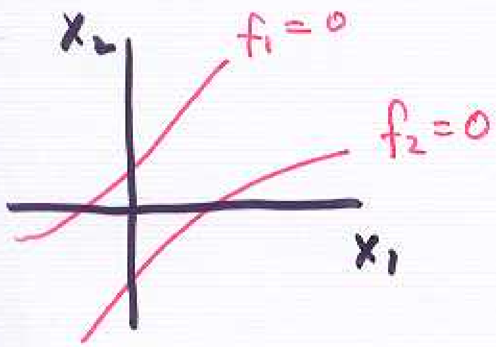
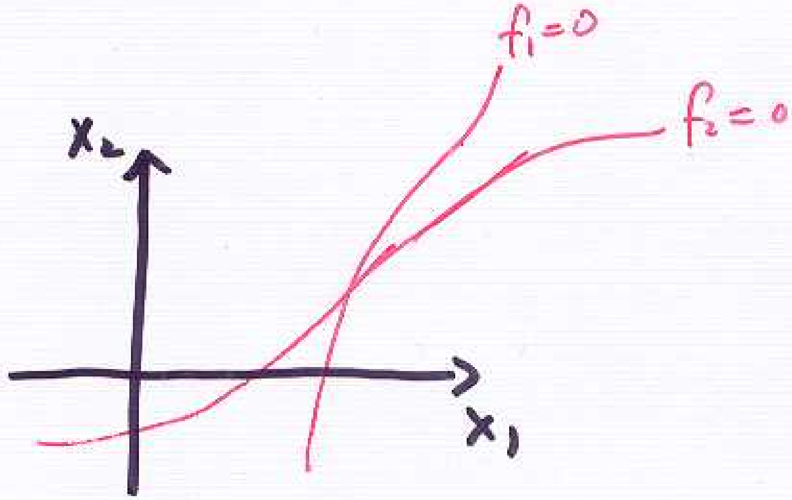
Systems of non-linear eqns.

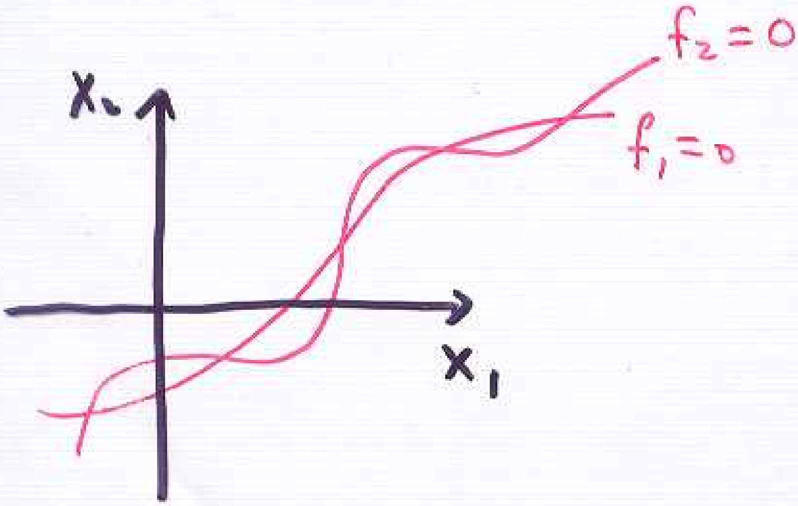
$$\left.\begin{array}{l} f_1(x_1, \ldots, x_n) = 0 \\ f_2(x_1, \ldots, x_n) = 0 \\ \quad\vdots \\ f_n(x_1, \ldots, x_n) = 0 \end{array}\right\} \rightarrow \underset{\sim}{F}(\underset{\sim}{x}) = \underset{\sim}{0} \quad \text{— (I)}$$

- Whether (I) has solns., and how many?

