффициенты сплайна и размещает их в области памяти пользователя, адрес этой области передается через параметр процедуры конфигурации.
df?Interpolate1D	<p>Вычисление значений сплайна и производных. Значения сплайна и производных вычисляются для массива значений переменной, который передается как параметр процедуры. Через параметры передаются номера производных, которые необходимо вычислить. Вычисленные значения размещаются в области памяти пользователя.</p> <p>В том случае, когда значения переменной, для которых вычисляются значения интерполяционного сплайна, образуют равномерную сетку, можно передать только концы отрезка.</p>
df?Integrate1D	Вычисление интегралов на основе сплайнов. Вычисляются интегралы от интерполяционного сплайна на множестве отрезков. Концы отрезков передаются как элементы двух массивов – один для левых концов, другой для правых концов отрезков. Вычисленные значения размещаются в области памяти пользователя. Массивы с концами отрезков и массив результатов

не должны пересекаться в памяти.

dfDeleteTask Вызов деструктора задачи, который освобождает ресурсы.

Как входные параметры процедуры принимают адреса массивов с элементами типа **double** или **float** и именованные константы, с помощью которых задается тип сплайна, тип разбиения отрезка, формат размещения данных в памяти и т.д.

Эти константы определены в заголовочном файле `mkl_df_defines.h`

Процедуры создания задачи

При создании задачи передаются данные, для которых выполняется сплайн-интерполяция – разбиение отрезка интерполяции и значения в точках интерполяции. Если процесс создания задачи завершился успешно, присваивается значение дескриптору задачи (`task`), который затем передается как параметр другим процедурам пакета. В заголовочном файле `mkl_df_types.h` тип задачи **DFTaskPtr** определен как **void***.

В процедурах интерполяции для одной и той же сетки узлов интерполяции (точек разбиения отрезка) можно одновременно вычислить сплайны для нескольких наборов данных, которые в документации трактуются как элементы векторной функции.

Значения векторной функции передаются в процедуры как одномерный массив; по умолчанию считается, что используется формат Row-major, т.е. первые `n` элементов массива – это значения первого элемента векторной функции в узлах интерполяции, следующие `n` элементов массива – это значения второго элемента векторной функции в узлах интерполяции и т.д.

В пакете интерполяции реализовано несколько типов разбиения отрезка интерполяции, в том числе, неравномерное и равномерное. При неравномерном разбиении узлов в процедуры передаются координаты всех узлов интерполяции, при равномерном разбиении – только концы отрезка интерполяции. Тип разбиения выбирается при создании задачи интерполяции и передается в виде именованной константы.

Процедуры

```
int dfdNewTask1D (DFTaskPtr* task,  
                 const MKL_INT nx, const double* x, const MKL_INT xhint,  
                 const MKL_INT ny, const double* y, const MKL_INT yhint);  
int dfsNewTask1D (DFTaskPtr* task,  
                 const MKL_INT nx, const float* x, const MKL_INT xhint,  
                 const MKL_INT ny, const float* y, const MKL_INT yhint);
```

Входные параметры	Описание
-------------------	----------

<i>const MKL_INT nx</i>	Число узлов интерполяции
<i>const MKL_INT xhint</i>	Именованная константа, которая определяет тип разбиения отрезка интерполяции. Значения в таблице 1.
<i>const double*x</i> или <i>const float*x</i>	Массив узлов интерполяции, для неравномерного разбиения содержит <i>nx</i> элементов, для равномерного 2 элемента (концы отрезка). Массив должен быть отсортирован в порядке возрастания.
<i>const MKL_INT ny</i>	Размерность векторной функции (число наборов значений, для которых строятся сплайны).
<i>const MKL_INT yhint</i>	Именованная константа, которая определяет формат хранения значений векторной функции. Для формата Row-major значение DF_MATRIX_STORAGE_ROWS.
<i>const double*y</i> или <i>const float*y</i>	Одномерный массив значений векторной функции в узлах интерполяции. Число элементов <i>nx*ny</i> . Формат упаковки значений функций в одномерный массив определяется значением параметра <i>yhint</i> .

