Interpolacja funkcjami sklejanymi

Michał Saturczak
Informatyka Stosowana, Akademia Górniczo-Hutnicza
Metody Numeryczne

19 kwietnia 2024

1 Wstęp teoretyczny

W niniejszym sprawozdaniu omówiona zostanie metoda interpolacji funkcjami sklejanymi, która rozpoczyna się od wyznaczenia wartości drugich pochodnych w węzłach funkcji. Wartości te oznaczane są jako m_i, gdzie indeks i oznacza numer węzła, dla którego wyznaczane są wartości. W tym przypadku nie można zastosować iloczynu różnicowego, jednak cały proces sprowadza się do rozwiązania pewnego układu równań liniowych.

Układ ten ma specyficzną strukturę na początku i na końcu, co jest związane z problemami, które mogą wystąpić na brzegach w przypadku niektórych zagadnień, takich jak transport ciepła. W celu rozwiązania tych problemów, przyjmuje się pewne wartości brzegowe dla drugiej pochodnej, oznaczone jako α i β. W tym przypadku, ze względu na brak możliwości ich wyznaczenia, przyjmowane są one jako równe 0.

Pozostałe parametry w układzie równań to odległość między węzłami $h_i = x_i - x_{i-1}$, wartość d_i zależna od wartości funkcji w węzłach i odległości między nimi, oraz λ i μ_i zależne od odległości między węzłami. Wartości x_i to położenia znanych węzłów funkcji, a y_i to wartość funkcji w poszczególnych węzłach.

W omawianym podejściu, interpolacja funkcjami sklejanymi pozwala na stworzenie gładkiej krzywej, która przechodzi przez zadane punkty, jednocześnie minimalizując efekt oscylacji, który często występuje w przypadku innych metod interpolacji.

2 Problem

2.1 Rozwiązanie problemu

- Napisano program w języku C++ do rozwiązania problemów związanych z Interpolacją
- Zapisano do trzech skoroszytów dane odpowiednio dla dwóch funkcji z krokiem co 0.1 dla wartości $x \in [-5, 5]$, oraz dla wartości $n = 5, 10, 15, 20$.
- Zrobiono wykresy aktualne funkcji oraz ich interpolacje.

2.2 Funkcje matematyczne dla których obliczono interpolacje funkcjami sklejanymi

- $f(x) = e^{-x^2}$
- $f(x) = -1$ dla $x < 0, f(x) =$ dla $x \geq 0$
- $f(x) = \cos 2x$

