Boundary Value Problems for ODEs

Basic Problem:

$$\text{BVP}\quad \begin{cases} \vec{y}\,' = \vec{f}(x,\vec{y}) & \text{on} \quad a < x < b \\ \vec{g}\,(\vec{y}(a), \vec{y}(b)) = \vec{0} \end{cases}$$

$$\vec{y} \in \mathbb{R}^n \qquad \vec{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$$
$$\vec{g} : \mathbb{R}^{2n} \to \mathbb{R}^n$$

General types of boundary conditions

- $\vec{g}$ may be linear or nonlinear function of $\vec{y}(a)$, $\vec{y}(b)$

- seperated ... each scaler equation (b.c.) represented by $\vec{g} = 0$ involves $\vec{y}(a)$ or $\vec{y}(b)$ but not both.

A common class of BVPs is the Sturm-Liouville problem

$$\begin{cases} \dfrac{d}{dx}\left(p(x)\dfrac{dy}{dx}\right) + g(x)\,y(x) + \lambda\, r(x)\, y(x) = 0 \\ \alpha_1 y(a) + \beta_1 y'(a) = \alpha \\ \alpha_2 y(b) + \beta_2 y'(b) = \beta \end{cases} \qquad ' = \frac{d}{dx}$$

Note: ~~to match~~ general form:

define $\vec{y} = \begin{bmatrix} y \\ y' \end{bmatrix} \equiv \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ $\qquad \vec{g} \equiv \begin{bmatrix} \alpha_1 y_1(a) + \beta_1 y_2(a) - \alpha \\ \alpha_2 y_1(b) + \beta_2 y_2(b) - \beta \end{bmatrix}$

$$\vec{y}\,' = \begin{bmatrix} y' \\ y'' \end{bmatrix} = \begin{bmatrix} y_2 \\ \frac{1}{p}\left(-g y_1 - \lambda r y_1 - p' y_2\right) \end{bmatrix} \equiv f(x,\vec{y})$$

EXAMPLE $(p=1,\ q=1,\ r=0,\ \alpha_1=\alpha_2=1,\ \beta_1=\beta_2=0$
$a=0$
$\alpha=0)$

$$\begin{cases} \dfrac{d^2y}{dx^2}+y=0 \\ y(0)=0 \\ y(b)=\beta \end{cases}$$

Gen. sol. $y(x)=A\cos x+B\sin x$

b.c. $\Rightarrow$ $y(x)=\dfrac{\beta}{\sin b}\sin x$ if $\sin b\neq 0$.

$\Rightarrow$ if $b=\pm\pi, 2\pi, \ldots$
- No solution if $\beta\neq 0$
- Infinitely many sol. if $\beta=0$

if $b\neq\pm\pi, 2\pi, \ldots$
- unique solution

For now we'll focus on a particular form of the ODE given by

$$y''=f(x,y,y')$$

where f is a scalar function of x, y, y'

Note again if we define $\vec{y}=\begin{bmatrix} y \\ y' \end{bmatrix}\equiv\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

$$\vec{y}\,'=\begin{bmatrix} y' \\ y'' \end{bmatrix}=\begin{bmatrix} y_2 \\ f(x,y_1,y_2) \end{bmatrix}\equiv\vec{f}(x,\vec{y})$$

General types of b.c. for $y'' = f(x, y, y')$

linear:

$y(a) = \alpha$	Dirichlet b.c.	separated
$y'(a) = \alpha$	Neumann b.c.	separated
$y(a) + \alpha_1 y'(a) = \alpha_2$	Robin / Mixed b.c.	separated
$y(a) = y(b)$, $y'(a) = y'(b)$	Periodic	not separated

nonlinear

$$y^2(a) + \alpha_1 y(a) = \alpha_2$$

$$y(a) + \alpha_1 (y'(a))^2 = \alpha_2$$

we'll proceed by examining some methods for solving $y'' = f(x, y, y')$

- since we know several methods to solve IVPs (initial value problems) we'll first explore a method that uses those ideas (the shooting method) along with nonlinear solvers.

Note: let's use $u(x)$ instead of $y(x)$ ...

