CS370: Interpolation

Graham Cooper

January 11th, 2017

See figure 2.1

$$y = p(x)$$

We want to find a function p, such that the curve is 'nice' (where nice is piecewise polynomial or polynomial)

Given:
$(x_1, y_1), (x_2, y_2)..., (x_n, y_n)$ n points $x_1 < x_2 < ... < x_n$

Find a polynomial P(x) of degree $< n$
In general:

$$p(x) = c_1 + c_2x + c_3x^2 + ... + c_nx^{n-1}$$
$$p(x_1) = y_1$$
$$p(x_2) = y_2$$
$$...$$
$$p(x_n) = y_n$$

n unknowns, n equations (linear)

Example:
$(-1, 1), (1, 1), (2, 5), (4, 1)$
See figure 2.2

$$p(x) = c_1 + c_2x + c_3x^2 + c_4x^3$$
$$p(-1) = c_1 - c_2 + c_3 - c_4 = 1$$
$$p(1) = c_1 + c_2 + c_3 + c_4 = 1$$
$$p(2) = c_1 + 2c_2 + 4c_3 + 8c_4 = 5$$
$$p(4) = c_1 + 4c_2 + 16c_3 + 64c_4 = 1$$

$$\left\{\begin{matrix} 1 & -1 & 1 & -1 & | & 1 \\ 1 & 1 & 1 & 1 & | & 1 \\ 1 & 2 & 4 & 8 & | & 5 \\ 1 & 4 & 16 & 64 & | & 1 \end{matrix}\right\} \text{// Solve the matrix!!}$$

Now we are just writing out the solution...

$$\begin{aligned} p(x) &= c_1 + c_2x + c_3x^2 + c_4x^3 \\ &= 1 + b_2(x-1) + b_3(x-1)^2 + b_4(x-1)^3 \\ &= L_1(x) + L_2(x) + 5L_3(x) + L_4(x) \end{aligned}$$

$$\begin{aligned} L_1(x) &= \frac{(x-1)(x-2)(x-4)}{-30} \\ L_2(x) &= \frac{(x+1)(x-2)(x-4)}{6} \\ L_3(x) &= \frac{(x+1)(x-1)(x-4)}{-6} \\ L_4(x) &= \frac{(x+1)(x-1)(x-2)}{30} \end{aligned}$$

I think we are writing it out this way so that we can easily plug in the vlues and get the correct points??

Question:

1. Does an interpolating polynomial always exist?
2. If (1) is true then is the answer always unique?

$$\begin{aligned} p(x) &= c_1 + c_2x + ...c_nx^{n-1} \\ p(x_1) &= c_1 + c_2x_1 + ...c_nx_1^{n-1} \\ p(x_2) &= c_1 + c_2x_2 + ...c_nx_2^{n-1} \\ &... \\ p(x_n) &= c_1 + c_2x_n + ...c_nx_n^{n-1} \end{aligned}$$

$$\left\{\begin{matrix} 1 & x_1 & x_1^2 & ... & x_1^{n-1} \\ 1 & x_2 & x_2^2 & ... & x_2^{n-1} \\ ... & ... & ... & ... & ... \\ 1 & x_n & x_n^2 & ... & x_n^{n-1} \end{matrix}\right\} \left\{\begin{matrix} c_1 \\ c_2 \\ ... \\ c_n \end{matrix}\right\} = \left\{\begin{matrix} y_1 \\ y_2 \\ ... \\ y_n \end{matrix}\right\}$$

The first matrix is the vandermonde (V) matrix.
V is invertable, $V \times \overrightarrow{c} = \overrightarrow{y}$
det $V \neq 0$ and det $V = \pi(x_i - x_j) \neq 0$ for $i < j$

Remember what a determinate is, remember what invertible is, but we will never be asked to do it.

