Program

```
static const G4double fTable[47][2] = {
   { 0.02, 21.5},
   { 0.03, 20.0},
   { 0.04, 18.0},
   { 0.05, 15.6},
   { 0.06, 15.0},
   { 0.07, 14.0},
   { 0.08, 13.5},
   { 0.09, 13.},
   { 0.1,  12.2},
   { 0.2,  9.25},
   { 0.3,  7.0},
   { 0.4,  6.0},
   { 0.5,  4.5},
   { 0.6,  3.5},
   { 0.7,  3.0},
   { 0.8,  2.5},
   { 0.9,  2.0},
   { 1.0,  1.7},
   { 1.2,  1.2},
   { 1.3,  1.0},
   { 1.4,  0.86},
   { 1.5,  0.7},
   { 1.6,  0.61},
   { 1.7,  0.52},
   { 1.8,  0.5},
   { 1.9,  0.43},
   { 2.0,  0.42},
   { 2.1,  0.3},
   { 2.4,  0.2},
   { 3.0,  0.13},
   { 3.08, 0.1},
   { 3.1,  0.09},
   { 3.3,  0.08},
   { 3.5,  0.07},
   { 3.8,  0.06},
   { 4.0,  0.051},
   { 4.1,  0.04},
   { 4.8,  0.03},
   { 5.0,  0.024},
   { 5.1,  0.02},
   { 6.0,  0.013},
   { 6.5,  0.01},
   { 7.0,  0.009},
   { 7.1,  0.008},
   { 8.0,  0.006},
```

```
   { 9.0,  0.0032},
   { 10.0, 0.0025} };

G4LPhysicsFreeVector* BarkasCorr = new G4LPhysicsFreeVector(47, 0.02, 10.);
for(size_t i=0; i<47; ++i) { BarkasCorr->PutValues(i, fTable[i][0], fTable[i][1]); }
BarkasCorr->SetSpline(true);

size_t N = BarkasCorr->GetVectorLength();
std::vector<double> left(N - 1);
std::vector<double> right(N - 1);
constexpr G4double eps = 1.0e-6;
std::vector<double> left_diff(N - 1);
std::vector<double> right_diff(N - 1);
for(size_t i = 0; i < N - 1; i++)
{
   G4double x1 = BarkasCorr->Energy(i);
   G4double x2 = BarkasCorr->Energy(i + 1);
   G4double h = x2 - x1;
   G4double C = h * h / 6.0;
   G4double y1 = BarkasCorr->Value(x1);
   G4double y2 = BarkasCorr->Value(x2);

   G4double v1 = BarkasCorr->Value(x1 + h * (1.0 / 3.0));
   G4double v2 = BarkasCorr->Value(x1 + h * (2.0 / 3.0));
   v1 -= (2.0 / 3.0) * y1 + (1.0 / 3.0) * y2;
   v2 -= (1.0 / 3.0) * y1 + (2.0 / 3.0) * y2;
   v1 *= -27.0 / (2.0 * C); // v1 = 5 * M1 + 4 * M2
   v2 *= -27.0 / (2.0 * C); // v2 = 4 * M1 + 5 * M2
   G4double M1 = (5.0 * v1 - 4.0 * v2) / 9.0;
   G4double M2 = (-4.0 * v1 + 5.0 * v2) / 9.0;

   left[i] = (-2.0 * M1 * C - M2 * C - y1 + y2) / h;
   right[i] = (M1 * C + 2.0 * M2 * C - y1 + y2) / h;

   left_diff[i] = (BarkasCorr->Value(x1 + 2.0 * eps * h) -
   BarkasCorr->Value(x1 + 1.0 * eps * h)) / (eps * h);
   right_diff[i] = (BarkasCorr->Value(x2 - 1.0 * eps * h) -
   BarkasCorr->Value(x2 - 2.0 * eps * h)) / (eps * h);
}

for(size_t i = 0; i < N - 2; i++)
{
   G4cout << i << "\t";
   G4cout << right[i] << " (" << right_diff[i] << ")" << "\t";
   G4cout << left[i + 1] << " (" << left_diff[i + 1] << ")" << G4endl;
}
```