Shooting Method

Consider

BVP
$$\begin{cases} u'' = f(x,u,u') & \text{on } a<x<b \\ u(a) = \alpha \\ u(b) = \beta \end{cases}$$

The shooting method uses the idea ...

... trade in the condition $u(b)=\beta$ for a different condition applied at $x=a$.

In this example a good choice would be $u'(a) = s$, where s is an unknown — to be determined — "shooting" parameter.

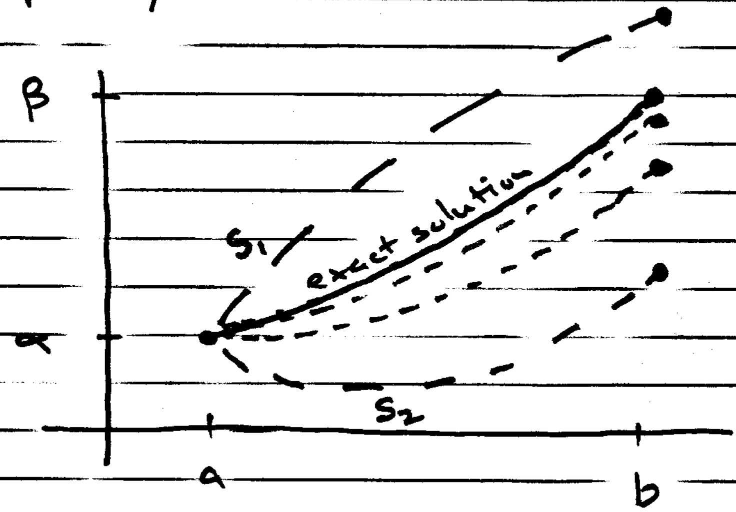
We seek a "good" value of s in which the initial value problem

$$\begin{cases} u'' = f(x,y,y') \\ u(a)=\alpha \\ u'(a)=s \end{cases}$$

happens to have the result $u(b) = \beta$.

We can think of guessing values of s until we find one that results in $u(b)=\beta$

Graphically ...



$u'(a) = s_1 \rightarrow u$ too high at b

$u'(a) = s_2 \rightarrow u$ too low at b

What are we really doing here?

- solving ODEs ... yes

BUT

* • $\boxed{\text{evaluating a function}}$ ← * This tells us when we have accomplished our goal.

what function?

$$z(s) = u_{IVP}(x=b; s) - \beta$$

Our basic goal is to find s such that

$$\underline{\underline{z(s) = 0}}.$$

This is a problem that we can characterize as solving a nonlinear function (i.e. finding a root of a nonlinear eq.)

- How does one evaluate this function?

  ... need to solve an initial value problem ✓

  Euler,
  Runge-Kutta
  ⋮

  ... so this potentially could be a costly function to evaluate!

- What numerical method should one use as the nonlinear solver?

  some options

  - Bisection — slow convergence but with a bracket convergence is [redacted] guaranteed.
    — only required to be able to evaluate the function. ✓

  - Newton's Method — fast, etc. ... but...

    we need $z'(s)$ ...

    (how do we get the derivative of the function?)

⋮

Shooting Method for Linear BVP (see also Burden + Faires)

Consider

$$\text{BVP} \begin{cases} u'' = p(x)u' + q(x)u + r(x) & a < x < b \\ u(a) = \alpha \\ u(b) = \beta \end{cases}$$

We can actually use the shooting method to solve this problem using two "shots".

First "shot":

replace $u(b) = \beta$ with $u'(a) = 0$

- call u_1 the solution of the IVP

$$\boxed{\text{IVP}_1 \begin{cases} u_1'' = p(x)u_1' + q(x)u_1 + r(x) \\ u_1(a) = \alpha \\ u_1'(a) = 0 \end{cases}}$$

- define $\tilde{u}_2 = u - u_1$ and note that $\tilde{u}_2$ satisfies

$$\begin{cases} \tilde{u}_2'' = p(x)\tilde{u}_2' + q(x)\tilde{u}_2 \\ \tilde{u}_2(a) = 0 \\ \tilde{u}_2(b) = \beta - u_1(b) \end{cases}$$

