02.04 systems of equations: nonlinear systems

why stop at one? part four.

1 multivariate newtons method

heres with one:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

and heres a few:

$$\begin{aligned} f_1(u, v, w) &= 0 \\ f_2(u, v, w) &= 0 \\ f_3(u, v, w) &= 0. \end{aligned}$$

let vector-valued function $F(u, v, w) = (f_1, f_2, f_3)$ and problem be $F(x) = 0$ where $x = (u, v, w)$.

the analogue of derivative f' for one variable becomes the **jacobian matrix**

$$DF(x) = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial w} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial w} \\ \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial w} \end{bmatrix}$$

both one-variable and multivariate derive from linear approximation afforded by taylor expansion. for the latter,

$$F(x) = F(x_0) + DF(x_0) \cdot (x - x_0) + \mathcal{O}(x - x_0)^2$$

eg, linear expansion of $F(u, v) = (e^{u+v}, sinu)$ around $x_0 = (0, 0)$,

$$\begin{aligned} F(x) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} e^0 & e^0 \\ cos0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \mathcal{O}(x^2) \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} u + v \\ u \end{bmatrix} + \mathcal{O}(x^2) \end{aligned}$$

oc ignore $\mathcal{O}(x^2)$ like 1D and let $x = r$ be the root and x_0 be current guess. then

$$0 = F(r) \approx F(x_0) + DF(x_0) \cdot (r - x_0)$$

$$\Rightarrow -DF(x_0)^{-1} F(x_0) \approx r - x_0$$

algorithm

mutlivariate newtons method

$$\begin{aligned} x_0 &= \text{initial vector} \\ x_{k+1} &= x_k - (DF(x_k))^{-1} F(x_k), \quad k = 0, 1, 2, \ldots \end{aligned}$$

inverses are costly so avoid them.

$$\begin{aligned} DF(x_k)s &= -F(x_k) \\ x_{k+1} &= x_k + s \end{aligned}$$

example 32

apply newtons multivariate with $x_0 = (1, 2)$,

$$v - u^3 = 0$$
$$u^2 + v^2 = 1$$

$$f_1(u,v) = v - u^3$$
$$f_2(u,v) = u^2 + v^2 - 1$$

$$DF(u,v) = \begin{bmatrix} -3u^2 & 1 \\ 2u & 2v \end{bmatrix}.$$

with $x_0 = (1,2)$,

$$\underbrace{\begin{bmatrix} -3 & 1 \\ 2 & 4 \end{bmatrix}}_{DF(x_0)} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -\underbrace{\begin{bmatrix} 1 \\ 4 \end{bmatrix}}_{F(x_0)}$$

with $s = (0,-1) \Rightarrow x_1 = x_0 + s = (1,1)$,

$$\underbrace{\begin{bmatrix} -3 & 1 \\ 2 & 2 \end{bmatrix}}_{DF(x_1)} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -\underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{F(x_1)}$$

with $s = (-\frac{1}{8}, -\frac{3}{8}) \Rightarrow x_2 = x_1 + s = (\frac{7}{8}, \frac{5}{8})$.

further steps below.

step	u	v
0	1.00000000000000	2.00000000000000
1	1.00000000000000	1.00000000000000
2	0.87500000000000	0.62500000000000
3	0.82903634826712	0.56434911242604
4	0.82604010817065	0.56361977350284
5	0.82603135773241	0.56362416213163
6	0.82603135765419	0.56362416216126
7	0.82603135765419	0.56362416216126

note the quadratic convergence, hooray.

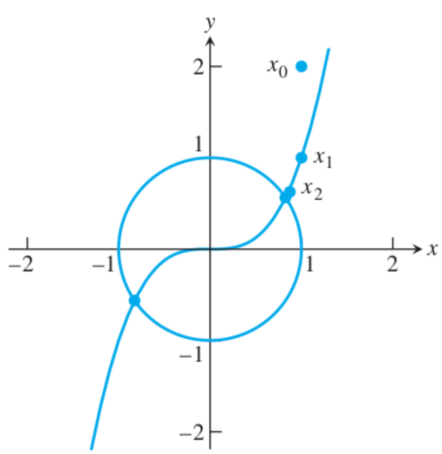

Figure 2.5 Newton's Method for Example 2.32. The two roots are the dots on the circle. Newton's Method produces the dots that are converging to the solution at approximately (0.8260, 0.5636).

2 broydens method

like how there is secant to newton-raphson 1D, here is broydens for when there are no derivatives.

suppose A_i is best approximation available at step i to the jacobian matrix, then

$$x_{i+1} = x_i - A_i^{-1} F(x_i).$$

to update A_i to A_{i+1}, respect the derivative aspect of jacobian DF,

$$A_{i+1} \delta_{i+1} = \Delta_{i+1},$$

where $\delta_{i+1} = x_{i+1} - x_i, \Delta_{i+1} = F(x_{i+1})$. bc no new information for orthogonal component of δ_{i+1},

$$A_{i+1} w = A_i w$$

for every w such that $\delta_{i+1}^T w = 0$.

$$A_{i+1} = A_i + \frac{(\Delta_{i+1} - A_i \delta_i)\delta_{i+1}^T}{\delta_{i+1}^T \delta_{i+1}}.$$

broydens needs intial guess x_0 and initial approximate jacobian A_0, which can be identity matrix if no better guess.

algorithm

broydens method i

```
x0 = initial vector
A0 = initial matrix
for i = 0, 1, 2, ...
  x[i+1] = x[i] - inv(A[i])F(x[i])
  A[i+1] = A[i] + (Δ[i+1]-A[i]δ[i+1])T(δ[i+1])/(T(δ[i+1])δ[i+1])
end

# where δ[i+1] = x[i+1] - x[i] and Δ[i+1] = F(x[i+1]) - F(x[i]).
```

cheaper

to remove costly solver at $A_i \delta_{i+1} = F(x_i)$, use an inverse.

let $B_i = A_i^{-1}$. then

$$\delta_{i+1} = B_{i+1} \Delta_{i+1},$$

where $\delta_{i+1} = x_{i+1} - x_i, \Delta_{i+1} = F(x_{i+1}) - F(x_i)$ and for every w satisfying $\delta_{i+1}^T w = 0$, still satisfy $A_{i+1} w = A_i w$, or

$$B_{i+1} A_i w = w.$$

a matrix that satisfies these two conditions is

$$B_{i+1} = B_i + \frac{(\delta_{i+1} - B_i \Delta_{i+1})\delta_{i+1}^T B_i}{\delta_{i+1}^T B_i \Delta_{i+1}}.$$

and

$$x_{i+1} = x_i + B_i F(x_i).$$

as with broydens i, ii needs intial guess x_0 and initial jacobian substitute B_0, which can be identity matrix if no better guess.

algorithm

broydens method ii

```
x0 = initial vector
B0 = initial matrix
for i = 0, 1, 2, ...
  x[i+1] = x[i] - B[i]F(x[i])
  B[i+1] = B[i] + (δ[i+1]-B[i]Δ[i+1])T(δ[i+1])B[i]/(T(δ[i+1])B[i]Δ[i+1])
end

# where δ[i] = x[i] - x[i-1] and Δ[i] = F(x[i]) - F(x[i-1]).
```

usw

a perceived disadvantage to broyden ii is that estimates for the jacobian are not easily available. B_i is an estimate for the matrix inverse of the jacobian. broyden i keeps track of A_i which is the estimate of the jacobian. which is why i is "good broyden" and ii is "bad broyden".

like secant to newton, broyden converges slower than newtons multivariate.