$p(x)$

$$p(x) = q_1(x)(x - x_1) + y_1$$

$$p(x) = q_2(x)(x - x_2) + y_2$$

$$...$$

$$p(x) = q_n(x)(x - x_n) + y_n$$

Lagrange Polynomial

$$(x_1, y_1), (x_2, y_2)...(x_n, y_n)$$

$$p(x) = y_1 L_1(x) + y_2 L_2(x) + ... + y_n L_n(x)$$

$L_i(x_i) = 1, L_i(x_j) = 0$ for $i \neq j$ and $deg(L_i) = n - 1$
Lets construct L_1 using the above

$$L_1(x) = \frac{(x - x_2)(x - x_3)...(x - x_n)}{(x_1 - x_2)(x_1 - x_3)...(x_1 - x_n)}$$

$$L_i(x) = \frac{(x - x_1)...(x - x_{i-1})(x - x_{i+1})...(x - x_n)}{(x_i - x_1)...(x_i - x_{i-1})...(x_i - x_n)}$$

$L_i(x_i) = 1$ and $L_j(x_j) = 0$ where $j \neq i$

For A1 Q3 (January 13th) - figuring out the solution to the recurrence - and using the answer to help

$$??\boxed{I_n} \leftarrow I_{n-1} \leftarrow I_{n-2} \leftarrow ... \leftarrow I_0$$

$$\checkmark \boxed{\hat{I_n}} \leftarrow \hat{I_{n-1}} \leftarrow ... \leftarrow \hat{I_1} \leftarrow \hat{I_0}$$

$$e_n \leftarrow e_{n-1} \leftarrow ... \leftarrow e_1 \leftarrow e_0$$

$$e_n = (-\alpha)^n e_0$$

$$I_n? = formula(I_0) =$$

Using p?

$$??\boxed{p_n} \leftarrow p_{n-1}p_{n-2}, p_{n-2}p_{n-3}, ..., p_1, p_0$$

$p_n = as^n + bt^n$ and a, b depend on p_0, p_1

$$\checkmark \boxed{\hat{p_n}} \leftarrow \hat{p_{n-1}}\hat{p_{n-2}}..., \hat{p_1}\hat{p_0}$$

This line but with hats (I got lazy) $p_n = as^n + bt^n$ and a, b depend on p_0, p_1
solve for e_n

Recall from Jan 11th: (regoing over the start of this page)

Lagrange Form (again)

For $x_1, x_2, ...x_n$ distinct, construct $L_1(x), L_2(x)...L_n(x)$
Satisfying:

1. $L_i(x)$ has degree n-1
2. $L_i(x_i) = 1$
3. $L_i(x_j) = 0$ if $i \neq j$

How do we construct this:

$$L_1(x) = \frac{(x - x_2)(x - x_3)...(x - x_n)}{(x_1 - x_2)(x_1 - x_3)...(x_1 - x_n)}$$

We divide like this in order to get an equation that satisfies that if we plug in x_1 we will end up getting 1 as required, otherwise we will be getting a 0. This is actually pretty cool. Neat!

$$L_i(x) = \frac{(x-x_1)...(x-x_{i-i})(x-x_{i+1})...(x-x_n)}{(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$

$$p(x) = y_1L_1(x) + y_2L_2(x) + ... + y_nL_n(x)$$

$$p(x_1) = y_11 + y_20 + ... + y_n0 = y_1$$

$$...$$

$$p(x_n) = y_10 + y_20 + ... + y_n1 = y_n$$

A question that he often has asked on midterms (and is almost 100% going to add it to ours):
Given: x_1, x_2, x_3x_4 as $-1, 1, 2, 117, 412$
Form $p(x) = L_1(x) + L_2(x) + L_3(x) + L_4(x)$
Write $p(x) = c_1 + c_2x + c_3x^2 + c_4x^3$

Draw the graph!
Solve for the 4 numbers, and find what is y at each of the 4 points?
Then we find out that f(x) = 1 for each
Therefore the solution is p(x) = 1

Cubic Hermite Interpolation

Another type of interpolation

Given: (x_L, y_L) more on the left side and (x_R, y_R) on the right side, S_L slope of the left side, and S_R the slope of the right side