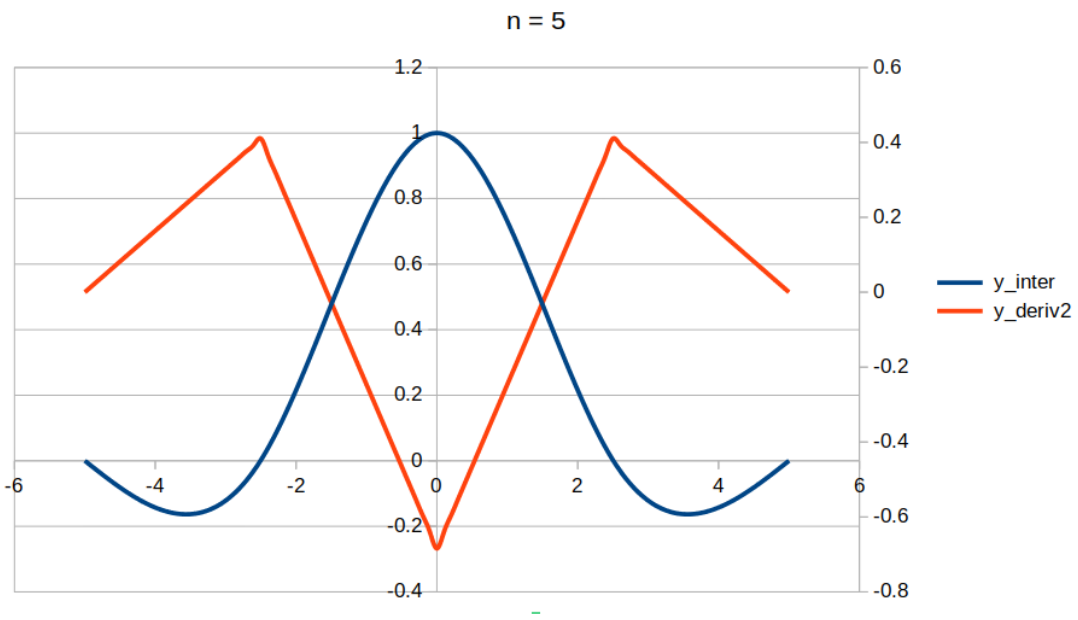
2.3 Funkcje w języku C++ użyte do generowania węzłów interpolacji (z pomocą biblioteki GSL)

```
void interpolation(std::function<double(double)> f, int n, const std::string &func_name)
{
    std::string filename = func_name + ".csv";
    std::ofstream file(filename);

    double xi, yi, x_interp, y_interp, y_deriv2;
    double x_min = -5.0, x_max = 5.0;
    double h = (x_max - x_min) / (n - 1);

    std::vector<double> x(n), y(n), y2(n);
    for (int i = 0; i < n; i++)
    {
        xi = x_min + i * h;
        x[i] = xi;
        y[i] = f(xi);
    }

    gsl_spline *spline = gsl_spline_alloc(gsl_interp_cspline, n);
    gsl_spline_init(spline, x.data(), y.data(), n);
    gsl_interp_accel *acc = gsl_interp_accel_alloc();

    for (int i = 0; i < n; i++)
    {
        y2[i] = gsl_spline_eval_deriv2(spline, x[i], acc);
    }

    file << "x_interp\t y_inter\t y_deriv2\t\n";
    for (x_interp = x_min; x_interp <= x_max; x_interp += 0.1)
    {
        y_interp = gsl_spline_eval(spline, x_interp, acc);
        y_deriv2 = gsl_spline_eval_deriv2(spline, x_interp, acc);
        file << x_interp << "\t" << y_interp << "\t" << y_deriv2 << "\t\n";
    }

    gsl_spline_free(spline);
    gsl_interp_accel_free(acc);
}
```

