Analysis Methods in Atmospheric and Oceanic Science

AOSC 652

Numerical Integration

Week 6, Day 2

5 Oct 2016

Numerical Integration

Also known as "quadrature"

In numerical analysis, a quadrature rule is a method for evaluating a definite integral of a function, usually stated as a weighted sum of the function values evaluated a specified points within the domain of integration.

Many of the classic algorithms assume that the value of the function is known at *equally spaced points*.

If a climate model is devised using altitude as the vertical coordinate, Then integrals wrt height could possibly be evaluated using a classic algorithms (i.e., Simpson's Rule).

If a climate model is devised using pressure as the vertical coordinate, then integrals wrt height can be must be evaluated using a technique designed for *unequally spaced points* (Trapezoidal Rule, Gaussian Integration)

Trapezoidal Rule:

$$\int_{x_1}^{x_N} f(x)\,dx \;\approx\; \sum_{i=1}^{N-1} \frac{(x_{i+1} - x_i)}{2}\,\left[f(x_{i+1}) + f(x_i)\right]$$

Will work for evenly OR unevenly spaced grid and any *N*

Simpson's Rule:

$$\int_{x_1}^{x_N} f(x)\,dx \;\approx\; \sum_{i=1,\;3,\;5}^{N-2} \frac{h}{3}\,\left[f(x_{i+2}) + 4f(x_{i+1}) + f(x_i)\right]$$

where $h = (x_N - x_1)\,/\,N$

Classic method will work only for *evenly spaced grid* and odd *N*

AOSC 652: Analysis Methods in AOSC

Simpson's Rule: N must be odd (or need work around); based on 2nd order polynomials
Error order 4th derivative of *f*

$$\int_{x_1}^{x_N} f(x)\, dx \;\approx\; \sum_{i=1,\,3,\,5}^{N-2} \frac{h}{3}\left[f(x_{i+2}) + 4 f(x_{i+1}) + f(x_i) \right]$$

where $h = (x_N - x_1)\,/\,N$

Simpson's 3/8 Rule: N must be evenly divisible by 4; also based on 3d order polynomials
Error order 4th derivative of *f*

$$\int_{x_1}^{x_N} f(x)\, dx \;\approx\; \sum_{i=1,\,4,\,6}^{N-3} \frac{3h}{8}\left[f(x_{i+3}) + 3 f(x_{i+2}) + 3 f(x_{i+1}) + f(x_i) \right]$$

where $h = (x_N - x_1)\,/\,N$

Boole's Rule: N must be evenly divisible by 5; based on 4th order Polynomials;
Error order 6th derivative of *f*

$$\int_{x_1}^{x_N} f(x)\, dx \;\approx\; \sum_{i=1,\,5,\,9}^{N-4} h\, \frac{14 f(x_{i+4}) + 64 f(x_{i+3}) + 24 f(x_{i+2}) + 64 f(x_{i+1}) + 14 f(x_i)}{45}$$

where $h = (x_N - x_1)\,/\,N$

Numerical Integration

Gaussian Quadrature:

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f(g(y))\,\frac{dx}{dy}\,dy \approx \frac{1}{2}(b-a)\sum_{i=1}^{n} c_i f(g(y_i))$$

$y_1, y_2, \ldots y_n$ **nodes** are not uniformly spaced
$c_1, c_2, \ldots c_n$ **Gauss coefficients** are determined once ***n*** is specified

Note:

$$\int_{-1}^{1} f(y)\,dy \approx f\left(\frac{-\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

produces exact result for polynomials of degree 3 or less

Nice descriptions of theory at http://en.wikipedia.org/wiki/Gaussian_quadrature
and http://www.efunda.com/math/num_integration/num_int_gauss.cfm

Numerical Integration

Please copy files:

~rjs/aosc652/week_06/data_generate.f and
~rjs/aosc652/week_06/data_integrate.f

to your work area.

Numerical Integration

We're going to use four "analytic functions" to generate "data files" of Y vs X, which we will then integrate:

$$Integral\ 1 = \int_0^1 Function\ 1\ (x)\ dx = \int_0^1 3\ x^2 dx$$

$$Integral\ 2 = \int_0^1 Function\ 2\ (x)\ dx = \int_0^1 5\ x^4 dx$$

$$Integral\ 3 = \int_0^1 Function\ 3\ (x)\ dx = \int_0^1 9\ x^8 dx$$

$$Integral\ 4 = \int_0^1 Function\ 4\ (x)\ dx = \int_0^1 13\ x^{12} dx$$

What do these functions look like?