$p(x)$ has degree at most 3 since we have 4 uknowns
$p(x_L) = y_L$, $p(x_R) = y_R$, $p'(x_L)S_L$, $p'(x_R) = S_R$

$p(x) = c_1 + c_2(x - x_L) + c_3(x - x_L)^2 + c_4(x - x_L)^3$ $p'(x) = c_2 + 2c_3(x - x_L) + 3c_4(x - x_L)^2$ $p(x_L) = y_L \implies c_1$ $p'(x_L) = S_L \implies c_2$ $p(x_R) = y_R \implies c_1 + c_2\Delta x + c_3\Delta x^2 + c_4\Delta x^3 = y_R$ $p'(x_R) = S_R \implies c_2 + 2c_3\Delta x + 3c_4\Delta x^2 = S_R$ where $\Delta x = x_R - x_L$

$$\left\{\begin{matrix} 1 & 0 & 0 & 0 & | & Y_L \\ 0 & 1 & 0 & 0 & | & S_L \\ 1 & \Delta x & \Delta x^2 & \Delta x^3 & | & Y_R \\ 0 & 1 & 2\Delta x & 3\Delta x^2 & | & S_L \end{matrix}\right\}$$

becomes

$$\left\{\begin{matrix} 1 & 0 & 0 & 0 & | & Y_L \\ 0 & 1 & 0 & 0 & | & S_L \\ 0 & 0 & 1 & 0 & | & \frac{3Y_R'-2S_L-S_R}{\Delta x} \\ 0 & 0 & 0 & 1 & | & \frac{S_R+S_L-2y_L'}{\Delta x^2} \end{matrix}\right\}$$

$$c_1 = y_L$$

$$c_2 = S_L$$

$$c_3 = \frac{3Y_R' - 2S_L - S_R}{\Delta x}$$

$$c_4 = \frac{S_R + S_L - 2y_L'}{\Delta x^2}$$

Sub into p(x)

$$p(x) = 3 - (x-1) + 3(x-1)^2 - (x-1)^3$$

From Jan 16th:
See image Interp1.1: He is showing that the polynomial (red line) could be bad, we want the green line instead.

Cubic Spline

Given: $(x_1, y_1), ..., (x_N, y_N)$ N points
$x_1 < x_2 < ... < x_{N-1} < x_N$

A cubic spline is a function S(x) defined on the interval $[x_1, x_N]$ which satisfies the following:
(see interp1.2 figure)

1. In each interval $[x_i, x_{i+1}]$ S(x) is a cubic polynomial. $S_i(x) = a_i + b_i(x - x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3$

2. S(x) interpolates the N points: $S(x_i) = y_i$

3. S'(x) is continuous

4. S"(x) is continuous

5. 2 other things??

Is this well defined?
How man unknowns? 4 per interval, N-1 intervals $\rightarrow 4N-4$ unknowns
How many conditions (equations)?
Condition(2) $\rightarrow$ 2 equations per interval $\rightarrow 2N-2$
$S_i(x_i) = y_i$ and $S_i(x_{i+1}) = y_{i+1}$
Condition(3) - 1 equation per interior point $\rightarrow N-2$
Condition(4) - 1 equation per interiour point $\rightarrow N-2$
In total we get $4N-6$ equations

Boundary Conditions

1. Natural cubic spline $S''(x_1) = 0$, $S''(x_N) = 0$

2. Clamped cubic spline $S'(x_1) = s_1$ and $S'(x_N) = s_N$ s_1, s_N are known

3. Periodic Cubic spline $S'(x_N) = S'(x_1)$ and $S''(x_N) = S''(x_1)$

4. Not-a-knot condition (Matlab default) $S'''(x)$ is continuous at x_2 and x_{N-1}

How do we compute a cubic spline?

