Modern alkalmazott analízis

Járai Antal

Ezek a programok csak szemléltetésre szolgálnak

Bevezetés

I. Mérték és integrál

▶ 1. Mértékelmélet

▶ 2. Integrálás

II. Funkcionálanalízis

▶ 3. Metrikus terek

▶ 4. Normált terek

▶ 5. Lineáris operátorok

▶ 6. Hilbert–terek

▶ 7. Spektrálelmélet

▶ 8. Kompakt operátorok

▶ 9. Differenciálszámítás

III. Vektoranalízis

```
> x:='x'; fsolve(2*sin(x)=x,x);
```

$$x := x$$

$$0. \qquad (17.1.1)$$

```
> x:=1.;2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);
  2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin
  (%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*
  sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);
  2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin
  (%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*
  sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);2*sin(%);
  2*sin(%);
```

$$x := 1.$$

1.682941970
1.987436530
1.828907553
1.933747642
1.869706154
1.911316179
1.885162348
1.901985210
1.891312851
1.898145620
1.893795924
1.896575152
1.894803507
1.895934551
1.895213162
1.895673550
1.895379846
1.895567260
1.895447689
1.895523984
1.895475305
1.895506365
1.895486548
1.895499192
1.895491125
1.895496272
1.895492988
1.895495083
1.895493746
1.895494599
1.895494055
1.895494402

1.895494181

1.895494322

1.895494232

1.895494289

1.895494253

1.895494276

1.895494261

1.895494271

1.895494265

1.895494268

1.895494266

1.895494268

1.895494266

1.895494268

1.895494266 (17.1.2)

```
> halfing:=proc(a,b,f::procedure,eps,feps,N) local aa,bb,ab,ff,
  delta,n;
    if f(a)<0 then aa:=a; bb:=b; else aa:=b; bb:=a; fi;
    if f(a)*f(b)>0 then error "bad interval" fi;
    delta:=abs(a-b);
    for n to N do
      ab:=(aa+bb)/2.;
      delta:=delta/2.;
      ff:=f(ab);
      if abs(ff)<feps and delta<eps then return ab fi;
      if ff<0 then aa:=ab else bb:=ab fi;
    od;
    FAIL;
  end;
```

$$halfing := \mathbf{proc}(a, b, f{::}procedure, eps, feps, N) \quad (17.1.3)$$

$$\mathbf{local}\ aa, bb, ab, ff, \delta, n;$$

$$\mathbf{if}\ f(a) < 0\ \mathbf{then}$$

$$aa := a;$$

$$bb := b$$

$$\mathbf{else}$$

$$aa := b;$$

$$bb := a$$

```
    end if;
    if 0 < f(a)*f(b) then
        error "bad interval"
    end if;
    δ := abs(a - b);
    for n to N do
        ab := (aa + bb) / 2.;
        δ := δ / 2.;
        ff := f(ab);
        if abs(ff) < feps and δ < eps then
            return ab
        end if;
        if ff < 0 then
            aa := ab
        else
            bb := ab
        end if
    end do;
    FAIL
end proc
```

```
> halfing(-1,1,x->x^2,1,1,10);
Error, (in halfing) bad interval
> halfing(1.,2.,x->2*sin(x)-x,0.0000001,100,10);
```

$$FAIL \tag{17.1.4}$$

```
> debug(halfing); halfing(1.,2.,x->2*sin(x)-x,0.000001,100,20);
```

$$halfing$$

```
{--> enter halfing, args = 1., 2., proc (x) options
operator, arrow; 2*sin(x)-x end proc, 0.1e-5, 100, 20
```

