JSpline1D.cc

The JSpline1D.cc program can be used to test the spline interpolation of a Gaussian function. The Gaussian function is taken from ROOT.

```
/**
 * Function g1().
 */
inline Double_t g1(const Double_t x)
{
  return TMath::Gaus(x, 0.0, 1.0, kTRUE);
}
```

With the following expression for its integral.

```
/**
 * Integral of g1().
 */
inline Double_t G1(const Double_t x)
{
  if (x < 0.0)
    return 0.5 * (TMath::Erfc(-sqrt(0.5)*x));
  else
    return 0.5 * (1.0 + TMath::Erf(sqrt(0.5)*x));
}
```

Definition and construction of spline iterpolator.

```
  typedef  JSplineFunction1D_t    JFunction1D_t;

  JFunction1D_t f1;
  JFunction1D_t F1;

  const Double_t xmin = -5.5;
  const Double_t xmax = +5.5;

  for (Double_t x = xmin; x <= xmax; x += (xmax - xmin) / numberOfPoints) {
    f1[x] = g1(x);
  }

  f1.compile();

  convertToIntegral(f1, F1);
```

The last statement directly converts the function object `f1` to its integral `F1`. The number of points can be specified on the command line (option `-n`). Hence the result can be checked for different number of points.

Test of spline interpolation.

```
  TH1D h0("h0", NULL, nx,  xmin, xmax);
  TH1D h1("h1", NULL, nx,  xmin, xmax);

  TH1D i0("i0", NULL, nx,  xmin, xmax);
  TH1D i1("i1", NULL, nx,  xmin, xmax);

  for (int i = 1; i <= h0.GetNbinsX(); ++i) {

    const Double_t x = h0.GetBinCenter(i);

    h0.SetBinContent(i, g1(x));
    i0.SetBinContent(i, G1(x));

    try {
      h1.SetBinContent(i, f1(x));
      i1.SetBinContent(i, F1(x));
    }
    catch(const exception& error) {
      ERROR(error.what() << endl);
    }
  }
```

Typical output is:

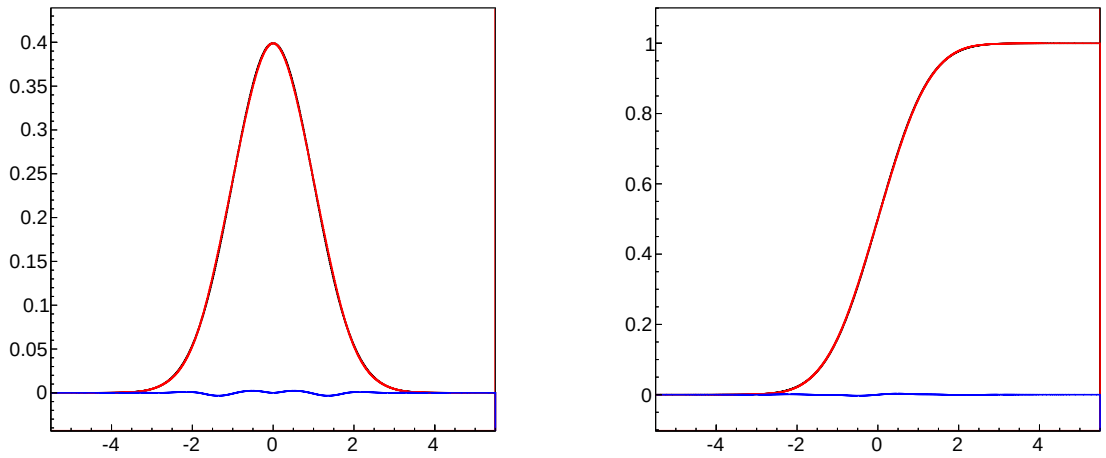


Figure 1: Left: Gauss function (analytical black; interpolated red; and differenc blue). Right: Erfc function (analytical black; interpolated red; and difference blue).