Monday 7:00 AM Class

Homework: ode_test.cpp, interpolation except for Ch 5. Include:
Using GSL interpolation functions, split function set.

How far did you get on session 9?
* Allow some time to finish or at least try out the pendulum.nb notebook. Most important.
* ode_test.cpp is a test code with an adaptive
  adaptive solver from GSL (the test code is based on it, mind.
  Change b automatically to keep the local error
  under control.
  Solves Van der Pol oscillator

\[
\frac{dx}{dt} + \mu \frac{dx}{dt} (x^2 - 1) + x = 0
\]

Tag \( \mu = 2 \) with initial conditions \( x_0 = 1.0, v_0 = 0.0 \) ode_test.dat
\( x_0 = 0.1, v_0 = 0.0 \) ode_test.dat
\( x_0 = -1.5, v_0 = 2.0 \) ode_test.dat

* Compile and link as described in code
* ode_test > ode_test1.dat (for \( x_0 = 1.0, v_0 = 0.0 \))
* ode_test > ode_test2.dat (for \( x_0 = 0.1, v_0 = 0.0 \))
* ode_test > ode_test3.dat (for \( x_0 = -1.5, v_0 = 2.0 \))

* Use gnuplot with columns 2, 3 (phase space) to
  see the "limit cycle" \( \Rightarrow \) isolated attractor.
Interpolation vs. Data Fitting

Our basic interpolation problem will be to take a table of function values \((x_i, y_i)\) and estimate \(f(x)\) for \(x \neq x_i\).

Applications:
1. We want to calculate \(\int f(x) \, dx\) using Gaussian quadrature.
2. We want the derivative (or 2nd derivative) of \(f\).
3. We want to solve an ODE with a GSL routine.

The assumption here is that the values are not noisy (although they will have roundoff noise).

So assume that in between \(x_i\) and \(x_{i+1}\), \(f(x)\) looks like a polynomial.

\[\Rightarrow\text{ What polynomial?}\]

### Lagrange Interpolation

- If you know all of the data is polynomial-like, then use high-order polynomial.

Lagrange: \((n-1)\text{-th order polynomial to } f(x)\text{) for } n\text{ values of } x_i\)

- Good to know the formulas.
- Apply only to small region.

5.5 Assessment leading to Breg-Wigner Fit - Fig. 5.1

- If not noisy, successive low-order Lagrange works fine.
- If noisy, use expected form (model) and fit parameters last guess.
\[ P_i(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \]

\[ P(x) = P_i(f(x)) \]

\[ f(x) = f_i(x) \]

\[ \text{Let computer figure out } P_i(x) \]

- requires function fits at \( x_i \) and \( 1^{st}, 2^{nd}, 3^{rd} \) derivatives continuous with next interval
- boundary conditions needed for \( f^{(3)}(0) \) and \( f^{(3)}(x) \) at endpoints.

- "natural spline" \( f^{(3)}(a) = f^{(3)}(b) = 0 \)

\[ \Rightarrow \text{get } 1^{st}, 2^{nd} \text{ derivatives as well, or do cubic spline quadratically} \]

\[ \Rightarrow \text{Go through handout} \]

- How good is the fit? (Address this on Wednesday)
- Ask about least square fitting

\[ \chi^2 = \sum_{i=1}^{N} \left( \frac{y_i - f(x_i; \{a_n\})}{\sigma_i} \right)^2 \]

\[ \text{linear if } f(x; \{a_n\}) \text{ are linear in the } a_i \]'s.

\[ \text{eg. } b_0 + b_1 x + b_2 x^2 = y(x) \]