$$aa := 2.$$

$$bb := 1.$$

$$\delta := 1.$$

$$ab := 1.500000000$$

$$\delta := 0.5000000000$$

$$ff := 0.494989973$$

$$bb := 1.500000000$$
$$ab := 1.750000000$$
$$\delta := 0.2500000000$$
$$ff := 0.217971894$$
$$bb := 1.750000000$$
$$ab := 1.875000000$$
$$\delta := 0.1250000000$$
$$ff := 0.033171563$$
$$bb := 1.875000000$$
$$ab := 1.937500000$$
$$\delta := 0.06250000000$$
$$ff := -0.070471438$$
$$aa := 1.937500000$$
$$ab := 1.906250000$$
$$\delta := 0.03125000000$$
$$ff := -0.017727883$$
$$aa := 1.906250000$$
$$ab := 1.890625000$$
$$\delta := 0.01562500000$$
$$ff := 0.007953596$$
$$bb := 1.890625000$$
$$ab := 1.898437500$$
$$\delta := 0.007812500000$$
$$ff := -0.004829355$$
$$aa := 1.898437500$$
$$ab := 1.894531250$$
$$\delta := 0.003906250000$$
$$ff := 0.001576586$$
$$bb := 1.894531250$$
$$ab := 1.896484375$$
$$\delta := 0.001953125000$$
$$ff := -0.001622770$$

$$aa := 1.896484375$$
$$ab := 1.895507812$$
$$\delta := 0.0009765625000$$
$$ff := -0.000022187$$
$$aa := 1.895507812$$
$$ab := 1.895019531$$
$$\delta := 0.0004882812500$$
$$ff := 0.000777425$$
$$bb := 1.895019531$$
$$ab := 1.895263672$$
$$\delta := 0.0002441406250$$
$$ff := 0.000377675$$
$$bb := 1.895263672$$
$$ab := 1.895385742$$
$$\delta := 0.0001220703125$$
$$ff := 0.000177758$$
$$bb := 1.895385742$$
$$ab := 1.895446777$$
$$\delta := 0.00006103515625$$
$$ff := 0.000077789$$
$$bb := 1.895446777$$
$$ab := 1.895477294$$
$$\delta := 0.00003051757812$$
$$ff := 0.000027802$$
$$bb := 1.895477294$$
$$ab := 1.895492553$$
$$\delta := 0.00001525878906$$
$$ff := 0.000002808$$
$$bb := 1.895492553$$
$$ab := 1.895500182$$
$$\delta := 0.000007629394530$$
$$ff := -0.000009689$$

$$aa := 1.895500182$$

$$ab := 1.895496368$$

$$\delta := 0.000003814697265$$

$$ff := -0.000003441$$

$$aa := 1.895496368$$

$$ab := 1.895494460$$

$$\delta := 0.000001907348632$$

$$ff := -3.16\,10^{-7}$$

$$aa := 1.895494460$$

$$ab := 1.895493506$$

$$\delta := 9.536743160\,10^{-7}$$

$$ff := 0.000001247$$

```
<-- exit halfing (now at top level) = 1.895493506}
```

$$1.895493506 \tag{17.1.5}$$

```
> regulafalsi:=proc(a,b,f::procedure,eps,feps,N) local aa,bb,
  ab,ff,fa,fb,delta,n;
    if f(a)<0 then aa:=a; bb:=b; else aa:=b; bb:=a; fi;
    fa:=f(aa); fb:=f(bb);
    if fa*fb>0 then error "bad interval" fi;
    delta:=abs(aa-bb);
    for n to N do
      ab:=aa-fa/(fb-fa)*(bb-aa);
      ff:=f(ab);
      if ff<0 then delta:=abs(aa-ab); aa:=ab; fa:=ff;
      else delta:=abs(ab-bb); bb:=ab; fb:=ff fi;
      if abs(ff)<feps and delta<eps then return ab fi;
    od;
    FAIL;
  end;
```

