Data Fitting via Interpolation

data fitting = art worth serious study (not here)

now: interpolate within a table

later: least-squares fit to data (search, matrices)

Rubin H Landau

With

Sally Haerer

Computational Physics for Undergraduates

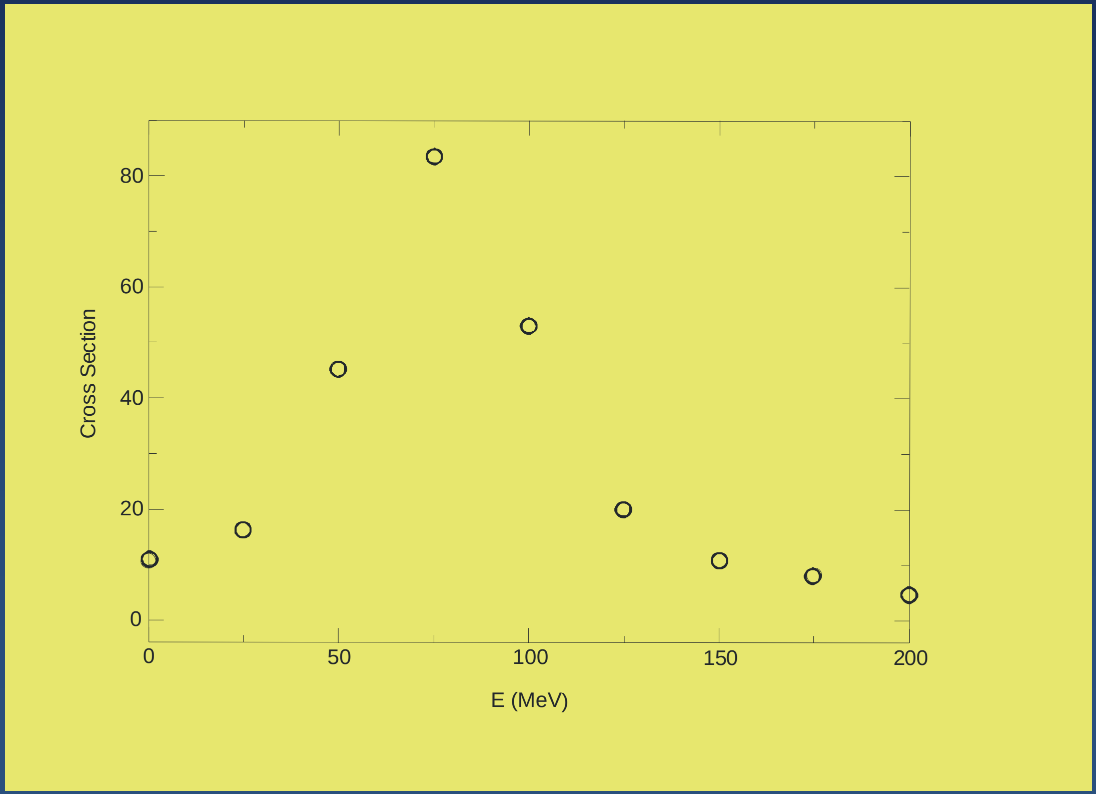
BS Degree Program: Oregon State University

"*Engaging People in Cyber Infrastructure*"

Support by EPICS/NSF & OSU

Problem: Fit (Interpolate) Data

i	1	2	3	4	5	6	7	8	9
E_i	0	25	50	75	100	125	150	175	200
$f(E_i)$	10.6	16.0	45.0	83.5	52.8	19.9	10.8	8.25	4.70
§ σ_i	9.34	17.9	41.5	85.5	51.5	21.5	10.8	6.29	4.14



- *i* denotes point #
- What's *f(E)* between i values?
- 1st interpolate
- "Fit" better in noisy (later)
- Local not global fit
- Table $\Rightarrow$ *f(E)*

Lagrange Interpolation (Method)

1. Independent variable = *x* (*E* in table)
2. Tabulated values x_i, (*i* = 1, 2, ...)
3. Dependent variable = function *g(x)* (*f(E)* in table)
4. Tabulated values $g_i = g(x_i)$
5. g(x) ≈ n-1 degree polynomial in *i* (*n* params)

$$g_i(x) \simeq a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}, \qquad (x \simeq x_i).$$

6. Local fit ⇒ different polynomials, different intervals
7. One high-order polynomial = dangerous oscillations