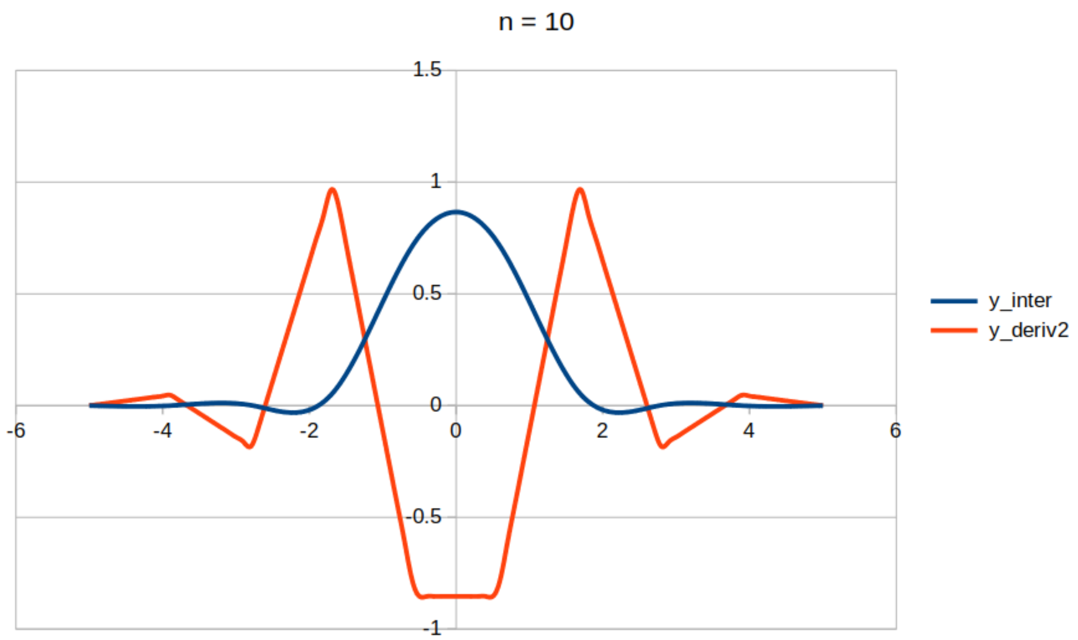
3 Wyniki

3.1 Wykresy interpolacji Lagrange'a dla funkcji $f(x) = e^{-x^2}$

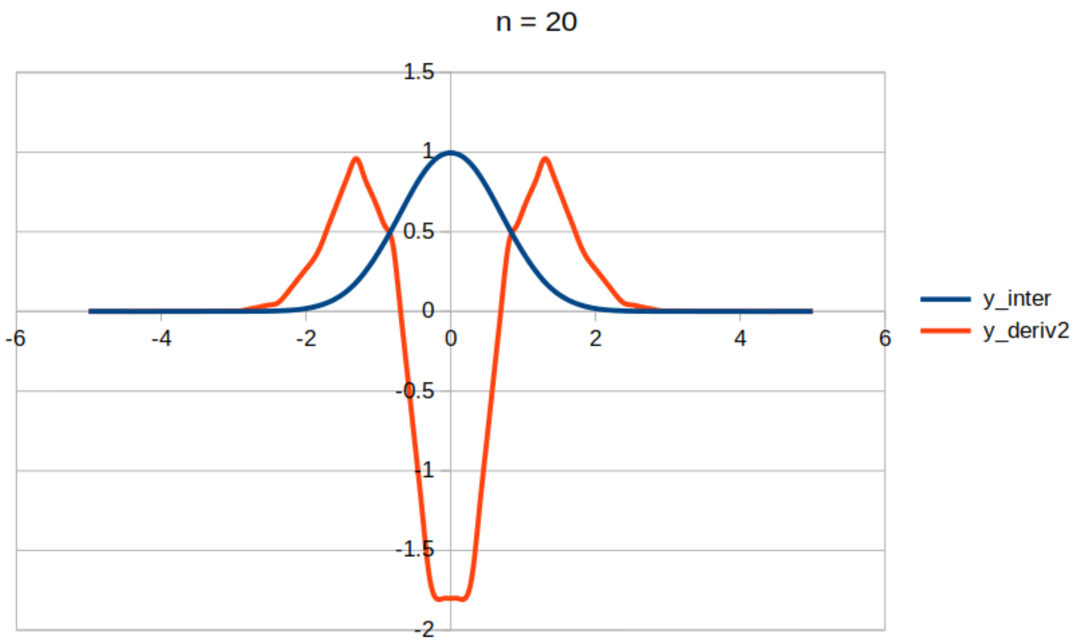
3.1.1 $n = 5$



3.1.2 $n = 10$

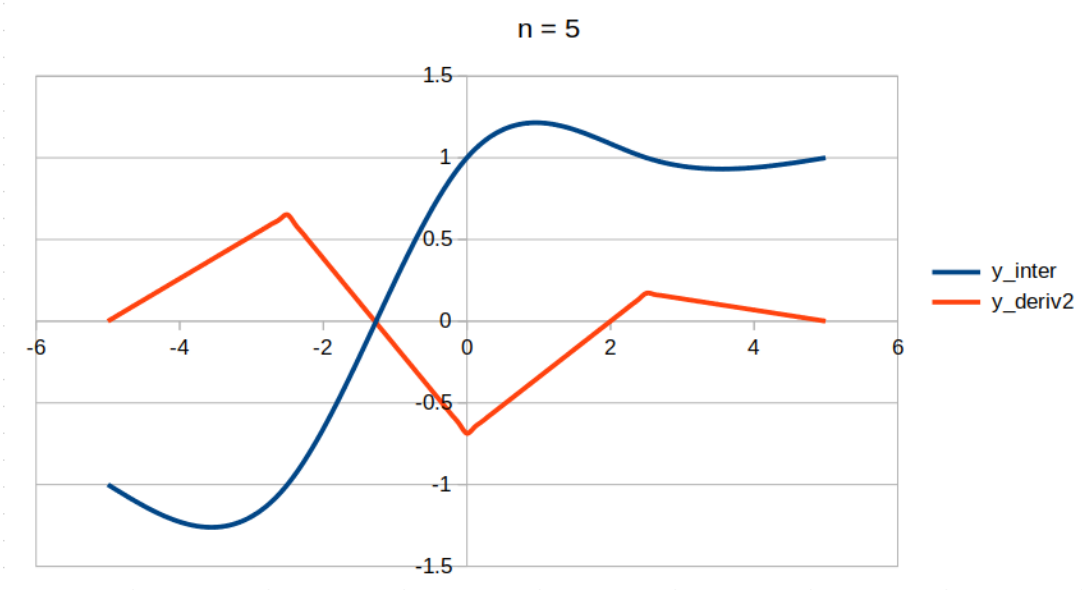


3.1.3 $n = 20$

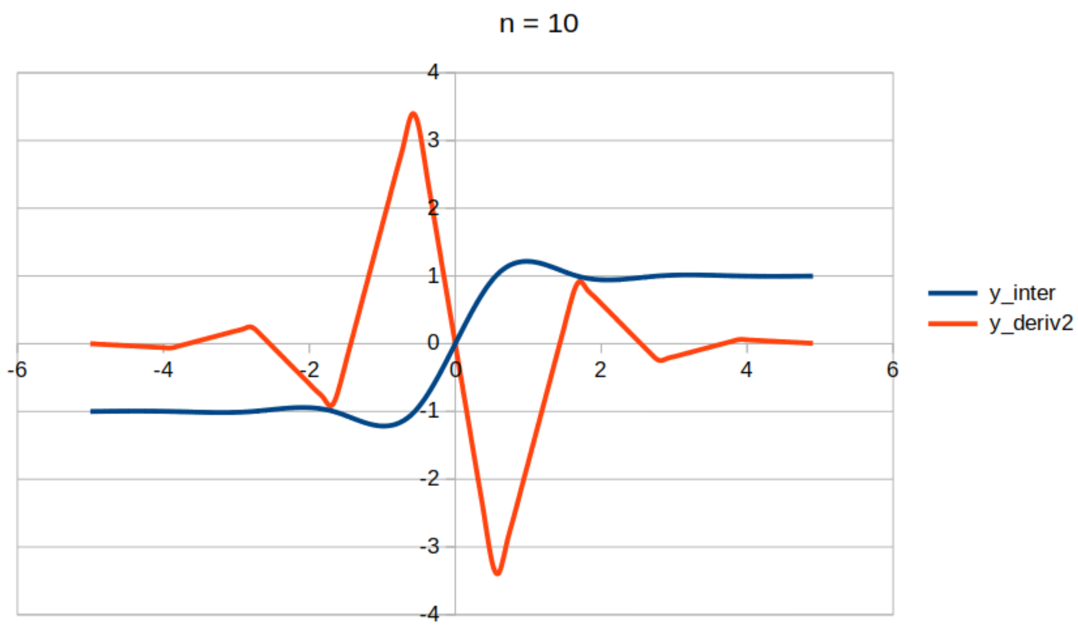