Method 1:
Have $4N-4$ uknowns and $4N-4$ linear equations $\rightarrow$ sove via Gaussian elimination
This is a cost of: $O((4N-4)^3) = O(N^3)$

Method 2:
Think of the deriviatives $S_1, S_2, ...S_N$ as the unknowns. We will set up linear equations for these derivatives
Then:

1. This will give us $S_1(x), S_2(x), ..., S_{N-1}(x)$

2. We will solve linear system in O(N) operations

Given: (-2,1), (0,0), (1,3), (4,-1), (5,2)
Clamped
(Figure Interp1.3)

S(x) =

$$a_1 + b_1(x+2) + c_!(x+2)^2 + d_1(x+2)^3\_for\_-2 \leq x \leq 0$$
$$a_2 + b_2x + c_2x^2 + d_2x^3\_for\_0 \leq x \leq 1$$
$$a_3 + b_3(x-1) + c_3(x-1)^2 + d_3(x-1)^3\_for\_1 \leq x \leq 4$$
$$a_4 + b_4(x-4) + c_4(x-4)^2 + d_4(x-4)^3\_for\_4 \leq x \leq 5$$

16 uknowns and 16 equations
$a_1 = 1$, $a_2 = 0$, $a_3 = 3$, $a_4 = -1$
$a_1 + 2b_1 + 4c_1 + 8d_1 = 0$
$a_2 + b_2 + c_2 + d_2 = 3$
✓
✓
$b_1 + 4c_1 + 12d_1 = b_2$
$2c_1 + 12d_1 = 2c_2$
✓
✓
✓
✓
$b_1 = 1$
$b_4 = 2c_4 + 3d_4 = 0$

Go and do the assignment question that looks like this

Generic example:
Given: $(x_1, y_1), (x_2, y_2) ..., (x_N, y_N)$
let $s_1, s_2, s_3 ... s_N$ denote the derivitive values of the spline S(x) at the points.
These are uknowns, but they exist.
Figure interp1.4
figure interp1.5 is a blown up of X_i

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$a_i = y_i$
$b_i = s_i$
$c_i = \frac{3y_i' - 2s_i - s_{i+1}}{\Delta x_i}$
$d_i = \frac{s_i + s_{i+1} - 2y_i'}{\Delta x_i^2}$
$\Delta x_i = x_{i+1} - x_i$
$Y_i' = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$

Additional information:

$S''(x_1) = 0, S_1''(x_2) = S_2''(x_2), S_2''(x_3) = S_3''(x_3), ...S_{N-2}''(x_{N-1}) = S_{N-1}''(X_{N-1}), S_{N-1}''(x_N) = 0$

Equation 1:
$2c_1 = 0$ ie. $\frac{3y_1' - 2s_1 - s_2}{\Delta x_1} = 0$
$2s_1 + s_2 = 3y_1'$

We now want to set up a linear set of equations for the esses

$$\begin{Bmatrix} * & * \\ ... & \\ ... & \\ ... & \\ ... & \\ * & * \end{Bmatrix} \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ ... \\ s_{n-1} \\ s_N \end{Bmatrix} = \begin{Bmatrix} * \\ ... \\ ... \\ ... \\ ... \\ * \end{Bmatrix}$$

$$c_{N-1} + 3d_{N-1}(x_N - x_{N-1}) = 0$$

$$3Y_{N-1}' - 2SN - 1 - S_N) + 3(S_{N-1} + S_N - 2Y_{N-1}') = 0$$

He deleted things on the bottom :(((

$$S_{N-1} + 2S_N = 3Y_{N-1}'$$

$$(x_i) = S_i''(x_i)$$

For $i = 2, 3, ...N - 1$

$$= 2c_i$$

$$= \frac{2(3Y_i' - 2s_i - s_{i+1})}{\Delta x_i}$$

$S''_{i-1}(x) = 2c_{i-1} + 6d_{i-1}(x - x_{i-1})$
$S_{i-1}(x) = a_{i-1} + b_{i-1}(x - x_i) + c_{i-1}(x0x_{i-1})^2 + d_{i-1}(x - x_{i-1})^3$

$c_{i-1} + 3d_{i-1}\Delta x_{i-1}$

$$\Delta x_i((3Y_{i-1}' - 2s_{i-1} - s_i) + 3(s_{i-1} + s_i - 2y_{i-1}')) = (3Y_i' - 2s_i - s_{i-1})\Delta x_{i-1}$$

$$\Delta x_i S_{i-1} + 2(\Delta x_i + \Delta x_{i-1})S_i$$

$$\Delta x_{i-1}S_{i+1} = 3y_{i-1}'\Delta x_i + 3y_i'\Delta x_{i-1}$$

The above is assignment question 5, but like wtf is going on...
Object is to find values for the S's in the matrix...