Note: if $u_1(b) = \beta$ we are already done — u_1 is the solution

- call u_2 the solution of the IVP

$$\boxed{\text{IVP}_2 \begin{cases} u_2'' = p(x)u_2' + q(x)u_2 \\ u_2(a) = 0 \\ u_2'(a) = 1 \end{cases}}$$

- Noting that u_2 (just computed) must also satisfy

$$\begin{cases} u_2'' = p(x)u_2' + q(x)u_2 \\ u_2(a) = 0 \\ u_2(b) = u_2(b) \end{cases}$$

← i.e. the value computed from IVP_2

By linearity + comparison with the equation for $\tilde{u}_2$ we can conclude

$$\tilde{u}_2(x) = \frac{\beta - u_1(b)}{u_2(b)} u_2(x)$$

(Note: if $u_2(b) = 0$ then $u_2(x) \equiv 0$ ... but this only happens if $\beta - u_1(b) = 0$ ... again by comparison to the $\tilde{u}_2$ eq. in which case we already have the solution). So proceeding assuming $u_2(b) \neq 0$, it follows that from $\tilde{u}_2 = u - u_1$ that

$$u(x) = u_1(x) + \tilde{u}_2(x)$$

$$\boxed{u(x) = u_1(x) + \frac{\beta - u_1(b)}{u_2(b)} u_2(x)}$$

Some questions to think about ..

- Do these ideas extend to higher order BVPs?
- What happens if you modify the boundary conditions?

Shooting Method for Nonlinear BVP

Consider

$$\text{BVP} \begin{cases} u'' = f(x,u,u') & \text{on } a<x<b \\ u(a) = \alpha \\ u(b) = \beta \end{cases}$$

Replace with

nonlinear solve

$$\begin{cases} \text{Find zero of } \; z(s) \equiv u(x=b;s) - \beta \\ \text{where } u(x=b;s) \text{ is evaluated by solving the IVP} \\ \text{IVP} \begin{cases} u'' = f(x,u,u') \\ u(a) = \alpha \\ u'(a) = s \end{cases} \end{cases}$$

Possible solvers...

Bisection: Find s_L and s_R such that $z(s_L)\cdot z(s_R)<0$ for a bracket (possibly just by guessing). Use this bracket to start the ~~[illegible]~~ bisection algorithm.
– iterate until convergence

Secant

$$s_{k+1} = s_k - z(s_k)\,\frac{(s_k - s_{k-1})}{z(s_k) - z(s_{k-1})} \qquad \cancel{k=2,3,\ldots}$$

- need two starting values of s.
  – iterate until convergence (hopefully!)

Newton's Method

Here we need

$$s_{k+1} = s_k - \frac{z(s_k)}{z'(s_k)} \qquad k = 1, 2, 3, \ldots$$

we need one initial guess for s and $z'(s)$
(or at least we need to be able to evaluate $z'(s_k)$).

Recall

$$z(s) \equiv u(x=b; s) - \beta$$

s_0

$$z'(s_k) = \frac{dz}{ds}(s_k) = \underbrace{\frac{\partial u}{\partial s}(x=b; s_k)}_{\text{can we find this?}}$$

Recall, the IVP we plan to solve is

$$\boxed{\text{IVP}_1 \quad \begin{cases} u''(x;s) = f(x, u(x;s), u'(x;s)) \\ u(a;s) = \alpha \\ u'(a;s) = s \end{cases}} \rightarrow z(s) = u(b;s) - \beta$$

Differentiate everywhere with respect to s

$$\begin{cases} \dfrac{\partial u''}{\partial s} = \dfrac{\partial f}{\partial u}\dfrac{\partial u}{\partial u} + \dfrac{\partial f}{\partial u'}\dfrac{\partial u'}{\partial s} \\[2ex] \dfrac{\partial u}{\partial s}(a;s) = 0 \\[2ex] \dfrac{\partial u'}{\partial s}(a;s) = 1 \end{cases}$$

Define $\quad w(x;s) \equiv \frac{\partial u}{\partial s}(x;s)$

so $\quad \frac{\partial(u'')}{\partial s} = \left(\frac{\partial u}{\partial s}\right)'' = w''(x;s)$

$\quad \frac{\partial(u')}{\partial s} = \left(\frac{\partial u}{\partial s}\right)' = w'(x;s)$

assume that the order of differentiation wrt x and s can be interchanged.