3.2 Wykresy interpolacji funkcjami sklejanymi dla funkcji $f(x) = -1$ dla $x < 0$, $f(x) =$ dla $x \geq 0$

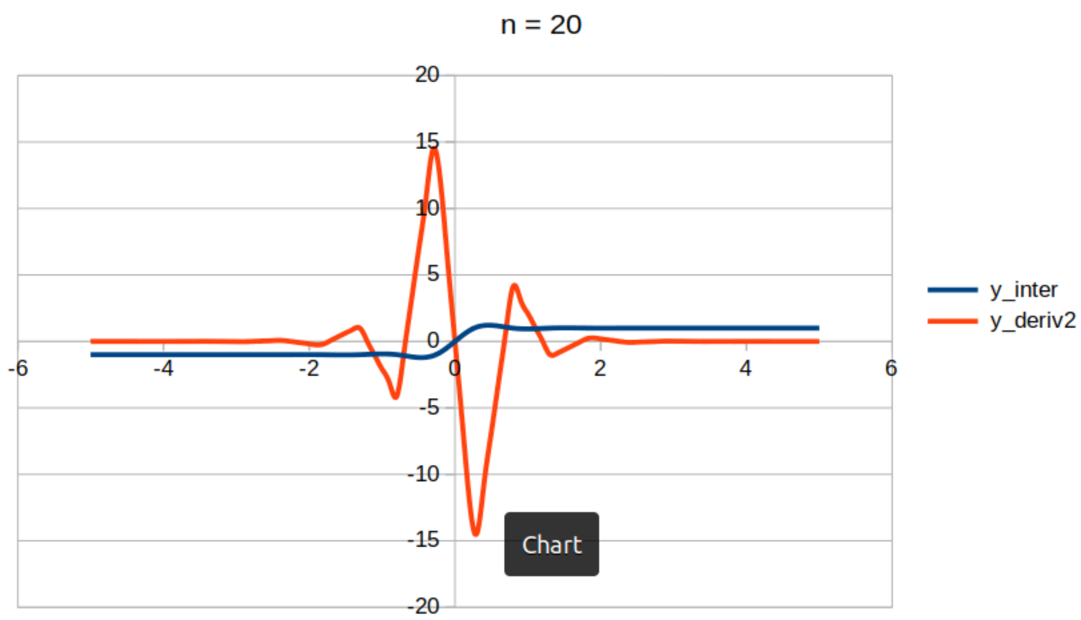
3.2.1 $n = 5$



3.2.2 $n = 10$

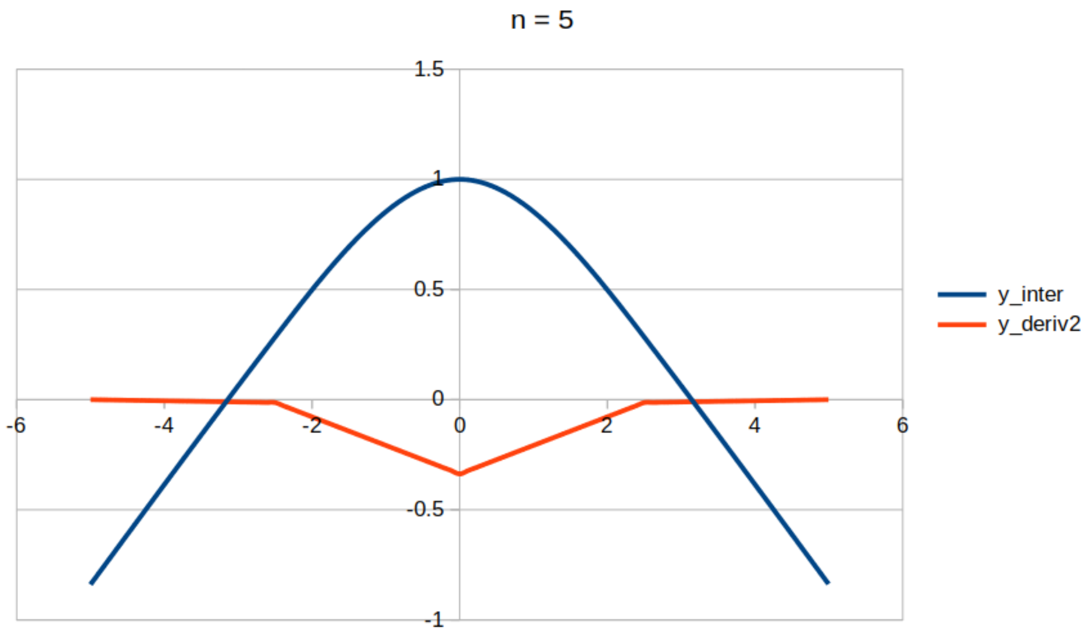


3.2.3 $n = 20$

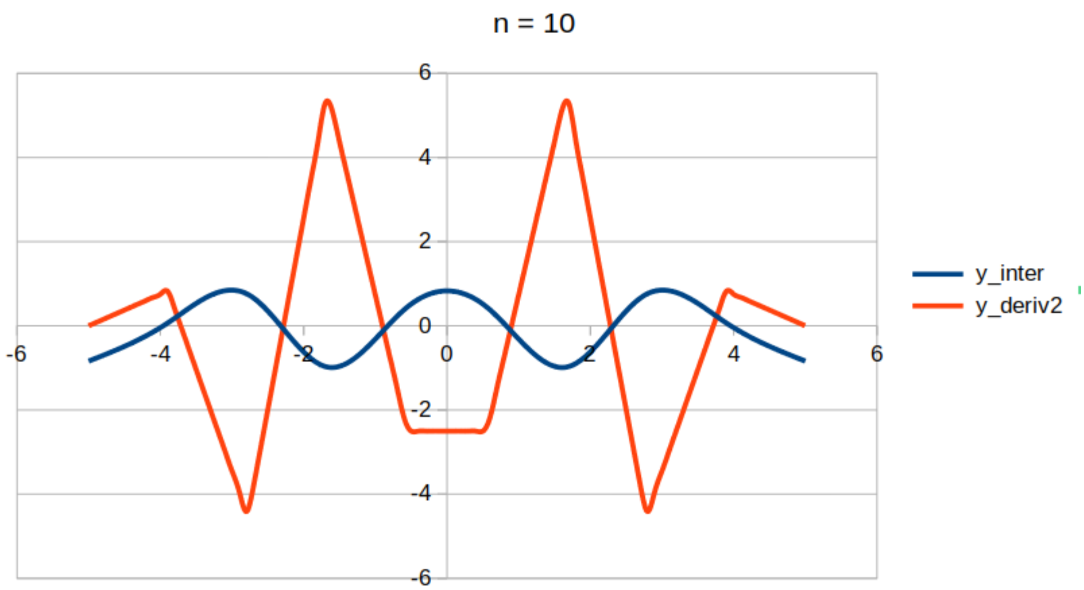