$$regulafalsi := \mathbf{proc}(a, b, f::procedure, eps, feps, N) \tag{17.1.6}$$

$$\mathbf{local}\ aa, bb, ab, ff, fa, fb, \delta, n;$$

$$\mathbf{if}\ f(a) < 0\ \mathbf{then}$$

$$aa := a;$$

$$bb := b$$

$$\mathbf{else}$$

$$aa := b;$$

$$bb := a$$

```
    end if;
    fa := f(aa);
    fb := f(bb);
    if 0 < fa*fb then
        error "bad interval"
    end if;
    δ := abs(aa - bb);
    for n to N do
        ab := aa - fa*(bb - aa)/(fb - fa);
        ff := f(ab);
        if ff < 0 then
            δ := abs(aa - ab);
            aa := ab;
            fa := ff
        else
            δ := abs(ab - bb);
            bb := ab;
            fb := ff
        end if;
        if abs(ff) < feps and δ < eps then
            return ab
        end if
    end do;
    FAIL
end proc
```

```
> regulafalsi(-1,1,x->x^2,1,1,10);
Error, (in regulafalsi) bad interval
> regulafalsi(1.,2.,x->2*sin(x)-x,0.000001,100,7);
                          1.895494264                          (17.1.7)

> secant:=proc(a,b,f::procedure,eps,feps,N) local aa,bb,ab,ff,
  fa,fb,delta,n;
    aa:=a; bb:=b;
    fa:=f(aa); fb:=f(bb);
    delta:=abs(aa-bb);
    for n to N do
```

```
        ab:=aa-fa/(fb-fa)*(bb-aa);
        ff:=f(ab);
        aa:=bb; fa:=fb; bb:=ab; fb:=ff;
        delta:=abs(aa-bb);
        if abs(ff)<feps and delta<eps then return ab fi;
      od;
      FAIL;
    end;
```

$$secant := \mathbf{proc}(a, b, f::procedure, eps, feps, N) \tag{17.1.8}$$

local $aa, bb, ab, ff, fa, fb, \delta, n;$

$aa := a;$

$bb := b;$

$fa := f(aa);$

$fb := f(bb);$

$\delta := \mathrm{abs}(aa - bb);$

for n **to** N **do**

$ab := aa - fa * (bb - aa) / (fb - fa);$

$ff := f(ab);$

$aa := bb;$

$fa := fb;$

$bb := ab;$

$fb := ff;$

$\delta := \mathrm{abs}(aa - bb);$

if $\mathrm{abs}(ff) < feps$ **and** $\delta < eps$ **then**

return ab

end if

end do;

$FAIL$

end proc

```
> secant(-1.,1.,x->x^2,10.,10.,10);
```

$$FAIL \tag{17.1.9}$$

```
> debug(secant); secant(-1.,1.,x->x^2,10.,10.,2);
```

$$secant$$

```
{--> enter secant, args = -1., 1., proc (x) options
operator, arrow; x^2 end proc, 10., 10., 2
```

$$aa := -1.$$

$$bb := 1.$$

$$fa := 1.$$

$$fb := 1.$$

$$\delta := 2.$$

$$ab := -Float(\infty)$$

$$ff := Float(\infty)$$

$$aa := 1.$$

$$fa := 1.$$

$$bb := -Float(\infty)$$

$$fb := Float(\infty)$$

$$\delta := Float(\infty)$$

$$ab := Float(undefined)$$

$$ff := Float(undefined)$$

$$aa := -Float(\infty)$$

$$fa := Float(\infty)$$

$$bb := Float(undefined)$$

$$fb := Float(undefined)$$

$$\delta := Float(undefined)$$

$$FAIL$$

```
<-- exit secant (now at top level) = FAIL}
```

$$FAIL \tag{17.1.10}$$

```
> secant(-1.,0.5,x->x^2,0.01,100.,10);
{--> enter secant, args = -1., .5, proc (x) options
operator, arrow; x^2 end proc, 0.1e-1, 100., 10
```

$$aa := -1.$$

$$bb := 0.5$$

$$fa := 1.$$

$$fb := 0.25$$

$$\delta := 1.5$$

$$ab := 1.000000000$$

$$ff := 1.000000000$$

$$aa := 0.5$$

$$fa := 0.25$$

$$bb := 1.000000000$$
$$fb := 1.000000000$$
$$\delta := 0.500000000$$
$$ab := 0.3333333334$$
$$ff := 0.1111111112$$
$$aa := 1.000000000$$
$$fa := 1.000000000$$
$$bb := 0.3333333334$$
$$fb := 0.1111111112$$
$$\delta := 0.6666666666$$
$$ab := 0.2500000001$$
$$ff := 0.06250000005$$
$$aa := 0.3333333334$$
$$fa := 0.1111111112$$
$$bb := 0.2500000001$$
$$fb := 0.06250000005$$
$$\delta := 0.0833333333$$
$$ab := 0.1428571430$$
$$ff := 0.02040816331$$
$$aa := 0.2500000001$$
$$fa := 0.06250000005$$
$$bb := 0.1428571430$$
$$fb := 0.02040816331$$
$$\delta := 0.1071428571$$
$$ab := 0.0909090909$$
$$ff := 0.008264462808$$
$$aa := 0.1428571430$$
$$fa := 0.02040816331$$
$$bb := 0.0909090909$$
$$fb := 0.008264462808$$
$$\delta := 0.0519480521$$
$$ab := 0.05555555558$$