Then

$$\boxed{\mathrm{IVP}_2 \begin{cases} w'' = \frac{\partial f}{\partial u}(x,u,u')\,w + \frac{\partial f}{\partial u'}(x,u,u')\,w' \\ w(a,s) = 0 \\ w'(a,s) = 1 \end{cases}}$$

Note that

$$w(b;s) = \frac{\partial u}{\partial s}(b;s) = z'(s).$$

So we can obtain $z'(s)$ in a similar way to $z(s)$, but note that IVP_2 is coupled to IVP_1 (at least if f is nonlinear so that $\frac{\partial f}{\partial u}$ and $\frac{\partial f}{\partial u'}$ depend on u, u').

So for Newton's method we need to apply

$$\boxed{s_{k+1} = s_k - \frac{z(s_k)}{z'(s_k)} \qquad k = 1, 2, \ldots}$$

where

$$z(s_k) = u(b; s_k) - \beta$$

$$z'(s_k) = w(b; s_k)$$

z, z' are evaluated by solving the coupled IVPs

$$\begin{cases} \text{IVP}_1 & \begin{cases} u'' = f(x, u, u') \\ u(a; s) = \alpha \\ u'(a; s) = s \end{cases} \\ \text{IVP}_2 & \begin{cases} w'' = \frac{\partial f}{\partial u}(x, u, u')\, w + \frac{\partial f}{\partial u'}(x, u, u')\, w' \\ w(a; s) = 0 \\ w'(a; s) = 1 \end{cases} \end{cases}$$

So...

1. guess s_0
2. solve $\text{IVP}_1(s_0)$, $\text{IVP}_2(s_0)$ to get $z(s_0)$, $z'(s_0)$
3. $s_1 = s_0 - \dfrac{z(s_0)}{z'(s_0)} \qquad \left(= s_0 - \dfrac{u(b; s_0) - \beta}{w(b; s_0)} \right)$
4. iterate (go back to step 2 with new s_1)

compare to the linear case...

iterate until convergence (hopefully)

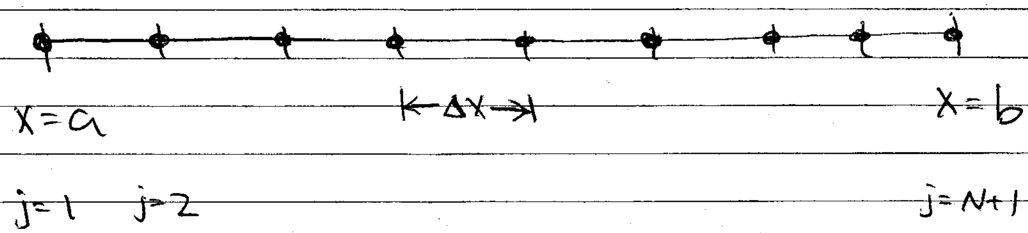
Finite Differences

EXAMPLE

Consider $\begin{cases} u'' + u = 0 \\ u(0) = 0 \\ u(1) = \beta \end{cases}$

How do we address this problem numerically?

- discretize the domain into N equal intervals



$a=0$
$b=1$
in the example

$$\Delta x = \frac{b-a}{N} \qquad x_j = a + \frac{(j-1)}{(N)}(b-a)$$

- seek a vector $\vec{u}$ in $\mathbb{R}^{N+1}$ that will approximate the continuous function $u(x)$ at grid points $x = x_j$

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{N+1} \end{bmatrix}$$

- identify $N+1$ equations to determine these $N+1$ unknowns.

$j=1$: $\quad u_1 = 0 \quad$ (enforces $u(0)=0$)

$j=2,\ldots,N$: $\quad \left(\frac{d^2u}{dx^2}\right)_j + u_j = 0 \quad$ (enforces ODE at interior points)

$j=N+1$ $\quad u_{N+1} = \beta \quad$ (enforces $u(1)=\beta$)

The key step is to ~~represent~~ approximate the derivatives (2nd deriv. in this case) of a continuous function (the exact sol.) using the discrete set of points $u_1, u_2, \ldots, u_{N+1}$.