Таблица 1. Именованные константы, которые определяют тип разбиения отрезка

<i>DF_NON_UNIFORM_PARTITION</i>	Неравномерное разбиение
<i>DF_UNIFORM_PARTITION</i>	Равномерное разбиение
<i>DF_SORTED_DATA</i>	Неравномерное разбиение, координаты отсортированы в порядке возрастания
<i>DF_NO_HINT</i>	Умолчание – неравномерное разбиение.

Если создание задачи завершилось успешно, возвращаемое значение равно DF_STATUS_OK. В противном случае возвращается ненулевой код ошибки.

Значения всех кодов ошибки можно найти в документации Intel MKL в таблице “Status Codes in the Data Fitting Component”.

Массивы *x* (узлы интерполяции) и *y* (значения векторной функции в узлах интерполяции) необходимо сохранять в течение всего времени работы задачи интерполяции, так как они не копируются и используются в процессе вычислений.

Параметры *x* и *nx*, которые определяют разбиение отрезка интерполяции [*a*, *b*], являются обязательными. Число узлов интерполяции должно быть не меньше 2.

Если из-за некорректных значений входных параметров не удастся создать задачу, возвращаемое значение дескриптора задачи равно нулю.

Процедуры конфигурации задачи

В процедурах конфигурации задачи выбирается тип сплайна и граничные условия для сплайна.

Тип граничных условий для сплайна (в том случае, если они необходимы для выбранного типа сплайна) передаются как комбинация именованных констант из таблицы 2. Если граничные условия требуют задания некоторых значений, они также передаются через параметры одной из процедур конфигурации.

Для некоторых типов сплайнов необходимо задать значения во внутренних точках ячеек сплайна. Эти значения также передаются через параметры одной из процедур конфигурации.

Процедуры

```
int dfdEditPPSpline1D (DFTaskPtr task, const MKL_INT s_order, const MKL_INT s_type,
                      const MKL_INT bc_type, const double* bc,
                      const MKL_INT ic_type, const double* ic,
                      const double* scoeff, const MKL_INT scoeffhint)

int dfsEditPPSpline1D (DFTaskPtr task, const MKL_INT s_order, const MKL_INT s_type,
                      const MKL_INT bc_type, const float* bc,
                      const MKL_INT ic_type, const float* ic,
                      const float* scoeff, const MKL_INT scoeffhint)
```

Входные параметры	Описание
<i>DFTaskPtr task</i>	Дескриптор задачи
<i>const MKL_INT s_order</i>	Именованная константа, которая определяет порядок сплайна (степень многочлена плюс 1). Значение DF_PP_CUBIC для кубических сплайнов.
<i>const MKL_INT s_type</i>	Именованная константа, которая определяет тип сплайна. Значение DF_PP_NATURAL для классического кубического сплайна.
<i>const MKL_INT bc_type</i>	Именованная константа, которая определяет тип граничных условий. Значения из таблицы 2.
<i>const double* bc</i>	Массив значений для граничных условий. Длина массива зависит от значения параметра bc_type .
<i>const MKL_INT ic_type</i>	Именованная константа, которая определяет тип условий во внутренних точках. Для классического кубического сплайна DF_NO_IC
<i>const double* ic</i>	Массив значений для условий во внутренних точках; для классического кубического сплайна значение NULL
<i>const double* scoeff</i>	Массив для коэффициентов сплайна. Число элементов массива ny * s_order * (nx-1) . Формат хранения коэффициентов в массиве определяется значением параметра scoeffhint .
<i>const MKL_INT scoeffhint</i>	Именованная константа, которая определяет формат упаковки коэффициентов сплайна в одномерный массив; значение по умолчанию DF_NO_HINT – формат Row-major.