$$ff := 0.003086419756$$

$$aa := 0.0909090909$$

$$fa := 0.008264462808$$

$$bb := 0.05555555558$$

$$fb := 0.003086419756$$

$$\delta := 0.03535353532$$

$$ab := 0.03448275864$$

$$ff := 0.001189060643$$

$$aa := 0.05555555558$$

$$fa := 0.003086419756$$

$$bb := 0.03448275864$$

$$fb := 0.001189060643$$

$$\delta := 0.02107279694$$

$$ab := 0.02127659577$$

$$ff := 0.0004526935276$$

$$aa := 0.03448275864$$

$$fa := 0.001189060643$$

$$bb := 0.02127659577$$

$$fb := 0.0004526935276$$

$$\delta := 0.01320616287$$

$$ab := 0.01315789474$$

$$ff := 0.0001731301940$$

$$aa := 0.02127659577$$

$$fa := 0.0004526935276$$

$$bb := 0.01315789474$$

$$fb := 0.0001731301940$$

$$\delta := 0.00811870103$$

```
<-- exit secant (now at top level) = 0.1315789474e-1}
```

$$0.01315789474 \tag{17.1.11}$$

```
> secant(1.,2.,x->2*sin(x)-x,0.000001,100,10);
{--> enter secant, args = 1., 2., proc (x) options
operator, arrow; 2*sin(x)-x end proc, 0.1e-5, 100, 10
```

$$aa := 1.$$

$$bb := 2.$$

$$fa := 0.682941970$$
$$fb := -.181405146$$
$$\delta := 1.$$
$$ab := 1.790124659$$
$$ff := 0.161962955$$
$$aa := 2.$$
$$fa := -.181405146$$
$$bb := 1.790124659$$
$$fb := 0.161962955$$
$$\delta := 0.209875341$$
$$ab := 1.889120548$$
$$ff := 0.010401910$$
$$aa := 1.790124659$$
$$fa := 0.161962955$$
$$bb := 1.889120548$$
$$fb := 0.010401910$$
$$\delta := 0.098995889$$
$$ab := 1.895914816$$
$$ff := -0.000689046$$
$$aa := 1.889120548$$
$$fa := 0.010401910$$
$$bb := 1.895914816$$
$$fb := -0.000689046$$
$$\delta := 0.006794268$$
$$ab := 1.895492710$$
$$ff := 0.000002550$$
$$aa := 1.895914816$$
$$fa := -0.000689046$$
$$bb := 1.895492710$$
$$fb := 0.000002550$$
$$\delta := 0.000422106$$
$$ab := 1.895494266$$

$$ff := 2.\,10^{-9}$$

$$aa := 1.895492710$$

$$fa := 0.000002550$$

$$bb := 1.895494266$$

$$fb := 2.\,10^{-9}$$

$$\delta := 0.000001556$$

$$ab := 1.895494267$$

$$ff := 0.$$

$$aa := 1.895494266$$

$$fa := 2.\,10^{-9}$$

$$bb := 1.895494267$$

$$fb := 0.$$

$$\delta := 1.\,10^{-9}$$

```
<-- exit secant (now at top level) = 1.895494267}
```

$$1.895494267 \qquad (17.1.12)$$

```
> debug(secant); secant(1.,2.,x->2*sin(x)-x,0.000001,100,6);
```

secant

```
{--> enter secant, args = 1., 2., proc (x) options
operator, arrow; 2*sin(x)-x end proc, 0.1e-5, 100, 6
```