One possible approximation for $\frac{d^2u}{dx^2}$ at $x = x_j$ is given by

$$\left(\frac{d^2u}{dx^2}\right)_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$$

[~~We'll see how to derive this and others shortly...~~]

- We are approximating the 2nd derivative at $x = x_j$ by a difference formula involving neighboring points u_{j-1}, u_j, u_{j+1}.

Note: to approximate the first derivative one ~~might refer back to the limit definition~~ of derivative

$$\frac{du}{dx} = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}$$

and take

$$\left(\frac{du}{dx}\right)_j = \frac{u_{j+1} - u_j}{h}$$

So in this example we find the unknowns $u_1, u_2, \ldots, u_{N+1}$ by solving

$(j=1)$ $$u_1 = 0$$

$j=2,\ldots,N$ $$\frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2} + u_j = 0$$

$(j=N+1)$ $$u_{N+1} = \beta$$

[Now here comes a critical connection between ODEs + linear algebra]

Writing this in matrix form and using the $j=1$ and $j=N+1$ information we have

$$\begin{bmatrix} -2+\Delta x^2 & 1 & & & 0 \\ 1 & -2+\Delta x^2 & 1 & & \\ & 1 & -2+\Delta x^2 & 1 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & & \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -\beta \end{bmatrix}$$

tri-diagonal system for unknowns $u_2, \ldots, u_N$!

Such a system can be set up and solved numerically (e.g. see Matlab code) for different values of N.

Compare $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}$ with exact sol.

$$\boxed{u(x) = \beta \frac{\sin x}{\sin 1}}$$

[see bvp_fd.m]

2.16

2.1 Finite Difference Formulas via Taylor Series

1D Taylor Expansion

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2!}h^2 f''(x) + \dots$$

$$+ \frac{h^n}{n!} f^{(n)}(x) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

for some c between x and $x+h$

2D Taylor Expansion (e.g. Thomas Calc., p. 1041)

$$f(x+h, y+k) = f(x,y) + \left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right) f\Big|_{(x,y)}$$

$$+ \frac{1}{2!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^2 f\Big|_{(x,y)}$$

$$+ \frac{1}{3!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^3 f\Big|_{(x,y)} + \dots$$

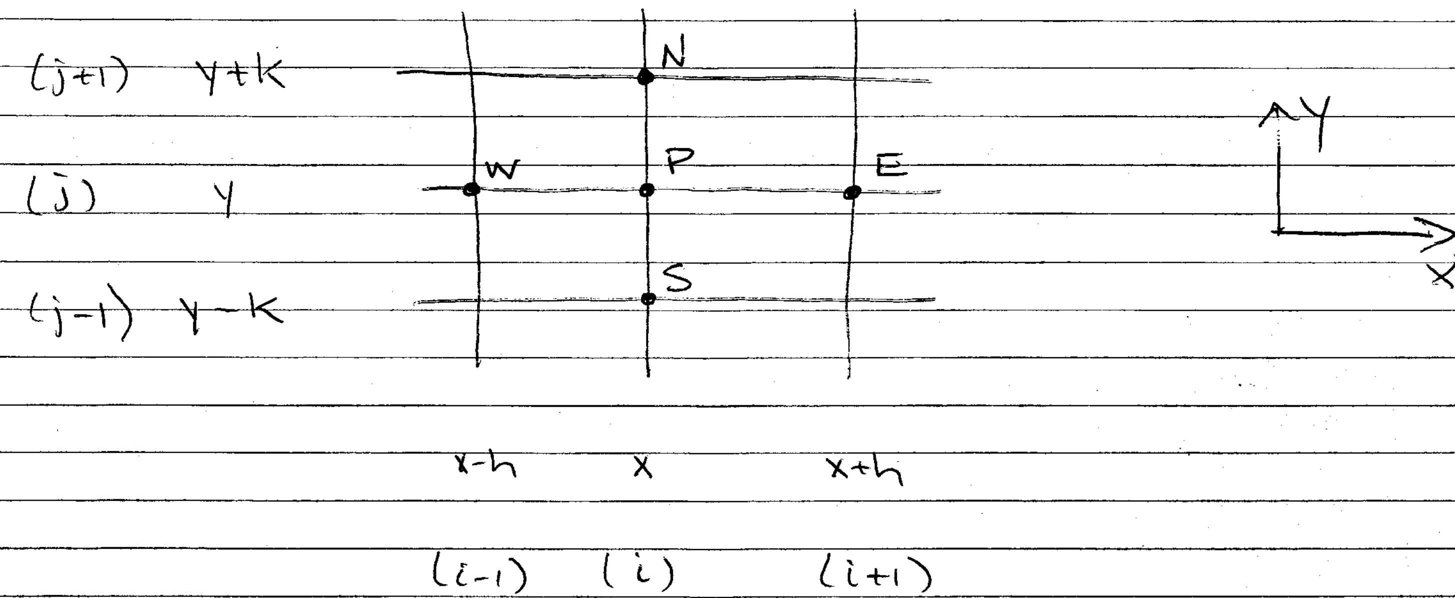
$$+ \frac{1}{n!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^n f\Big|_{(x,y)} +$$