Compile code data_generate.f, run using "100" for number of intervals, and plot resulting functions using:

Black for Function 1; Blue for Function 2;
Green for Function 3; Red for Function 4

Numerical Integration: Let's fill in the following tables:

a) Run program **data_generate.e** to generate **"fn_for_integration*.dat"** files, for 25, 10, and 6 intervals (in addition to the file for 100 intervals you should have already generated)

b) Compile code **data_integrate.f**

c) Run program **data_integrate.e** to complete table

Numerical Integration: Let’s fill in the following tables:

Trapezoidal Rule:

Function	100 intervals	25 intervals	10 intervals	6 intervals
$3\ x^2$				
$5\ x^4$				
$9\ x^8$				
$13\ x^{12}$				

Simpson’s Rule:

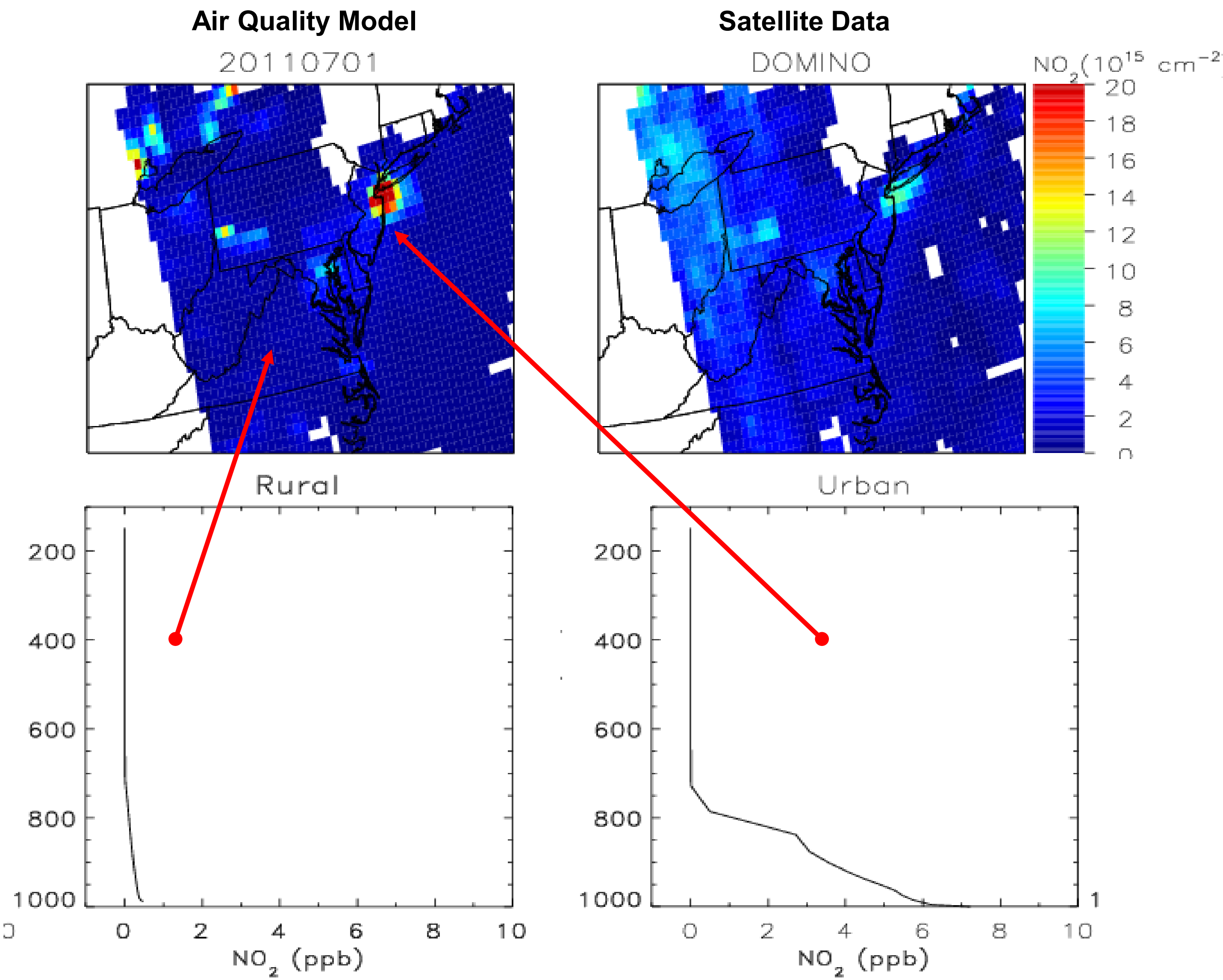
Function	100 intervals	25 intervals	10 intervals	6 intervals
$3\ x^2$				
$5\ x^4$				
$9\ x^8$				
$13\ x^{12}$				