$$aa := 1.$$

$$bb := 2.$$

$$fa := 0.682941970$$

$$fb := -.181405146$$

$$\delta := 1.$$

$$ab := 1.790124659$$

$$ff := 0.161962955$$

$$aa := 2.$$

$$fa := -.181405146$$

$$bb := 1.790124659$$

$$fb := 0.161962955$$

$$\delta := 0.209875341$$

$$ab := 1.889120548$$

$$ff := 0.010401910$$

$$aa := 1.790124659$$
$$fa := 0.161962955$$
$$bb := 1.889120548$$
$$fb := 0.010401910$$
$$\delta := 0.098995889$$
$$ab := 1.895914816$$
$$ff := -0.000689046$$
$$aa := 1.889120548$$
$$fa := 0.010401910$$
$$bb := 1.895914816$$
$$fb := -0.000689046$$
$$\delta := 0.006794268$$
$$ab := 1.895492710$$
$$ff := 0.000002550$$
$$aa := 1.895914816$$
$$fa := -0.000689046$$
$$bb := 1.895492710$$
$$fb := 0.000002550$$
$$\delta := 0.000422106$$
$$ab := 1.895494266$$
$$ff := 2.\,10^{-9}$$
$$aa := 1.895492710$$
$$fa := 0.000002550$$
$$bb := 1.895494266$$
$$fb := 2.\,10^{-9}$$
$$\delta := 0.000001556$$
$$ab := 1.895494267$$
$$ff := 0.$$
$$aa := 1.895494266$$
$$fa := 2.\,10^{-9}$$
$$bb := 1.895494267$$
$$fb := 0.$$

$$\delta := 1.\,10^{-9}$$

```
<-- exit secant (now at top level) = 1.895494267}
```

$$1.895494267 \tag{17.1.13}$$

```
> Student[Calculus1][NewtonsMethodTutor](x-cos(x),0);
Error, (in Student:-Calculus1:-NewtonsMethodTutor) the
derivative is 0 after the 1st iteration
> Newton:=proc(x0,f::procedure,fp::procedure,eps,feps,N) local
  x,xx,delta,n;
  x:=x0;
  for n to N do
    xx:=x-evalf(f(x)/fp(x));
    delta:=abs(x-xx);
    if abs(f(xx))<feps and delta<eps then return xx fi;
    x:=xx;
  od; FAIL; end;
```

$$Newton := \mathbf{proc}(x0,\ f{::}procedure,\ fp{::}procedure,\ eps,\ feps,\ N) \tag{17.1.14}$$

$$\mathbf{local}\ x,\ xx,\ \delta,\ n;$$

$$x := x0;$$

$$\mathbf{for}\ n\ \mathbf{to}\ N\ \mathbf{do}$$

$$xx := x - evalf(f(x)\ /\ fp(x));$$

$$\delta := \mathrm{abs}(x - xx);$$

$$\mathbf{if}\ \mathrm{abs}(f(xx)) < feps\ \mathbf{and}\ \delta < eps\ \mathbf{then}$$

$$\mathbf{return}\ xx$$

$$\mathbf{end\ if};$$

$$x := xx$$

$$\mathbf{end\ do};$$

$$FAIL$$

$$\mathbf{end\ proc}$$

```
> Newton(1.,x->2*sin(x)-x,x->2*cos(x)-1,0.000001,100,6);
```

$$FAIL \tag{17.1.15}$$

```
> debug(Newton); Newton(1.,x->2*sin(x)-x,x->2*cos(x)-1,
  0.000001,100,6);
```

$$Newton$$

```
{--> enter Newton, args = 1., proc (x) options operator,
arrow; 2*sin(x)-x end proc, proc (x) options operator,
arrow; 2*cos(x)-1 end proc, 0.1e-5, 100, 6
```

$$x := 1.$$

$$xx := -7.472740617$$

$$\delta := 8.472740617$$

$$x := -7.472740617$$

$$xx := 14.47852462$$

$$\delta := 21.95126524$$

$$x := 14.47852462$$

$$xx := 6.935146381$$

$$\delta := 7.543378239$$

$$x := 6.935146381$$

$$xx := 16.63630229$$

$$\delta := 9.701155909$$

$$x := 16.63630229$$

$$xx := 8.340204744$$

$$\delta := 8.296097546$$

$$x := 8.340204744$$

$$xx := 4.943086934$$

$$\delta := 3.397117810$$

$$x := 4.943086934$$

$$\mathit{FAIL}$$

```
<-- exit Newton (now at top level) = FAIL}
```

$$\mathit{FAIL} \tag{17.1.16}$$

```
> Newton(1.5,x->2*sin(x)-x,x->2*cos(x)-1,0.000001,100,6);
{--> enter Newton, args = 1.5, proc (x) options operator,
arrow; 2*sin(x)-x end proc, proc (x) options operator,
arrow; 2*cos(x)-1 end proc, 0.1e-5, 100, 6
```

