Overview of Data Fitting Component in Intel® Math Kernel Library (Intel® MKL)

Intel Corporation

Agenda

- 1D interpolation problem statement
- Computation flow
- Application areas
- Data fitting in Intel® MKL
- Data fitting API and usage model
- Data fitting performance
- Data fitting in applications

Optimization Notice

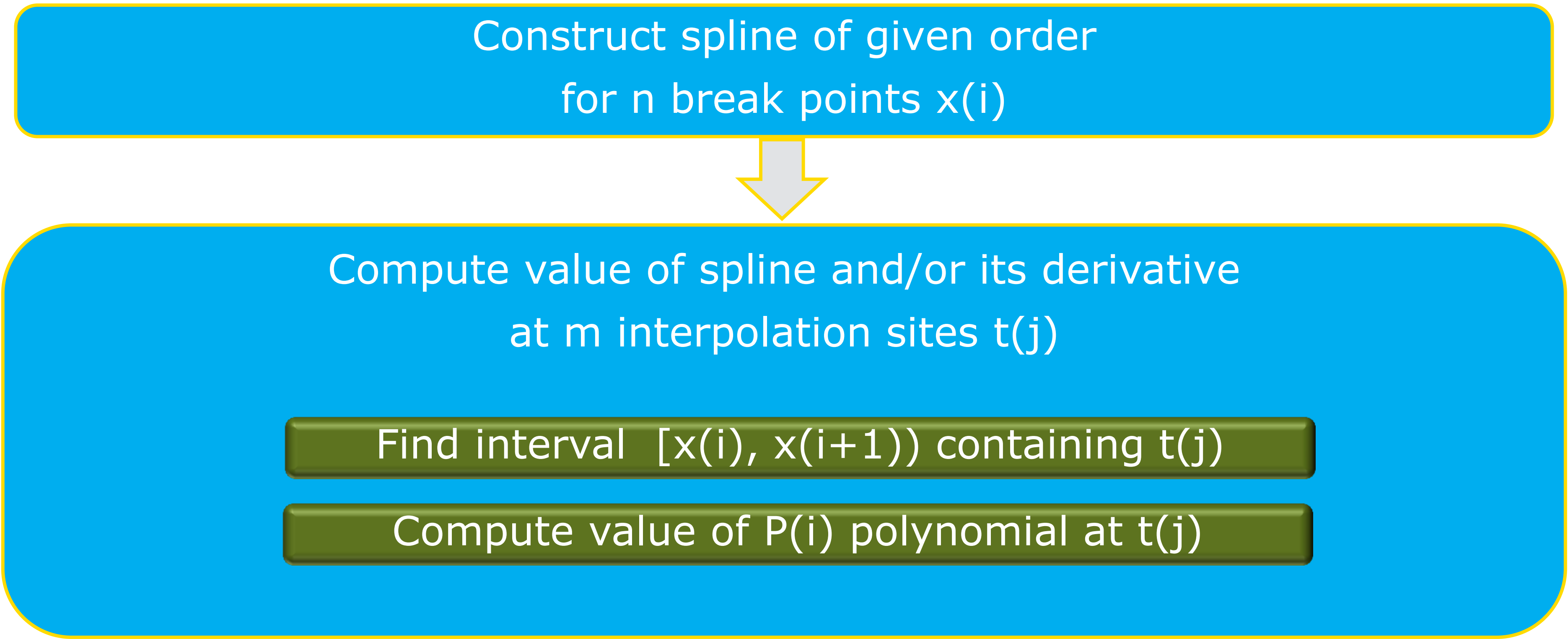
1D Interpolation Problem Statement

- For given table function {x(i), y(i)}, i=1,…,n
  - x(i): breakpoints in ascending order
  - y(i): values
- Approximate function f(x): f(x(i))=y(i)
- Evaluate value f(t(j)) and derivative f’ (t(j))
  - t(j), j=1,…,m: sites fall between or outside of interpolation interval [x(1),x(n)]
- Evaluate integral of f(x) over interval [a(j),b(j))
  - Integration limits a(j) and b(j) fall between or outside of interpolation interval [x(1),x(n)], j=1,…,m

Spline Based 1D Interpolation

- What is a spline?
  - Piecewise polynomial functions
  - g(x) := Pj(x), x belongs to [x(j), x(j+1))
  - Pj(x) - polynomial of degree k on the interval [x(j), x(j+1))
  - smooth up to order q at x(j) if derivatives up to order q for P(j-1) and Pj at x(j) exist and are equal
- Spline based methods are preferable over polynomial interpolation
  - Avoiding Runge’s phenomenon: Interpolation error increases when the order of the polynomial increases