Если выполнение процедуры завершилось успешно, возвращаемое значение равно DF_STATUS_OK.

Таблица 2. Именованные константы, которые определяют граничные условия для кубических сплайнов

<i>Значение перечисления</i>	<i>Граничное условие</i>
DF_NO_BC	Граничные условия не заданы
DF_BC_NOT_A_KNOT	Условие “отсутствие узла”
DF_BC_FREE_END	Условие свободного конца (free-end) – вторая производная на концах отрезка равна 0
DF_BC_1ST_LEFT_DER	Первая производная на левом конце отрезка
DF_BC_1ST_RIGHT_DER	Первая производная на правом конце отрезка
DF_BC_2ND_LEFT_DER	Вторая производная на левом конце отрезка
DF_BC_2ND_RIGHT_DER	Вторая производная на правом конце отрезка
DF_BC_PERIODIC	Периодические граничные условия.

Значения перечисления, которые определяют тип граничных условий, можно комбинировать с помощью побитовой операции или. Чтобы передать первую производную на левом конце отрезка и вторую производную на правом конце, граничные условия надо задать так

DF_BC_1ST_LEFT_DER | DF_BC_2ND_RIGHT_DER.

Если для граничных условий на концах отрезка необходимо задать значения, они передаются как массив из двух элементов.

Если используется периодическое граничное условие, значения функции на концах отрезка интерполяции должны быть равны, в противном случае возвращается код ошибки.

Для значений DF_MATRIX_STORAGE_ROWS и DF_NO_HINT параметра *scoeffhint* коэффициенты сплайна хранятся в формате Row-major.

Сплайн степени k для первого элемента векторной функции:

$$P_1(x) = c_{1,0} + c_{1,1}(x - x_1) + \dots + c_{1,k}(x - x_1)^k, x \in (x_1, x_2)$$

...

$$P_j(x) = c_{j,0} + c_{j,1}(x - x_j) + \dots + c_{j,k}(x - x_j)^k, x \in (x_j, x_{j+1})$$

...

$$P_{n-1}(x) = c_{n-1,0} + c_{n-1,1}(x - x_{n-1}) + \dots + c_{n-1,k}(x - x_{n-1})^k, x \in (x_{n-1}, x_n)$$

Коэффициенты сплайна хранятся в массиве *scoeff* в следующем порядке

$$C_{1,0}, C_{1,1}, \dots, C_{1,k}, \dots, C_{j,0}, C_{j,1}, C_{j,k}, \dots, C_{n-1,0}, C_{n-1,1}, C_{n-1,k}$$

Затем коэффициенты сплайна для второго элемента векторной функции и т.д.

Вычислительные процедуры

Вычислительные процедуры сплайн-интерполяции выполняют построение сплайна, вычисление значений производных и интегралов, поиск индексов ячеек (cells), содержащих заданные точки.

После того как задача создана и необходимые параметры инициализированы, вычислительные процедуры можно вызывать многократно.

Построение интерполяционного сплайна

```
int dfsConstruct1D (DFTaskPtr task, const MKL_INT s_format, const MKL_INT method)
int dfdConstruct1D (DFTaskPtr task, const MKL_INT s_format, const MKL_INT method)
```

Входные параметры	Описание
<i>DFTaskPtr task</i>	Дескриптор задачи
<i>const MKL_INT s_format</i>	В настоящей версии поддерживается только одно значение <i>DF_PP_SPLINE</i>
<i>const MKL_INT method</i>	В настоящей версии поддерживается только одно значение <i>DF_METHOD_STD</i>