$$x := 1.5$$

$$xx := 2.076558200$$

$$\delta := 0.576558200$$

$$x := 2.076558200$$

$$xx := 1.910506616$$

$$\delta := 0.166051584$$

$$x := 1.910506616$$

$$xx := 1.895622003$$

$$\delta := 0.014884613$$

$$x := 1.895622003$$

$$xx := 1.895494276$$

$$\delta := 0.000127727$$

$$x := 1.895494276$$

$$xx := 1.895494267$$

$$\delta := 9.\,10^{-9}$$

```
<-- exit Newton (now at top level) = 1.895494267}
```

$$1.895494267 \tag{17.1.17}$$

```
> modNewton:=proc(x0,f::procedure,q,eps,feps,N) local x,xx,
  delta,n;
  x:=x0;
  for n to N do
    xx:=x-evalf(f(x)/q);
    delta:=abs(x-xx);
    if abs(f(xx))<feps and delta<eps then return xx fi;
    x:=xx;
  od; FAIL; end;
```

$$modNewton := \mathbf{proc}(x0,\ f::procedure,\ q,\ eps,\ feps,\ N) \tag{17.1.18}$$

$$\mathbf{local}\ x,\ xx,\ \delta,\ n;$$

$$x := x0;$$

$$\mathbf{for}\ n\ \mathbf{to}\ N\ \mathbf{do}$$

$$xx := x - evalf(f(x)\ /\ q);$$

$$\delta := \mathrm{abs}(x - xx);$$

$$\mathbf{if}\ \mathrm{abs}(f(xx)) < feps\ \mathbf{and}\ \delta < eps\ \mathbf{then}$$

$$\mathbf{return}\ xx$$

$$\mathbf{end\ if};$$

$$x := xx$$

$$\mathbf{end\ do};$$

$$FAIL$$

$$\mathbf{end\ proc}$$

```
> modNewton(2.,x->2*sin(x)-x,2*cos(2.)-1,0.000001,100,10);
```

$$1.895494338 \tag{17.1.19}$$

```
> debug(modNewton); modNewton(1.5,x->2*sin(x)-x,2*cos(1.5)-1,
  0.000001,100,10);
```

$$modNewton$$

```
{--> enter modNewton, args = 1.5, proc (x) options
operator, arrow; 2*sin(x)-x end proc, -.8585255967, 0.1e
-5, 100, 10
```

$x := 1.5$

$xx := 2.076558200$

$\delta := 0.576558200$

$x := 2.076558200$

$xx := 1.695734425$

$\delta := 0.380823775$

$x := 1.695734425$

$xx := 2.031981058$

$\delta := 0.336246633$

$x := 2.031981058$

$xx := 1.751349659$

$\delta := 0.280631399$

$x := 1.751349659$

$xx := 2.003106043$

$\delta := 0.251756384$

$x := 2.003106043$

$xx := 1.785168338$

$\delta := 0.217937705$

$x := 1.785168338$

$xx := 1.982078046$

$\delta := 0.196909708$

$x := 1.982078046$

$xx := 1.808688088$

$\delta := 0.173389958$

$x := 1.808688088$

$xx := 1.965917892$

$\delta := 0.157229804$

$x := 1.965917892$

$xx := 1.826122019$

$\delta := 0.139795873$

$x := 1.826122019$

FAIL

```
<-- exit modNewton (now at top level) = FAIL}
```

$$FAIL \tag{17.1.20}$$

```
>
```

*17.27. Egy példa: a konvergencia rendje.

```
> quasiNewton:=proc(x0,f::procedure,fp::procedure,alpha,eps,
  feps,N) local x,xx,delta,n;
  x:=x0;
  for n to N do
    xx:=x-alpha*evalf(f(x)/fp(x));
    delta:=abs(x-xx);
    if abs(f(xx))<feps and delta<eps then return xx fi;
    x:=xx;
  od; FAIL; end;
```