From Jan 20th

See slides.

From Jan 25th

Slide: Forward Euler Example #1:

a):

$$y_{n+1} = y_n + hf(t_r, y_n)$$

$$y_{n+1} = y_n + (1) \times 2y_n$$

$$y_{n+1} = 3y_n$$

First equation

n	t_n	y_n	$y(t_n)$
0	1	3	3
1	2	9	22
2	3	27	164
3	4	81	1210
4	5	243	8943

Second equation

n	t_n	y_n	$y(t_n)$
0	1	3	3
1	1.5	6	8
2	2	12	22
3	2.5	81	60
4	3	243	163
5	3.5	96	445
6	4	192	1210

Third Equation

n	t_n	y_n	$y(t_n)$
0	1	3	3
1	2	9	22
2	3	27	164
3	4	81	1210
4	5	243	8943

Example 2

a)

$y_{n+1} = y_n + hf(t_n, y_n)$

$$\begin{bmatrix} X_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} X_n \\ y_n \end{bmatrix} + 2 \times \begin{bmatrix} -y_n \\ x_n \end{bmatrix}$$

n	t_n	x_n	y_n
0	0	2	0
1	2	2	4
2	4	-6	8
3	6	-22	-4

$x_1 = x_0 - 2y_0 = 2 - 2(0) = 2$
$y_1 = y_0 + 2x_0 = 0 + 2(2) = 4$
$x_2 = x_1 - 2y_1 = 2 - 2(4) = -6$
$y_2 = y_1 + 2x_1 = 4 + 2(2) = 8$

Deriving Forward Euler

Method 1

Approximate y' with "finite differences":
0DE: $y'(t) = f(t, y(t))$
Approximate: $y'(t_n) \approx \frac{y_{n+1} - y_n}{h} = f(t_n, y_n)$

$\rightarrow y_{n+1} = y_n + hf(t_n, y_n)$

<u>Method 2</u>
Use a Taylor series to estimate $y(t_{n+1})$ and drop high order terms.

$$y(t_{n+1}) = y(t_n + h) \approx y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3)$$

Remove terms due to estimations

$$y(t_{n+1}) = y(t_n + h) \approx y(t_n) + hy'(t_n)$$

$$y_{n+1} = y_n + hf(t_n, y_n)$$

<u>Jan 27th</u>

Forward Euler Error

Slide Understanding Forward Euler:

FE is $y_{n+1} = y_n + hf(t_n, y_n)$
Taylor series is: $y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3)$
The error is the difference, assuming exact data, at t_n ie $y_n = y(t_n)$

So FE becomes:

$$y_{n+1} = y(t_n) + hf(t_n, y(t_n)) = y(t_n) + hy'(t_n)$$

The difference is:

$$\begin{aligned} y_{n+1} - y(t_n) &= \frac{-h^2}{2}y''(t_n) + O(h^3) \\ &= O(h^2) \end{aligned}$$

The local truncation error of FE is $O(h^2)$. Error decreases quadratically with h (the time step)

Deriving Trapezoidal Rule

Keeping more terms slide

Taylor series is: $y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + O(h^3)$
Replace the y" term with finit differences approximation:
$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}[\frac{y'(t_{n+1} - y'(t_n))}{g} + O(h)](t_n) + O(h^3)$
The muliploicatino by h^2 gets rid of the O(h)

Simplify:

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hy'(t_n) + \frac{h}{2}y'(t_{n+1}) - \frac{h}{2}y'(t_n) + O(h^3) \\ &= y(t_n) + \frac{h}{2}[f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))] + O(h^3) \end{aligned}$$