Вычисление значений интерполяционного сплайна и его производных

```
Int dfdInterpolate1D (DFTaskPtr task,
                     const MKL_INT type, const MKL_INT method,
                     const MKL_INT nsite, const double* site, const MKL_INT sitehint,
                     const MKL_INT ndorder, const MKL_INT* dorder,
                     const double* datahint,
                     double* r, const MKL_INT rhint, MKL_INT* cell)
Int dfsInterpolate1D (DFTaskPtr task,
                     const MKL_INT type, const MKL_INT method,
                     const MKL_INT nsite, const float* site, const MKL_INT sitehint,
                     const MKL_INT ndorder, const MKL_INT* dorder,
                     const float * datahint,
                     float* r, const MKL_INT rhint, MKL_INT* cell)
```

Входные параметры	Описание
<i>DFTaskPtr task</i>	Дескриптор задачи
<i>const MKL_INT type</i>	Именованная константа, которая определяет тип

	вычислений для сплайна. Значение для вычисления сплайна и его производных <i>DF_INTERP</i> .
<i>const MKL_INT method</i>	В настоящей версии поддерживается только одно значение <i>DF_METHOD_PP</i> .
<i>const MKL_INT nsite</i>	Число значений переменных, в которых вычисляются значения сплайна и его производных.
<i>const double* site</i>	При неравномерном разбиении массив координат точек длиной <i>nsite</i> , в которых вычисляются значения сплайна и его производных. При равномерном разбиении массив из двух элементов, который содержит концы отрезка.
<i>const MKL_INT sitehint</i>	Именованная константа, которая определяет тип разбиения для точек, в которых вычисляются значения сплайна и его производных. Значения из Таблицы 1.
<i>const MKL_INT ndorder</i>	Увеличенный на единицу максимальный порядок производной, для которых надо вычислить значения. Например, если вычисляются значения второй производной <i>ndorder</i> = 3.
<i>const MKL_INT* dorder</i>	Целочисленный массив размера <i>ndorder</i> , который содержит информацию о том, значения каких производных надо вычислить. Элементы массива должны быть равны 0 или 1. Для <i>dorder</i> с элементами {1, 1, 0, 1} будут вычислены значения сплайна и производные первого и третьего порядков. Значения сплайна считаются производной нулевого порядка.
<i>const double* datahint</i>	Массив, содержащий дополнительную информацию об узлах интерполяции и массиве точек, в которых вычисляются значения сплайна. Эти данные помогают ускорить вычисления. Если значение равно NULL, то при вычислениях используются значения по умолчанию.
<i>double* r</i>	Массив результатов. Число элементов массива равно <i>ny * nnder * nsites</i> , где <i>nnder</i> – число ненулевых элементов в массиве <i>dorder</i> . Массивы <i>r</i> и <i>site</i> не должны пересекаться в памяти.
<i>const MKL_INT rhint</i>	Именованная константа, которая определяет формат упаковки результатов в одномерный массив. Для значения <i>DF_MATRIX_STORAGE_ROWS</i> в одномерном массиве сначала размещается блок из <i>nnder * nsites</i> значений для первого элемента функции, затем блок из <i>nnder * nsites</i> значений для второго элемента функции и т.д. Внутри блока размещаются значения только тех производных, для которых соответствующая компонента в массиве не равна 0, сначала все производные для первой точки, затем для второй точки и т.д.

<i>MKL_INT* cell</i>	Целочисленный массив с информацией об индексах ячеек. Можно присвоить значение NULL.
----------------------	--------------------------------------------------------------------------------------

Если выполнение процедуры завершилось успешно, возвращаемое значение равно DF_STATUS_OK.

При вычислении значений сплайна в некоторой точке сначала выполняется поиск ячейки, которой принадлежит эта точка. Если для неравномерного разбиения массив отсортирован в порядке возрастания, то поиск ячеек выполняется быстрее. Поэтому если массив упорядочен, то параметру *sit hint* следует присвоить значение DF_SORTED_DATA.