Computation Flow



Construct spline of given order
for n break points x(i)

Compute value of spline and/or its derivative
at m interpolation sites t(j)

Find interval [x(i), x(i+1)) containing t(j)

Compute value of P(i) polynomial at t(j)

- **Integration has similar computational flow**
- **Cell search is the key building block**

Application Areas

- Data analysis and analytics
  - Approximation of statistical estimates like histogram
- Manufacturing
  - Geometrical modeling
  - *“B-spline recurrence relations … were used at Boeing, …, five hundred million times a day”* Carl de Boor, On Wings of Splines Newsletter of Institute for Mathematical Sciences, ISSUE 5 2004
- Energy
  - Surface approximation
- ISV
- Life sciences
  - Molecular dynamics simulation

Data Fitting in Intel® MKL

- Intel® MKL Data Fitting – SW solution for
  - Spline construction
  - Spline based interpolation and computation of derivatives
  - Spline based integration
  - Cell Search
- Current version of Intel® MKL supports 1-dimensional data fitting computations

Spline Construction and Boundary/Internal Conditions

Spline	Spline type
Linear	
Quadratic	Default, Subbotin
Cubic	Default, Natural, Hermite, Bessel, Akima
Look-up	
Stepwise constant	Continuous-right, Continuous-left
User-defined	

Boundary conditions	Internal conditions
Not-a-knot	1st derivative
Free-end	2nd derivative
1st derivative at the left/right endpoint	Knot array
2nd derivative at the left/right endpoint	
Periodic	
Function value at mid point of first cell	

Rich collection of splines that support different boundary or/and internal conditions

Interpolation, Extrapolation, and Integration

Feature	Comment
Computation of values, derivatives of arbitrary order	• Support of *a priori* information about structure of partition, and/or interpolation sites • In addition to default spline based interpolation library supports user-defined functions to • re-define default spline based computations on interpolation or/and extrapolation intervals • re-define cell search functions • Option to get results of cell search simultaneously with interpolation • User defined threading-friendly API
Computation of integrals	• Support of *a priori* info about structure of partition, and/or integration limits • In addition to default spline based interpolation library supports user-defined functions to • re-define default integration on interpolation or/and extrapolation intervals • re-define cell search functions • User defined threading-friendly API

Flexibility in support of various usage models for spline based computations

Optimization Notice

Cell Search

Feature	Comment
Computation of cell indices containing given sites	• Support of *a priori* information about structure of partition, and/or interpolation sites • In addition to default cell search computation library supports user-defined function to • re-define cell search functions • User defined threading-friendly API

Flexibility in support of various usage models for cell search

Data Fitting API and Usage Model

Step	Code example	Comment
Create a task	`status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );`	You can call the Data Fitting function several times to create multiple tasks
Modify the task parameters.	`status = dfdEditPPSpline1D( task, s_order, c_type, bc_type, bc, ic_type, ic, scoeff, scoeffhint );`	
Perform Data Fitting spline-based computations	`status = dfdInterpolate1D(task, estimate, method, nsite, site, sitehint, ndorder, dorder, datahint, r, rhint, cell );`	You may reiterate steps 2-3 as needed
Destroy the task or tasks	`status = dfDeleteTask( &task );`	

Cubic Spline Interpolation Example

- This example can also be found in the online Intel® MKL Reference Manual (link)

```
/* Initialize the partition and set their values */
. . .
. . .
/* Create a Data Fitting task */
status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );

/* Initialize spline parameters */
s_order = DF_PP_CUBIC; /* Spline is of the fourth order (cubic spline). */
s_type = DF_PP_BESSEL; /* Spline is of the Bessel cubic type. */

bc_type = DF_BC_NOT_A_KNOT; bc = NULL; /* Use not-a-knot boundary conditions */

/* Set spline parameters in the Data Fitting task */
status = dfdEditPPSpline1D( task, s_order, s_type, bc_type, bc, ic_type, ic, scoeff, scoeffhint );

/* Construct a cubic Bessel spline: */
status = dfdConstruct1D( task, DF_PP_SPLINE, DF_METHOD_STD );

/* Initialize interpolation parameters and set site values */
. . .
/* Compute the sline values at site(i), i=0,..., nsite-1 and place the results to array r */
status = dfdInterpolate1D(task, DF_INTERP, DF_METHOD_STD,nsite, site, sitehint, ndorder, &dorder,
                           datahint, r, rhint, cell );

/* De-allocate Data Fitting task resources */
status = dfdDeleteTask( &task );
```