$$quasiNewton := \mathbf{proc}(x0,\ f::procedure,\ fp::procedure,\ \alpha,\ eps,\ feps,\ N) \tag{17.2.1}$$

$$\begin{aligned}
&\mathbf{local}\ x,\ xx,\ \delta,\ n;\\
&x := x0;\\
&\mathbf{for}\ n\ \mathbf{to}\ N\ \mathbf{do}\\
&\quad xx := x - \alpha * evalf(f(x)\ /\ fp(x));\\
&\quad \delta := \mathrm{abs}(x - xx);\\
&\quad \mathbf{if}\ \mathrm{abs}(f(xx)) < feps\ \mathbf{and}\ \delta < eps\ \mathbf{then}\\
&\qquad \mathbf{return}\ xx\\
&\quad \mathbf{end\ if};\\
&\quad x := xx\\
&\mathbf{end\ do};\\
&FAIL\\
&\mathbf{end\ proc}
\end{aligned}$$

```
> debug(quasiNewton);
  quasiNewton(1,x->sin(x)^4,x->4*sin(x)^3*cos(x),1.,0.000001,
  100,10);
```

$$quasiNewton$$

```
{--> enter quasiNewton, args = 1, proc (x) options
operator, arrow; sin(x)^4 end proc, proc (x) options
operator, arrow; 4*sin(x)^3*cos(x) end proc, 1., 0.1e-5,
100, 10
```

$$x := 1$$

$$xx := 0.6106480688$$

$$\delta := 0.3893519312$$

$$x := 0.6106480688$$
$$xx := 0.4356770834$$
$$\delta := 0.1749709854$$
$$x := 0.4356770834$$
$$xx := 0.3192995341$$
$$\delta := 0.1163775493$$
$$x := 0.3192995341$$
$$xx := 0.2366464853$$
$$\delta := 0.0826530488$$
$$x := 0.2366464853$$
$$xx := 0.1763551704$$
$$\delta := 0.0602913149$$
$$x := 0.1763551704$$
$$xx := 0.1318035485$$
$$\delta := 0.0445516219$$
$$x := 0.1318035485$$
$$xx := 0.09866051655$$
$$\delta := 0.03314303195$$
$$x := 0.09866051655$$
$$xx := 0.07391504530$$
$$\delta := 0.02474547125$$
$$x := 0.07391504530$$
$$xx := 0.05540255777$$
$$\delta := 0.01851248753$$
$$x := 0.05540255777$$
$$xx := 0.04153772965$$
$$\delta := 0.01386482812$$
$$x := 0.04153772965$$
$$\mathit{FAIL}$$

```
<-- exit quasiNewton (now at top level) = FAIL}
```

$$\mathit{FAIL} \tag{17.2.2}$$

```
> quasiNewton(1,x->sin(x)^4,x->4*sin(x)^3*cos(x),4.,0.000001,
  100,10);
```

```
{--> enter quasiNewton, args = 1, proc (x) options
operator, arrow; sin(x)^4 end proc, proc (x) options
operator, arrow; 4*sin(x)^3*cos(x) end proc, 4., 0.1e-5,
100, 10
```

$$x := 1$$

$$xx := -.557407725$$

$$\delta := 1.557407725$$

$$x := -.557407725$$

$$xx := 0.0659364522$$

$$\delta := 0.6233441772$$

$$x := 0.0659364522$$

$$xx := -0.00009572196$$

$$\delta := 0.06603217416$$

$$x := -0.00009572196$$

$$xx := 2.8\,10^{-13}$$

$$\delta := 0.00009572196028$$

$$x := 2.8\,10^{-13}$$

$$xx := 0.$$

$$\delta := 2.8\,10^{-13}$$

```
<-- exit quasiNewton (now at top level) = 0.}
```