Numerical Integration

Let's conduct a graphical analysis
to investigate behavior of integration
by Trapezoidal rule and Simpson's rule:

- **a) focus first on the 6 interval calculation of function = $13x^{12}$**
- **b) what should we plot ?!?**

For which model NO_2 profile did Tim check the details of his integration scheme ?



Numerical Integration

Gaussian Quadrature:

$$\int_{-1}^{1} f(y)\,dy \approx \sum_{i=1}^{n} c_i f(y_i)$$

Nice description of theory at http://en.wikipedia.org/wiki/Gaussian_quadrature

Numerical Integration

Gaussian Quadrature:

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f(g(y))\,\frac{dx}{dy}\,dy \approx \sum_{i=1}^{n} c_i f(g(y_i)) \bullet \left(\frac{dx}{dy}\right)$$

$g(y)$ is a linear function: i.e., g(y)=Intercept + Slope •y,

that relates $x = a$ to $y = -1$ and $x = b$ to $y = 1$

To use Gaussian Quadrature
for the evaluation of integrals with arbitrary limits,
must execute a Variable Transformation

Numerical Integration

Gaussian Quadrature:

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f(g(y))\,\frac{dx}{dy}\,dy \approx \frac{1}{2}(b-a)\sum_{i=1}^{n} c_i f(g(y_i))$$

$y_1, y_2, \ldots y_n$ **nodes** are not uniformly spaced
$c_1, c_2, \ldots c_n$ **Gauss coefficients** are determined once ***n*** is specified

Location of nodes y_i (places where the function is evaluated) and weights c_i both vary as a function of n

See http://en.wikipedia.org/wiki/Gaussian_quadrature

Numerical Integration

Gaussian Quadrature:

http://www.efunda.com/math/num_integration/findgausslaguerre.cfm
provides Gaussian nodes and weights as a function of *n*

Abscissas and Weights of Gauss-Legendre Integration

[2] **points Gauss-Legendre Integration** Go

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{b+a}{2}\right)d\xi \approx \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k) f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right)$$

No. k	Abscissas ξ_k	Weight $w(\xi_k)$
1	-0.57735026919	1
2	0.57735026919	1

Abscissas and Weights of Gauss-Legendre Integration

10 points Gauss-Legendre Integration Go

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}\xi + \frac{b+a}{2}\right)d\xi \approx \frac{b-a}{2}\sum_{k=1}^{n} w(\xi_k) f\left(\frac{b-a}{2}\xi_k + \frac{b+a}{2}\right)$$

No. k	Abscissas ξ_k	Weight $w(\xi_k)$
1	-0.973906528517	0.0666713443087
2	-0.865063366689	0.149451349151
3	-0.679409568299	0.219086362516
4	-0.433395394129	0.26926671931
5	-0.148874338982	0.295524224715
6	0.148874338982	0.295524224715
7	0.433395394129	0.26926671931
8	0.679409568299	0.219086362516
9	0.865063366689	0.149451349151
10	0.973906528517	0.0666713443087

Gaussian Quadrature:

$$\int_{-1}^{1} f(y)\, dy \;\approx\; f\left(\frac{-\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

Where do weights of $c_1 = 1$; $c_2 = 1$ and nodes of $y_1 = \left(\frac{-\sqrt{3}}{3}\right)$; $y_2 = \left(\frac{\sqrt{3}}{3}\right)$ come from ?

