Följande tumregel kan användas för Newton-Raphsons metod (observera dock att det är en *tumregel*, inte en allmängiltig matematisk sanning): *antalet korrekta decimaler fördubblas i varje iteration.*

Man kan visa följande:

> Newton-Raphsons metod konvergerar alltid (mot en enkelrot) om startapproximationen x_0 väljs tillräckligt nära roten x^*.

Om man känner ett litet intervall, där roten x^* ligger, är det ofta lätt att verifiera att konvergensvillkoret för Newton-Raphsons metod

$$\left|\varphi'(x)\right| = \left|\frac{f(x)f''(x)}{(f'(x))^2}\right| \le m < 1,$$

är uppfyllt. I regel bryr man sig emellertid inte om att kontrollera konvergens i förväg, eftersom man snabbt upptäcker eventuell divergens. I några viktiga specialfall kan man relativt enkelt bevisa att Newton-Raphsons metod konvergerar för godtyckliga startvärden, se Avsnitt 4.7. Den allmänna teorin för Newton-Raphsons metod kan studeras i böckerna, som nämns i slutet av kapitlet.

4.5 Feluppskattning

När man beräknar approximationer till en rot x^* med hjälp av en iterationsmetod, görs avrundningsfel. Det framgår av Sats 4.4.1 att så länge man inte gör stora fel (så att man hamnar utanför intervallet J) så har dessa fel ingen inverkan alls på den slutliga noggrannheten. Man brukar säga att iterationsmetoder är *självkorrigerande*.

Konvergensordningen hos en iterationsmetod ger en uppfattning om felets uppförande asymptotiskt, dvs efter ett stort antal steg. I praktiken måste man avbryta iterationerna efter ett ändligt antal steg. Vi måste kunna uppskatta felet i en approximation $\bar{x}$ till enkelroten x^*. Med hjälp av medelvärdessatsen får vi

$$f(\bar{x}) = f(\bar{x}) - f(x^*) = (\bar{x} - x^*)f'(\xi),$$

där ξ ligger mellan $\bar{x}$ och x^*. Eftersom x^* är enkelrot så gäller $f'(x^*) \neq 0$, och om $\bar{x}$ är nära x^* så är också $f'(\xi) \neq 0$ (f' förutsätts kontinuerlig). Då får vi

$$|\bar{x} - x^*| = \frac{|f(\bar{x})|}{|f'(\xi)|},$$

och om $|f'(x)| \geq M$ för alla x nära x^* kan vi göra följande feluppskattning

$$|\bar{x} - x^*| \leq \frac{|f(\bar{x})|}{M}.$$

I praktiken beräknas ett närmevärde $\bar{f}(\bar{x})$ till $f(\bar{x})$. Om det absoluta felet i detta värde begränsas av δ, dvs $\left|\bar{f}(\bar{x}) - f(\bar{x})\right| \leq \delta$, så gäller

$$|f(\bar{x})| \leq \left|\bar{f}(\bar{x})\right| + \delta,$$

som vi kan använda i feluppskattningen. Denna feluppskattning gäller oberoende av vilken metod vi har använt för att komma fram till vår approximation $\bar{x}$, och kallas därför **metodoberoende feluppskattning**.

Vi sammanfattar:

Metodoberoende feluppskattning.

$$|\bar{x} - x^*| \leq \frac{\left|\bar{f}(\bar{x})\right| + \delta}{M},$$

där $\bar{f}(\bar{x})$ är ett närmevärde till $f(\bar{x})$, och

$$\left|\bar{f}(\bar{x}) - f(\bar{x})\right| \leq \delta,$$

och där

$$\left|f'(x)\right| \geq M,$$

för alla x nära x^*.

Exempel Vi har löst ekvationen $f(x) = x - \cos x = 0$ med olika metoder i föregående avsnitt och kommit fram till att roten bör vara ungefär 0.73908516. Vi skall nu visa strikt hur noggrann denna approximation är.

Sätt $\bar{x} = 0.73908516$ och beräkna $\bar{f}(\bar{x}) = 4.48274 \cdot 10^{-8}$. Vi har beräknat funktionen i Matlab i IEEE dubbel precision med avrundningsenhet $1.1 \cdot 10^{-16}$; därför är δ av storleksordning 10^{-16}, vilket är försumbart jämfört med funktionsvärdet. Vidare har vi $f'(x) = 1 + \sin x$ så att

$$f'(\bar{x}) \approx 1.673.$$

Vi skall välja M så att $|f'(x)| \geq M$ nära roten. Här tar vi $M = 1.6$. Vi får slutligen

$$|\bar{x} - x^*| \leq \frac{4.5 \cdot 10^{-8}}{1.6} \leq 2.9 \cdot 10^{-8},$$

eller ekvivalent

$$x^* = 0.73908516 \pm 0.29 \cdot 10^{-7} = 0.7390852 \pm 0.7 \cdot 10^{-7}.$$

Genom att använda metodoberoende feluppskattning, kan vi nu skriva en Matlabkod för att lösa en ekvation med en viss absolut noggrannhet.

```
function [x,it]=newton(f,fprim,x0,tol)
% Lös f(x)=0 med noggrannhet tol med Newton-Raphsons
% metod och startvärdet x0
% Iterera tills korrektionen är mindre än tol
it=0;
korr=feval(f,x0)/feval(fprim,x0);
while abs(korr)>tol,
  x=x0-korr
  x0=x;
  it=it+1;
  korr=feval(f,x0)/feval(fprim,x0);
end
```

Om vi anropar denna funktion med

```
>>    [x,it]=newton(@fx,@fxprim,0.7,1e-10)
```

där `fx` och `fxprim` definieras på sidan 73, får vi resultatet

```
x = 0.73908513321516
it = 3
```

vilket faktiskt har ett fel mindre än $0.78 \cdot 10^{-15}$.

Avbrottskriteriet i funktionen `newton` fungerar väl om toleransen är avsevärt större än felet δ i funktionsvärdet; om man vill beräkna roten mycket noggrannt krävs en noggrannare felanalys och ett avbrottskriterium baserad