$$\frac{1}{(n+1)!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^{n+1} f\Big|_{(x+ch, y+ck)}$$

[last term evaluated on line segment between (x,y) and $(x+h, y+k)$]

2.17

Consider the rectangular grid in 2D...

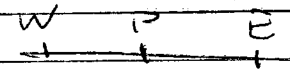


Identify

$$\phi(x,y) = \phi_P = \phi_{i,j}$$
$$\phi(x+h,y) = \phi_E = \phi_{i+1,j}$$
$$\phi(x-h,y) = \phi_W = \phi_{i-1,j}$$
$$\phi(x,y+k) = \phi_N = \phi_{i,j+1}$$
$$\phi(x,y-k) = \phi_S = \phi_{i,j-1}$$

We shall identify derivative formulas for $\frac{\partial\phi}{\partial x}, \frac{\partial^2\phi}{\partial x^2}, \frac{\partial^2\phi}{\partial x\partial y}, \ldots$ at point P (i,j) in terms of function values at neighboring points $\phi_E, \phi_W, \phi_N, \phi_S$; and ϕ_P.

1D Cases (focus only on East/West direction - x-dependence only)



$$\boxed{\frac{d\phi}{dx}}$$

Note

$$\phi_E = \phi(x+h) = \phi(x) + h\phi'(x) + \frac{1}{2!}h^2\phi''(x) + \frac{1}{3!}h^3\phi'''(x) + \frac{1}{4!}h^4\phi^{(iv)}(x) + \dots$$

and

$$\phi_W = \phi(x-h) = \phi(x) - h\phi'(x) + \frac{1}{2!}h^2\phi''(x) - \frac{1}{3!}h^3\phi'''(x) + \frac{1}{4!}h^4\phi^{(iv)}(x) + \dots$$

subtracting those

$$\phi(x+h) - \phi(x-h) = 2h\phi'(x) + \frac{2}{3!}h^3\phi'''(x) + O(h^5)$$

solving for $\phi'(x)$

$$\phi'(x) = \frac{\phi(x+h) - \phi(x-h)}{2h} \underbrace{- \frac{h^2}{3!}\phi'''(x) + \dots}_{\text{error} \sim h^2 \text{ [second order]}}$$

The formula

$$\boxed{\left(\frac{d\phi}{dx}\right)_P = \frac{\phi_E - \phi_W}{2h}}$$

is called a second order accurate central difference formula for $\frac{d\phi}{dx}$ at $x = x_j$

Note:

$$\left(\frac{d\phi}{dx}\right)_P = \frac{\phi_E - \phi_P}{h} \qquad \overset{\text{Error}}{\left(-\frac{1}{2}h\phi''(x)\right)}$$ [see Exercise 2.1]

and

$$\left(\frac{d\phi}{dx}\right)_P = \frac{\phi_P - \phi_W}{h} \qquad \left(+\frac{1}{2}h\phi''(x)\right)$$

are both first order accurate one-sided difference formulas for $\frac{d\phi}{dx}$

$$\boxed{\frac{d^2\phi}{dx^2}}$$

Add formulas for ϕ_E and ϕ_W

$$\phi_E + \phi_W = 2\phi_P + \frac{2}{2!}h^2\phi''(x) + \frac{2}{4!}h^4\phi^{iv}(x) + \ldots$$