3.3 Wykresy interpolacji funkcjami sklejanymi dla funkcji $f(x) = \cos 2x$

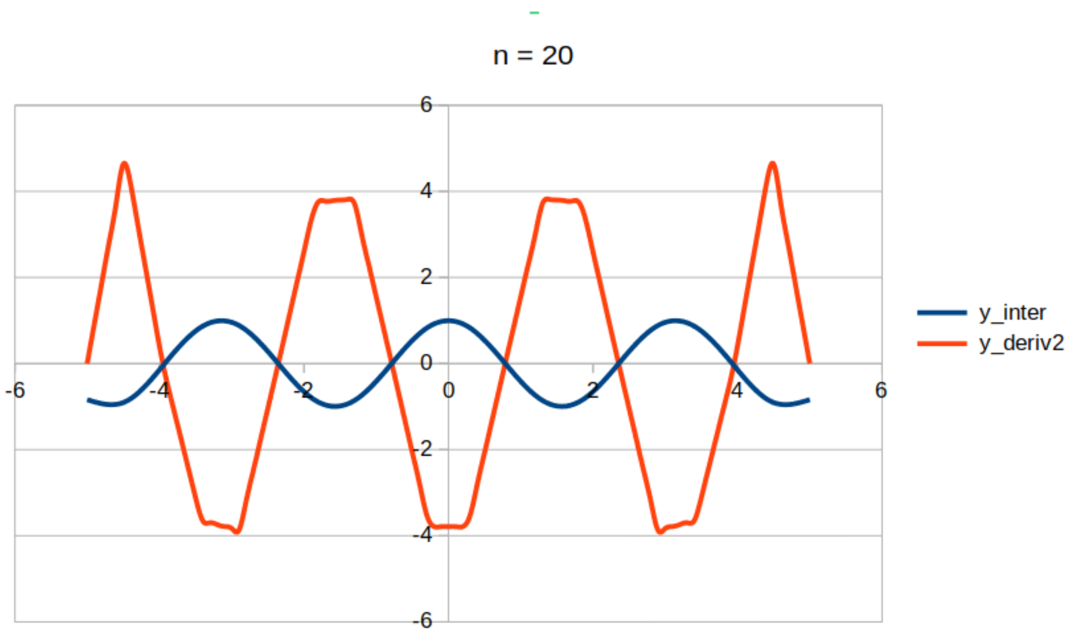
3.3.1 $n = 5$



3.3.2 $n = 10$



3.3.3 $n = 20$



4 Wnioski

Stwierdzenie, że omawiana metoda nie jest tak precyzyjna jak metoda Lagrange'a, wydaje się być uzasadnione. Obserwuje się, że wartości dla kolejnych n zbliżają się do wartości określonych przez funkcję, lecz nie są one identyczne, jak to miało miejsce w przypadku interpolacji funkcji Lagrange'a. Mimo to, niekiedy metoda ta wykazuje większą skuteczność niż metoda Lagrange'a.