Вычисление интегралов от интерполяционного сплайна

```
int dfdIntegrate1D (DFTaskPtr task, const MKL_INT method, const MKL_INT nlim,
                   const double* llim, const MKL_INT llimhint,
                   const double* rlim, const MKL_INT rlimhint,
                   const double* ldatahint, const double* rdatahint,
                   double* r, const MKL_INT rhint)
```

```
int dfsIntegrate1D (DFTaskPtr task, const MKL_INT method, const MKL_INT nlim,
                   const float* llim, const MKL_INT llimhint,
                   const float* rlim, const MKL_INT rlimhint,
                   const float* ldatahint, const float* rdatahint,
                   float* r, const MKL_INT rhint)
```

Процедуры вычисляют *nlim* интегралов от интерполяционных сплайнов по набору отрезков.

Входные параметры	Описание
<i>DFTaskPtr task</i>	Дескриптор задачи.
<i>const MKL_INT method</i>	В настоящей версии поддерживается только одно значение DF_METHOD_PP
<i>const MKL_INT nlim</i>	Число пар пределов интегрирования.
<i>const double* llim</i> или <i>const float* llim</i>	Массив длиной <i>nlim</i> , который содержит левые концы отрезков интегрирования.
<i>const MKL_INT llimhint</i>	Именованная константа, определяющая структуру массива, который содержит левые концы отрезков интегрирования. Значения из таблицы 1.
<i>const double* rlim</i> или <i>const float* rlim</i>	Массив, который содержит правые концы отрезков интегрирования.

<i>const MKL_INT rlimhint</i>	Именованная константа, определяющая структуру массива, который содержит правые концы отрезков интегрирования. Значения из таблицы 1.
<i>const double* ldatahint</i> или <i>const float* ldatahint</i>	Дополнительная информация о левых концах отрезков интегрирования и разбиении отрезка интерполяции, которая может ускорить вычисления. Можно использовать значение NULL.
<i>const double* rdatahint</i> или <i>const float* rdatahint</i>	Дополнительная информация о правых концах отрезков интегрирования и разбиении отрезка интерполяции, которая может ускорить вычисления. Можно использовать значение NULL.
<i>double* r</i> или <i>float* r</i>	Массив, который содержит вычисленные значения интегралов. Число элементов массива равно <i>ny * nlim</i> . Формат упаковки значений в массив определяется параметром <i>rhint</i> .
<i>const MKL_INT rhint</i>	Именованная константа, которая определяет формат хранения значений интегралов. Для значения DF_MATRIX_STORAGE_ROWS первые <i>nlim</i> элементов – это значения интегралов для первого элемента векторной функции, следующие <i>nlim</i> элементов – это значения для второго элемента векторной функции и т.д.

Если выполнение процедуры завершилось успешно, возвращаемое значение равно DF_STATUS_OK.

Процедуры интегрирования не поддерживают вычисления “in-place” – массивы ***llim***, ***rlim*** и ***r*** не должны пересекаться в памяти.

Деструктор задачи

int dfDeleteTask(DFTaskPtr * task);

Процедура удаляет дескриптор задачи и освобождает память, выделенную для задачи. Возвращаемое значение равно DF_STATUS_OK, если задача удалена успешно, и ненулевой код ошибки в противном случае.

////////////////////////////////////

Обратите внимание, что в методах

int dfdNewTask1D (DFTaskPtr task,...);*

*int dfDeleteTask(DFTaskPtr * task);*

первый параметр имеет тип – указатель на DFTaskPtr.

В остальных методах первый параметр имеет тип DFTaskPtr:

```
int dfdEditPPSpline1D (DFTaskPtr task, ...);  
int dfdConstruct1D (DFTaskPtr task, ...);  
int dfdInterpolate1D (DFTaskPtr task, ...);  
int dfdIntegrate1D (DFTaskPtr task, ...);
```

В методе dfdConstruct1D параметр method может принимать только значение DF_METHOD_STD.

В методах dfdInterpolate1D и dfdIntegrate1D параметр method может принимать только значение DF_METHOD_PP (это разные константы).