Example:







Picard iteration:

- Decompose $F = Ax + H(x)$

$$\underbrace{A}_{\text{linear}} \qquad \underbrace{H(x)}_{\text{nonlinear part}}$$

$x = A^{-1}F - A^{-1}H$

$F(x) = 0$

- Perform iteration

$$X_{i+1} = -A^{-1} \cdot H(x_i)$$

$\Rightarrow$ This is of the form $X_{i+1} = G(x_i)$

- Assume soln. X^* of $F(x^*) = 0 \Leftrightarrow X^* = G(x^*)$

- Suppose $\| G'(x) \| \leq \gamma < 1$ for $\| x - x^* \| < \beta$

where $G' \equiv \begin{bmatrix} \frac{\partial G_1}{\partial x_1} & \frac{\partial G_1}{\partial x_2} & \cdots \\ & \vdots & \\ & & \end{bmatrix}$,

iteration converges if $\| x_0 - x^* \| < \beta$

pf $\quad \dfrac{G(x_0) - G(x^*)}{x_0 - x^*} = G'(c) \quad$ c between x^* & x_1

$x_{i+1} = G(x_i) \qquad \| G(x_0) - G(x^*) \| = \|G'(c)\| \, \| x_0 - x^* \|$

$$\| x_1 - x^* \| = \| G'(c) \| \, \| x_0 - x^* \|$$

$$\| x_1 - x^* \| \leq \gamma \| x_0 - x^* \|$$

$$\| x_2 - x^* \| \leq \gamma \| x_1 - x^* \| < \gamma^2 \| x_0 - x^* \|$$

$$\Rightarrow \| x_n - x^* \| \leq \gamma^n \| x_0 - x^* \|$$

$\Rightarrow$ linear convergence

Fixed Point Iteration

$$\vec{x}^{(n+1)} = \vec{g}(\vec{x}^{(n)})$$ with fixed point $\vec{\alpha}$: $\vec{\alpha} = \vec{g}(\vec{\alpha})$

with slight modifications, contraction mapping theorem from 1-D generalizes directly to multi-dimensions.

$\Rightarrow$ The absolute values are replaced with norms.

Error:

$$\vec{\alpha} - \vec{x}^{(n+1)} = \vec{g}(\alpha) - \vec{g}(\vec{x}^{(n)})$$

$$= \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{pmatrix} (\vec{\xi}^{(n)}) \, (\vec{\alpha} - \vec{x}^{(n)})$$

$\vec{\xi}^{(n)}$ is on line joining $\vec{\alpha}$ & $\vec{x}^{(n)}$

$$\| \vec{\alpha} - \vec{x}^{(n+1)} \| \leq \| G(\vec{\xi}_n) \| \, \| \vec{\alpha} - \vec{x}^{(n)} \|$$

Jacobian G is analog to g' from 1-D case.

Convergence if $\|G\| < 1$.

For a domain to have contraction map.
the domain must be convex, all lines connecting two points in domain must lie entirely in domain.

Summary:

let $\vec{\alpha}$ be a fixed point of $\vec{g}(\vec{x})$. assume that $\vec{g}$ is cont. diff. in a neighborhood of $\vec{\alpha}$ & $\|G(\alpha)\| <$ [illegible]
For $\vec{x}_0$ sufficiently close to $\vec{\alpha}$, $\vec{x}^{(n+1)} = \vec{g}(\vec{x}^{(n)})$ will converge to $\vec{\alpha}$.

Converting root finding to fixed pt. iteration:

$$\vec{x} = \vec{x} + A \cdot \vec{f}(\vec{x}) \equiv \vec{g}(\vec{x})$$

A: $m \times m$ matrix (of constants, for example)

Jacobian $G = I + A \cdot \vec{F}(\vec{x})$, $\vec{F}$ is Jacobian of $\vec{f}$

- Recall that $\| G(\alpha) \| < 1$ for convergence
- Higher order of convergence if $\| G(\alpha) \| = 0$

$$\| G(\alpha) \| = \| I + A F(\vec{\alpha}) \| = 0$$

$$\therefore \quad A = - F^{-1}(\vec{\alpha}) \quad \text{or in the iteration}$$

$$A = - F^{-1}(\vec{x}^n)$$

$$\therefore \quad \vec{x}^{n+1} = \vec{x}^n - F^{-1}(\vec{x}^n) \cdot \vec{f}(\vec{x}^n)$$