$\therefore$ Trapezoidal is $O(h^3)$:

$$y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$$

Deriving Modified Euler

Error of improved euler slide:
Trapezoidal: $y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$
ForwardEuler: $y(t_{n+1}) = y(t_n) + hy'(t_n) + O(h^2)$
Let $y*_{n+1} = y(t_n) + hf(t_n, y(t_n))$ so 1st step has error:
$y(t_{n+1}) - y*_n + 1 = O(h^2)$

Taylor expanding f, we get:

$$f(t_{n+1}, y(t_{n+1})) = f(t_{n+1}, y*_{n+1}) + \frac{2f}{2y}(t_{n+1}, y*_{n+1})(y(t_{n+1}) - y*_{n+1}) + O((y(t_{n+1}) - y*_{n+1})^2)$$

$$\therefore f(t_{n+1}, y(t_{n+1})) = f(t_{n+1}, y*_{n+1}) + O(h^2)$$

Plugging into trapezoidal:

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + \frac{h}{2}[f(t_n, y(t_n)) + f(t_{n+1}, y*_{n+1}) + O(h^2)] + O(h^3) \\ &= y(t_n) + \frac{h}{2}[f(t_n, y(t_n) + f(t_{n+1}, y*_{n+1})] + O(h^3) \end{aligned}$$

Improved Euler Example

Apply Improved Euler with h = 2 for 2 steps:
General Form: $y*_{n+1} = y_n + hf(t_n, y_n)$
$y_{n+1}y_n\frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y*_{n+1})]$
$IVR = X'(t) = -y(t)$ and $y'(t) = x(t)$
$\therefore$ we have

$$X^t_{n+1} = x_n + 2(-y_n)$$

$$y^t_{n+1} = y_n + 2(x_n)$$

$$x_{n+1} = x_n + \frac{2}{2}[-y_n + -y*_{n+1}]$$

$$y_{n+1} = y_n + \frac{2}{2}[X_n + X*_{n+1}]$$

Step through
Step1 (n = 0)

$$x_1* = X_0 - 2y_0 = 2 - 2(0) = 2$$

$$y_1* = y_0 + 2X_0 = 0 + 2(2) = 4$$

$$x_1 = x_0 - y_0 - y_1* = 2 - 0 - 4 = -2$$

$$y_1 = y_0 + x_0 + x_1* = 0 + 2 + 2 = 4$$

n	t_n	X_n	y_n	$X*_n$	$Y*_n$
0	0	2	0	X	X
1	2	-2	4	2	4
2	4	-6	-8	-10	0

Trapezoidal Example

Generic for: $y_{n+1} = y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$
For this problem with h = 2

$$X_{n+1} = X_n + \frac{2}{2}[-y_n - y_{n+1}]$$

$$y_{n+1} = y_n + \frac{2}{2}[x_n + x_{n+1}]$$

$$X_{n+1} + y_{n+1} = X_n + y_n$$
$$-X_{n+1} + y_{n+1} = X_n + y_n$$

OR
$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n - y_n \\ x_n + y_n \end{bmatrix}$
Slide form Deriving BDF1/BDF2

Fit lagrange polnomial to (t_n, y_n) and (t_{n+1}, y_{n+1})

$$p(t) = y_n(\frac{t - t_{n+1}}{t_n - t_{n+1}}) + y_{n+1}(\frac{t - t_n}{t_{n+1} - t_n})$$
$$p(t) = y_n(\frac{t - t_{n+1}}{-h}) + y_{n+1}(\frac{t - t_n}{h})$$
$$p'(t) = \frac{y_{n+1} - y_n}{h}$$

This gives the slope which rerequire to match f at the end of step time

$$p'(t_{n+1}) = \frac{y_{n+1} - y_n}{h} = f(t_{n+1}, y_{n+1})$$

Be is:

$$y_{n+1}y_n + hf(t_{n-1}, y_{n-1})$$