Output

```
0   -126.973 (-126.973) -176.973 (-176.973)
1   -246.054 (-246.054) -252.304 (-252.305)
2   -146.309 (-146.31)  -146.872 (-146.871)
3   -61.3333 (-61.3331) -61.1083 (-61.1086)
4   -88.2449 (-88.2451) -88.2526 (-88.2524)
5   -35.8967 (-35.8967) -35.8962 (-35.8963)
6   -68.1616 (-68.1615) -68.1616 (-68.1616)
7   -81.4575 (-81.4576) -81.0908 (-81.0906)
8   -15.552 (-15.552)   -12.5173 (-12.5174)
9   -18.7705 (-18.7706) -18.4716 (-18.4715)
10  -10.4985 (-10.4985) -10.5165 (-10.5165)
11  -14.4985 (-14.4985) -14.4978 (-14.4977)
12  -6.49111 (-6.49111) -6.49114 (-6.49114)
13  -4.53774 (-4.53774) -4.53774 (-4.53774)
14  -5.35788 (-5.35788) -5.35788 (-5.35789)
15  -4.03072 (-4.03072) -4.03072 (-4.03071)
16  -2.51924 (-2.51924) -2.53142 (-2.53142)
17  -2.29875 (-2.29876) -2.299 (-2.299)
18  -1.58755 (-1.58755) -1.59443 (-1.59443)
19  -1.53705 (-1.53705) -1.5344 (-1.5344)
20  -1.26268 (-1.26268) -1.26281 (-1.2628)
21  -0.914626 (-0.914625)   -0.914605 (-0.914606)
22  -0.47873 (-0.478733)    -0.478731 (-0.478728)
23  -0.470473 (-0.47047)    -0.470473 (-0.470475)
24  -0.339378 (-0.339381)   -0.339378 (-0.339375)
25  -0.572015 (-0.572011)   -0.572015 (-0.572018)
26  -1.27256 (-1.27257) -1.26039 (-1.26038)
27  0.0722015 (0.0722018)   0.0877939 (0.0877933)
28  -0.293618 (-0.293617)   -0.30406 (-0.30406)
29  -0.496955 (-0.496955)   -0.495536 (-0.495537)
30  -0.465685 (-0.465685)   -0.465407 (-0.465404)
31  0.0498697 (0.0498703)   0.0759281 (0.0759274)
32  -0.086189 (-0.0861891)  -0.0820991 (-0.0820988)
33  -0.016127 (-0.016127)   -0.0150399 (-0.01504)
34  -0.0953514 (-0.0953512) -0.0951547 (-0.0951548)
35  -0.104319 (-0.104319)   -0.103674 (-0.103673)
36  -0.0189563 (-0.018956)  -0.0099065 (-0.00990658)
37  -0.0419786 (-0.0419786) -0.0407022 (-0.0407022)
38  -0.0366638 (-0.0366638) -0.036653 (-0.0366529)
39  -0.00375953 (-0.00375947)   -0.00138904 (-0.00138907)
40  -0.00364484 (-0.00364487)   -0.00332772 (-0.00332769)
41  -0.00866584 (-0.00866579)   -0.0085481 (-0.00854811)
42  -0.00997192 (-0.00997193)   -0.00987357 (-0.00987353)
43  -0.000492008 (-0.000491995) -4.56124e-05 (-4.56264e-05)
44  -0.00365219 (-0.0036522)    -0.00365219 (-0.00365219)
```

Formulas description.

To calculate derivatives of cubic spline at its knots analytically we can do the following. On interval $[x_i, x_{i+1}]$ cubic polynomial y(x) could be set by:

$$y(x) = M_i C(b^3 - b) + M_{i+1} C(a^3 - a) + by_i + ay_{i+1}, \quad (1)$$

where $M_i = y''(x_i)$,

$$a = (x - x_i)/h,$$
$$b = (x_{i+1} - x)/h,$$
$$C = h^2/6,$$
$$h = x_{i+1} - x_i > 0.$$

At point $x = x_i$: $a = 0, b = 1$, => $y(x_i) = y_i$.

At point $x = x_{i+1}$: $a = 1, b = 0$, => $y(x_{i+1}) = y_{i+1}$.

$$a' = 1/h,$$
$$b' = -1/h.$$

First derivative:

$$y'(x) = \frac{M_i C(1 - 3b^2) + M_{i+1} C(3a^2 - 1) - y_i + y_{i+1}}{h}.$$

$$y'(x_i) = \frac{-2M_i C - M_{i+1} C - y_i + y_{i+1}}{h},$$

$$y'(x_{i+1}) = \frac{M_i C + 2M_{i+1} C - y_i + y_{i+1}}{h}.$$

Second derivative:

$$y''(x) = M_i b + M_{i+1} a.$$
$$y''(x_i) = M_i,$$
$$y''(x_{i+1}) = M_{i+1}.$$

So, function y(x) and y''(x) in form (1) is automatically continuous at points x_i. Let's choose intermediate points on (x_i, x_{i+1}):

$$x'_i = x_i + {}^1/_3\,(x_{i+1} - x_i) \qquad \text{(a=1/3, b=2/3)}$$

$$x''_i = x_i + {}^2/_3\,(x_{i+1} - x_i) \qquad \text{(a=2/3, b=1/3)}$$

For these points

$$y(x'_i) = -{}^{10}/_{27}\,M_i C - {}^8/_{27}\,M_{i+1} C + {}^2/_3\,y_i + {}^1/_3\,y_{i+1}$$

$$y(x''_i) = -{}^8/_{27}\,M_i C - {}^{10}/_{27}\,M_{i+1} C + {}^1/_3\,y_i + {}^2/_3\,y_{i+1}$$

If we can calculate values $y(x'_i)$ and $y(x'_i)$, we can calculate M_i and M_{i+1} and, correspondingly, $y'(x_i)$ and $y'(x_{i+1})$.