$\Rightarrow$ Newton's method for systems

Numerically, $\vec{x}^{n+1} = \vec{x}^n + \vec{\delta}^{n+1}$

where

$$F(\vec{x}^n)\, \vec{\delta}^{n+1} = - \vec{f}(\vec{x}^n)$$

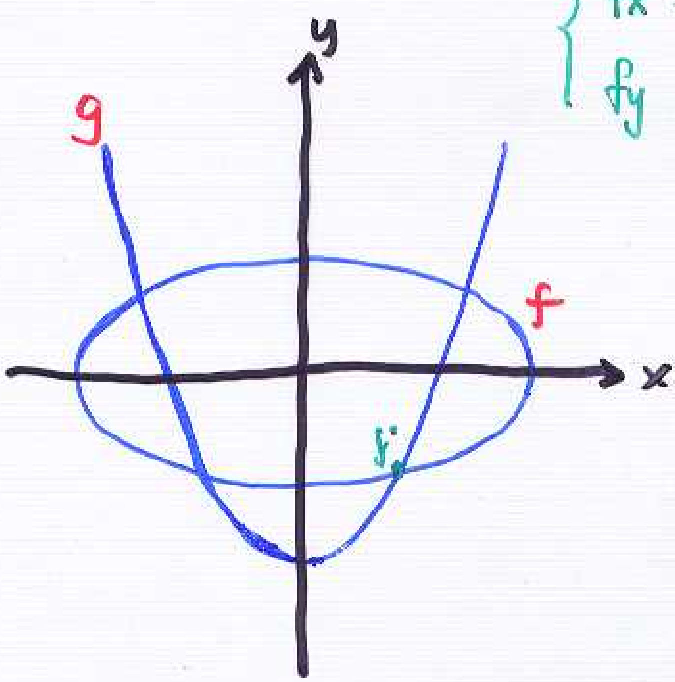
Use linear solver to find $\vec{\delta}^{n+1}$

- Recall that convergence results follow from fixed pt. thm.
- Computationally Expansive step for each iteration: soln. of linear system of equations $\sim \mathcal{O}(n^3)$
- A lot less iterations as we get closer to the fixed pt. $\Rightarrow$ Jacobian is almost const.

Example:

$$f(x,y) \equiv x^2 + 4y^2 - 9 = 0$$

$$g(x,y) \equiv 18y - 14x^2 + 45 = 0$$



$$\begin{cases} f_x \equiv \dfrac{\partial f}{\partial x} = 2x \\ f_y \equiv \dfrac{\partial f}{\partial y} = 8y \end{cases}$$

$$\begin{cases} g_x = -28x \\ g_y = 18 \end{cases}$$

Newton's method:

$$\begin{cases} x^{n+1} = x^n + \delta_x^{n+1} \\ y^{n+1} = y^n + \delta_y^{n+1} \end{cases}$$

$\delta = \text{rhs}/A$

$$\underbrace{\begin{bmatrix} f_x^n & f_y^n \\ g_x^n & g_y^n \end{bmatrix}}_{A} \begin{bmatrix} \delta_x^{n+1} \\ \delta_y^{n+1} \end{bmatrix} = - \underbrace{\begin{bmatrix} f^n \\ g^n \end{bmatrix}}_{\text{rhs}}$$

$$\begin{bmatrix} 2x^n & 8y^n \\ -28x^n & 18 \end{bmatrix} \begin{bmatrix} \delta_x^{n+1} \\ \delta_y^{n+1} \end{bmatrix} = - \begin{bmatrix} x^{n^2} + 4y^{n^2} - 9 \\ 18y^n - 14x^{n^2} + 45 \end{bmatrix}$$

$(x^0, y^0) = (1, -1)$

$$x^1 = 1 + \delta_x^1$$
$$y^1 = -1 + \delta_y^1$$

$$\begin{bmatrix} 2 & -8 \\ -28 & 18 \end{bmatrix} \begin{bmatrix} \delta_x^1 \\ \delta_y^1 \end{bmatrix} = - \begin{bmatrix} -4 \\ 49 \end{bmatrix}, \quad \begin{bmatrix} \delta_x^1 \\ \delta_y^1 \end{bmatrix} = \begin{bmatrix} 0.1702\cdots \\ -0.4574\cdots \end{bmatrix}$$

$(x^2, y^2) = (1.202158829, -1.376760321)$

$(x^3, y^3) = (1.203165807, -1.374083487)$

$(x^4, y^4) = (1.203166963, -1.374085342)$

$\vdots$

Example: $f(x) = x^2+1 = 0$

$x = a+bi$

$$f(a,b) = \underbrace{a^2-b^2+1}_{\text{Real}} + \underbrace{2abi}_{\text{Imag.}}$$