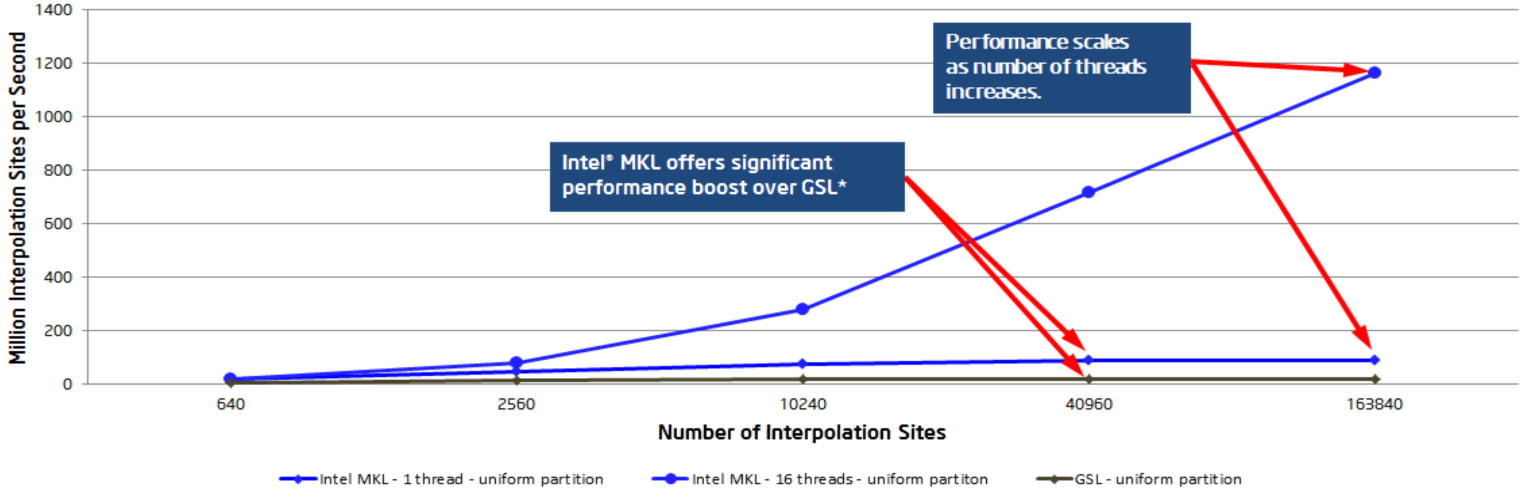
Cell Search Example

- This example can also be found in the online Intel® MKL Reference Manual (link)

```
/* Initialize a uniform partition */
. . .

/* Initialize function parameters; in cell search, function values are not necessary */
ny = 0;
y = NULL;
yhint = DF_NO_HINT;

/* Create a Data Fitting task */
status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );
. . .

/* Initialize interpolation (cell search) parameters */
nsite = NSITE;
/* Set sites in the ascending order */
. . .
sitehint = DF_SORTED_DATA; /* Sites are provided in the ascending order. */
datahint = DF_NO_APRIORI_INFO; /* No additional information about breakpoints/sites is provided.*/

/* Compute indices of the cells that contain interpolation sites.
   Results are stored in cell(i), i=0,...,nsite-1 */
status = dfSearchCell1D( task, DF_METHOD_STD, nsite, site, sitehint, datahint, cell );

/* Delete task and de-allocating resources */
status = dfDeleteTask( &task );
```

Data Fitting Performance: Interpolation

**Performance Improves using Intel® Math Kernel Library versus GSL*
Spline Construction and Interpolation on Intel® Xeon® E5-2600 Family Processor**



Performance scales as number of threads increases.

Intel® MKL offers significant performance boost over GSL*

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 11.0; Hardware: Intel® Xeon® Processor E5-2690, 2 Eight-Core CPUs (20MB LLC, 2.9GHz), 32GB of RAM; Operating System: RHEL 6 GA x86_64; Benchmark Source: Intel Corporation.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm.
* Other brands and names are the property of their respective owners

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

Construction of natural cubic spline with free end boundary conditions for function defined on uniform partition. Partition size is 1280. Spline-based values and first derivatives are computed.

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Copyright © , Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

intel®