Gaussian Quadrature:

Four unknowns: c_1 , c_2 , y_1 , y_2

Assume: $\int_{-1}^{1} f(y)\, dy \;\approx\; \mathrm{c}_1 f(y_1) + c_2 f(y_2)$ is exact for polynomials up to order 3

Then: $$\int_{-1}^{1} f(y)\, dy = c_1\,(a_0 + a_1 y_1 + a_2 y_1^2 + a_3 y_1^3) + c_2\,(a_0 + a_1 y_2 + a_2 y_2^2 + a_3 y_2^3)$$

Or: $$\int_{-1}^{1} a_0\; dy = 2a_0 = (c_1 + c_2)a_0 \qquad \int_{-1}^{1} a_1\; y\; dy = 0 = (c_1 y_1 + c_2 y_2)a_1$$

$$\int_{-1}^{1} a_2\; y^2\; dy = \frac{2}{3} a_2 = (c_1 {y_1}^2 + c_2 {y_2}^2)a_2 \qquad \int_{-1}^{1} a_3\; y^3\; dy = 0 = (c_1 {y_1}^3 + c_2 {y_2}^3)a_3$$

Gaussian Quadrature:

Four unknowns: c_1 , c_2 , y_1 , y_2

Assume: $\int_{-1}^{1} f(y)\, dy \;\approx\; c_1 f(y_1) + c_2 f(y_2)$ is exact for polynomials up to order 3

Then: $$\int_{-1}^{1} f(y)\, dy = c_1\,(a_0 + a_1 y_1 + a_2 y_1^2 + a_3 y_1^3) + c_2\,(a_0 + a_1 y_2 + a_2 y_2^2 + a_3 y_2^3)$$

Or: $$\int_{-1}^{1} a_0\; dy = 2 \quad = (c_1 + c_2)$$

$$\int_{-1}^{1} a_1\; y\; dy = 0 = (c_1 y_1 + c_2 y_2)$$

$$\int_{-1}^{1} a_2\; y^2\; dy = \frac{2}{3} \quad = (c_1 y_1{}^2 + c_2 y_2{}^2)$$

$$\int_{-1}^{1} a_3\; y^3\; dy = 0 = (c_1 y_1{}^3 + c_2 y_2{}^3)$$

Four equations, four unknowns: c_1 , c_2 , y_1 , y_2

Gaussian Quadrature:

Four unknowns: c_1 , c_2 , y_1 , y_2

Assume: $\int_{-1}^{1} f(y)\,dy \;\approx\; c_1 f(y_1) + c_2 f(y_2)$ is exact for polynomials up to order 3

Then: $$\int_{-1}^{1} f(y)\,dy = c_1\,(a_0 + a_1 y_1 + a_2 y_1^2 + a_3 y_1^3) + c_2\,(a_0 + a_1 y_2 + a_2 y_2^2 + a_3 y_2^3)$$

Or: $$\int_{-1}^{1} a_0\;dy = 2 \quad = (c_1 + c_2) \qquad \int_{-1}^{1} a_1\;y\;dy = 0 = (c_1 y_1 + c_2 y_2)$$

$$\int_{-1}^{1} a_2\;y^2\;dy = \frac{2}{3} \quad = (c_1 {y_1}^2 + c_2 {y_2}^2) \qquad \int_{-1}^{1} a_3\;y^3\;dy = 0 = (c_1 {y_1}^3 + c_2 {y_2}^3)$$

Solution: $c_1 = 1,\; c_2 = 1\;,\; y_1 = \frac{-\sqrt{3}}{3}\;,\; y_2 = \frac{\sqrt{3}}{3}$

Numerical Integration

Gaussian Quadrature: we would like to fill in the following table:

Function	2 nodes	10 nodes	Error, 10 nodes
$3\ x^2$			
$5\ x^4$			
$9\ x^8$			
$13\ x^{12}$			

a) Assignment #6, part 1: Start with ~rjs/aosc/week_06/gauss_integrate.f

b) A few lines in subroutines fn1, fn2, fn3, and fn4 need to be completed

These lines relate variables ***x*** and ***y***, as well as d***x***/d***y***, where ***x*** is the dependent variable used for our functions so far today (limits of 0 to 1) & ***y*** is the dependent variable for Gaussian integration (limits of −1 to 1)

Once these lines are filled in, the cells in the table can be populated

Calculation of multi-dimensional integrals adds additional level of complexity ☺

IDL and Matlab have various tools for the evaluation of integrals of data:

- **good for you, the user, to understand how these tools work !**
- **pay attention to:**
  - **grid spacing:** **are grid pts evenly spaced?**
    should they be evenly spaced?
    should they be "linear"?
    how does integrand vary relative to grid points?
  - **number of points:**
    some methods place strict limits on "mod N"
    may need to modify IDL or MATLAB integration tool code to properly handle end points