$$\Rightarrow \quad f(a,b) = Re\left[f(x=a+bi)\right] = a^2-b^2+1$$

$$g(a,b) = Im\left[f(x=a+bi)\right] = 2ab$$

$$f_a = 2a \qquad f_b = -2b$$

$$g_a = 2b \qquad g_b = 2a$$

$$a^{i+1} = a^i + \delta_a^{i+1}$$

$$b^{i+1} = b^i + \delta_b^{i+1}$$

$$\begin{bmatrix} f_a^i & f_b^i \\ g_a^i & g_b^i \end{bmatrix} \begin{bmatrix} \delta_a^{i+1} \\ \delta_b^{i+1} \end{bmatrix} = - \begin{bmatrix} a^{i2}-b^{i2}+1 \\ 2a^ib^i \end{bmatrix}$$

$$\begin{matrix} a^0 = 1 \\ b^0 = -\frac{1}{2} \end{matrix} \rightarrow \begin{cases} a^1 = 1 - \frac{9}{10} = \frac{1}{10} \\ b^1 = -\frac{1}{2} + \frac{1}{20} = -\frac{9}{20} \end{cases}$$

$$\rightarrow \begin{cases} a^2 = -0.09862 \\ b^2 = -0.437652 \end{cases}$$

$$\vdots$$

p. 9

Table 7.7. The Modified Newton's Method for Solving the Nonlinear System (7.65) with $n = 5$

k	$\|\alpha - x^{(k)}\|$	Ratio
0	3.58E − 1	
1	3.42E − 2	0.096
2	7.03E − 3	0.206
3	6.83E − 4	0.097
4	1.54E − 4	0.225
5	2.09E − 5	0.136
6	2.30E − 6	0.110
7	5.68E − 7	0.247
8	5.53E − 8	0.097
9	7.24E − 9	0.131

Newton's method (7.62) and the modified Newton's method (7.71)

$$x^{(k+1)} = x^{(k)} - A^{-1}F(x^{(k)})$$

are examples of fixed point iteration:

$$x^{(k+1)} = g(x^{(k)}), \qquad k \geq 0$$

where $g(x)$ is a vector function

$$g(x) = \begin{bmatrix} g_1(x_1, \ldots, x_n) \\ \vdots \\ g_n(x_1, \ldots, x_n) \end{bmatrix}$$

math.uiowa.edu/
ftp/atkinson/
ENA_Materials

There is a general theory for such fixed-point iteration methods involving a strengthened form of Theorem 3.4.2; but we do not consider it here [e.g., see Atkinson (1989, §2.10)].

MATLAB PROGRAM. We give a MATLAB program for the Newton method applied to the solution of a systems of two nonlinear equations in two unknowns:

$$F_1(x_1, x_2) = 0$$
$$F_2(x_1, x_2) = 0$$

```
function solution = newton_sys(x_init,err_tol,max_iterates)
%
% The calling sequence for newton_sys.m is
```

P.10


```
% solution = newton_sys(x_init,err_tol,max_iterates)
% This solves a pair of two nonlinear equations in two unknowns,
%              f(x) = 0
% with f(x) a column vector of length 2.  The definition of f(x)
% is to be given below in the function named fsys; and you
% also need to give the Jacobian matrix for f(x) in the
% function named deriv_fsys.
%
% x_init is a vector of length 2, and it is an initial guess
% at the solution.
%
% The parameters err_tol and max_iterates are upper limits on
% the desired error in the solution and the maximum number of
% iterates to be computed.

% Initialization.
x0 = zeros(2,1);
for i=1:2
   x0(i) = x_init(i);
end

error = inf;
it_count = 0;

% Begin the main loop.
while(error > err_tol & it_count < max_iterates)
   it_count = it_count + 1;
   rhs = fsys(x0);
   A = deriv_fsys(x0);
   delta = A\rhs;
   x1 = x0 - delta;
   error = norm(delta,inf);
   % The following statement is an internal print to show
   % the course of the iteration.  It and the pause
   % statement following it can be commented out.
   [it_count x1' error]
   pause
   x0 = x1;  <-- swapping
end

% Return with the solution.
solution = x1;
if it_count == max_iterates
   disp(' ')
   disp('*** Your answers may possibly not ...
```

gthened
§2.10)].

plied to

P.11


```
%                satisfy your error tolerance.')
end

%%%%%%%%%%%% Definition of functions %%%%%%%%%%%%%%%%%%%%
function f_val = fsys(x)
%
% The equations being solved are
%   x(1)^2 + 4*x(2)^2 - 9 = 0
%   18*x(2) - 14*x(1)^2 + 45 = 0

f_val = [x(1)^2+4*x(2)^2-9, 18*x(2)-14*x(1)^2+45]';

function df_val = deriv_fsys(x)
%
% This defines the Jacobian matrix for the function
% given in fsys

df_val = [2*x(1), 8*x(2); -28*x(1), 18];
```

PROBLEMS

1. Find the remaining three solutions of the system (7.52). Note that symmetry can be used to reduce the number of solutions that must be found computationally.