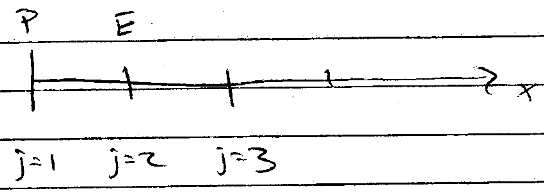
so $$\phi''(x) = \left(\frac{d^2\phi}{dx^2}\right)_P = \frac{\phi_E - 2\phi_P + \phi_W}{h^2} + O(h^2)$$

so $$\boxed{\left(\frac{d^2\phi}{dx^2}\right)_P = \frac{\phi_E - 2\phi_P + \phi_W}{h^2}}$$

is a second order accurate central difference formula for $\frac{d^2\phi}{dx^2}$ at $x = x_j$

$\boxed{\frac{d\phi}{dx} \text{ at boundaries}}$

left boundary



- to approximate $\frac{d\phi}{dx}$ at $x = x_1$ $(j=1)$ we could use the first order accurate

$$\left(\frac{d\phi}{dx}\right)_P = \frac{\phi_E - \phi_P}{h}$$

- to get a higher order accurate approximation we can use information at $j=3$

$$\begin{cases} \phi(x+h) = \phi(x) + h\phi'(x) + \frac{1}{2!}h^2\phi''(x) + \frac{1}{3!}h^3\phi'''(x) + \dots \\ \phi(x+2h) = \phi(x) + 2h\,\phi'(x) + \frac{1}{2!}4h^2\phi''(x) + \frac{8}{3!}h^3\phi'''(x) + \end{cases}$$

Combine these so that the $h^2\phi''$ terms drop out...

$$4\phi(x+h) - \phi(x+2h) = 3\phi(x) + 2h\phi'(x) + O(h^3)$$

$$\phi'(x) = \frac{-3\phi(x) + 4\phi(x+h) - \phi(x+2h)}{2h} + O(h^2)$$

So

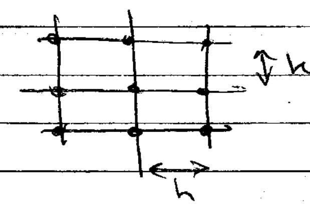
$$\boxed{\left(\frac{d\phi}{dx}\right)_{j=1} = \frac{-3\phi_1 + 4\phi_2 - \phi_3}{2h}}$$

2nd order accurate one-sided difference formula.

back to 2D

• $\boxed{\dfrac{\partial^2\phi}{\partial x\partial y}} = \dfrac{\partial}{\partial x}\left[\dfrac{\partial\phi}{\partial y}\right]$ expand first in y and then x (or vice verse)

$$\frac{\partial}{\partial x}\left[\frac{\phi_{i,j+1}-\phi_{i,j-1}}{2k}\right] + O(k^2)$$



$$= \frac{\left(\dfrac{\phi_{i+1,j+1}-\phi_{i+1,j-1}}{2k}\right) - \left(\dfrac{\phi_{i-1,j+1}-\phi_{i-1,j-1}}{2k}\right)}{2h} + O(h^2,k^2)$$

$$\boxed{\begin{aligned}\frac{\partial^2\phi}{\partial x\partial y} &= \frac{1}{4kh}\left[\phi_{i+1,j+1} - \phi_{i+1,j-1} - \phi_{i-1,j+1} + \phi_{i-1,j-1}\right] \\ &= \frac{1}{4kh}\left[\phi_{NE} - \phi_{SE} - \phi_{NW} + \phi_{SW}\right]\end{aligned}}$$

Error term: $-\dfrac{1}{hk}\left(\dfrac{hk^3}{3!}\dfrac{\partial^4\phi}{\partial x^3\partial y} + \dfrac{kh^3}{3!}\dfrac{\partial^4\phi}{\partial x\partial y^3}\right)$

by combining terms from Taylor Expansion.

(see Exercise 2.4)