2. Find all solutions to the following systems, using Newton's method (7.55–7.56).
   (a) $x^2 + y^2 = 4, \quad x^2 - y^2 = 1$
   (b) $x^2 + 4y^2 = 4, \quad y = x^2 - 0.4x - 1.96$
   (c) $x^2 + y^2 = 1, \quad 2y = 2x^3 + x + 1$
   (d) $x + y - 2xy = 0, \quad x^2 + y^2 - 2x + 2y + 1 = 0$

3. Find all solutions to the system
$$x^2 + xy^3 = 9, \qquad 3x^2y - y^3 = 4$$
Use each of the initial guesses $(x_0, y_0) = (1.2, 2.5)$, $(-2, 2.5)$, $(-1.2, -2.5)$, $(2, -2.5)$. Observe the root to which each of the iterations converges and the number of iterates computed. Comment on your results.

4. Solve the nonlinear system (7.65) for $n = 10, 15, 20$, to the full accuracy of your computer. Compare the needed number of iterates. Graph the solutions x for each n, as in Figure 7.10.

5. Use the modified Newton's method (7.71) to solve for the roots of (7.52), presenting results as in Table 7.7. Use $A = F'(x^{(0)})$.

6. Repeat Problem 5 for the systems given in Problem 2.

Example:

consider an initial value problem:

$$\frac{\partial y}{\partial t} = f(x, y) \qquad y(0) = y_0$$

use backward Euler:

$$\frac{y_{k+1} - y_k}{\Delta t} = f(x_{k+1}, y_{k+1})$$

$$y_{k+1} = y_k + \Delta t \cdot f(x_{k+1}, y_{k+1})$$

Apply Newton method:

- y_k known
- need to solve for $y_{k+1} \equiv u$

$$F(u) \equiv u - \Delta t\, f(x_{k+1}, u) - y_k = 0$$

$$u_{i+1} = u_i - \frac{F(u_i)}{F'(u_i)}$$

$$= u_i - \frac{u_i - \Delta t \cdot f(x_{k+1}, u_i) - y_k}{1 - \Delta t\, f_u}$$

EXTRA

Notes

$$\|x - x^*\| < \beta$$

$$\|x - x^*\| = \sqrt{\sum_i (x_i - x_i^*)^2}$$

$$\vec{\alpha} - \vec{x}^{(n+1)} = \vec{g}(\alpha) - \vec{g}(\vec{x}^{(n)})$$

$$= \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{pmatrix} (\vec{\xi}_n) \cdot (\vec{\alpha} - \vec{x}^{(n)})$$

first comp.

$$\alpha_1 - x_1 = \frac{\partial g_1}{\partial x_1} (\alpha_1 - x_1^{(n)})$$

$$+ \frac{\partial g_1}{\partial x_2} (\alpha_2 - x_2^{(n)})$$

converting root finding prob

$\rightarrow$ fixed point problem.

$\Rightarrow$ root of $f(x)$

$$\Rightarrow \quad x = x + c \cdot f(x)$$

$$\bar{F} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ & \vdots & & \\ & & & \end{bmatrix}$$

$$\bar{F} = \frac{\partial f}{\partial x}$$

$$\bar{F}^{-1} = \frac{1}{\frac{\partial f}{\partial x}}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\bar{x}_{n+1} = \bar{x}_n - \bar{F}^{-1}(x_n) \cdot \bar{f}(\bar{x}_n)$$

$$\begin{bmatrix} 2x^n & 8y^n \\ -28x^n & 18 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = - \begin{bmatrix} f^n \\ g^n \end{bmatrix}$$

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = - \begin{bmatrix} & \\ & \end{bmatrix}^{-1} \begin{bmatrix} f^n \\ g^n \end{bmatrix}$$

$(x^0, y^0) = (1, -1)$

$$\begin{bmatrix} \delta_x^1 \\ \delta_y^1 \end{bmatrix} = - \begin{bmatrix} 2 & -8 \\ -28 & 18 \end{bmatrix}^{-1} \begin{bmatrix} f^0 \\ g^0 \end{bmatrix}$$