$$0. \qquad (17.2.3)$$

```
> adaptiveNewton:=proc(x0,f::procedure,fp::procedure,eps,feps,
  N)
  local x,xx,xxx,alpha,delta,q,n,m;
  x:=x0; alpha:=1.;
  for n to N do
    xx:=x-alpha*evalf(f(x)/fp(x));
    xxx:=xx-alpha*evalf(f(xx)/fp(xx));
    delta:=abs(xxx-xx);
    q:=(xxx-xx)/(xx-x);
    m:=alpha/(1-q); alpha:=m;
    if abs(f(xxx))<feps and delta<eps then return xxx,alpha fi;
    x:=xxx;
  od; FAIL; end;
```

$$adaptiveNewton := \mathbf{proc}(x0,\ f::procedure,\ fp::procedure,\ eps,\ feps,\ N) \qquad (17.2.4)$$

$$\mathbf{local}\ x,\ xx,\ xxx,\ \alpha,\ \delta,\ q,\ n,\ m;$$

$$x := x0;$$

$$\alpha := 1.;$$

```
    for n to N do
        xx := x - α * evalf(f(x) / fp(x));
        xxx := xx - α * evalf(f(xx) / fp(xx));
        δ := abs(xxx - xx);
        q := (xxx - xx) / (xx - x);
        m := α / (1 - q);
        α := m;
        if abs(f(xxx)) < feps and δ < eps then
            return xxx, α
        end if;
        x := xxx
    end do;
    FAIL
end proc
```

```
> adaptiveNewton(1.,x->sin(x)^4,x->4*sin(x)^3*cos(x),0.000001,
  100,10);
```

$$1.714\,10^{-20},\ 4.000000000 \tag{17.2.5}$$

```
> debug(adaptiveNewton);
  adaptiveNewton(1.,x->sin(x)^4,x->4*sin(x)^3*cos(x),0.000001,
  100,10);
```

adaptiveNewton

```
{--> enter adaptiveNewton, args = 1., proc (x) options
operator, arrow; sin(x)^4 end proc, proc (x) options
operator, arrow; 4*sin(x)^3*cos(x) end proc, 0.1e-5, 100,
10
```

$$x := 1.$$

$$\alpha := 1.$$

$$xx := 0.6106480687$$

$$xxx := 0.4356770833$$

$$\delta := 0.1749709854$$

$$q := 0.4493903108$$

$$m := 1.816168548$$

$$\alpha := 1.816168548$$

$$x := 0.4356770833$$

$$xx := 0.2243158386$$

$$xxx := 0.1207236359$$

$$\delta := 0.1035922027$$

$$q := 0.4901191931$$

$$m := 3.561947270$$

$$\alpha := 3.561947270$$

$$x := 0.1207236359$$

$$xx := 0.0126955112$$

$$xxx := 0.00138971842$$

$$\delta := 0.01130579278$$

$$q := 0.1046560126$$

$$m := 3.978300318$$

$$\alpha := 3.978300318$$

$$x := 0.00138971842$$

$$xx := 0.000007538222$$

$$xxx := 4.0894255\ 10^{-8}$$

$$\delta := 0.000007497327745$$

$$q := 0.005424276629$$

$$m := 3.999997410$$

$$\alpha := 3.999997410$$

$$x := 4.0894255\ 10^{-8}$$

$$xx := 2.648\ 10^{-14}$$

$$xxx := 1.714\ 10^{-20}$$

$$\delta := 2.647998286\ 10^{-14}$$

$$q := 6.475237171\ 10^{-7}$$

$$m := 4.000000000$$

$$\alpha := 4.000000000$$

```
<-- exit adaptiveNewton (now at top level) = 0.1714e-19,
4.000000000}
```

$$1.714\ 10^{-20}, 4.000000000 \tag{17.2.6}$$

▶ *17.28. Feladat.

▶ *17.29. Feladat.

*17.30. Feladat.

*17.31. Feladat.

*Feladat.

*Feladat.

*Feladat.

```
> alpha:=1.*10^(-12); beta:=0.025; Gamma:=0.4; mu:=0.001; T:=
  .4; R:=3333.; e:=evalf(exp(1.));
```

$$\alpha := 1.000000000\ 10^{-12}$$

$$\beta := 0.025$$

$$\Gamma := 0.4$$

$$\mu := 0.001$$

$$T := 0.4$$

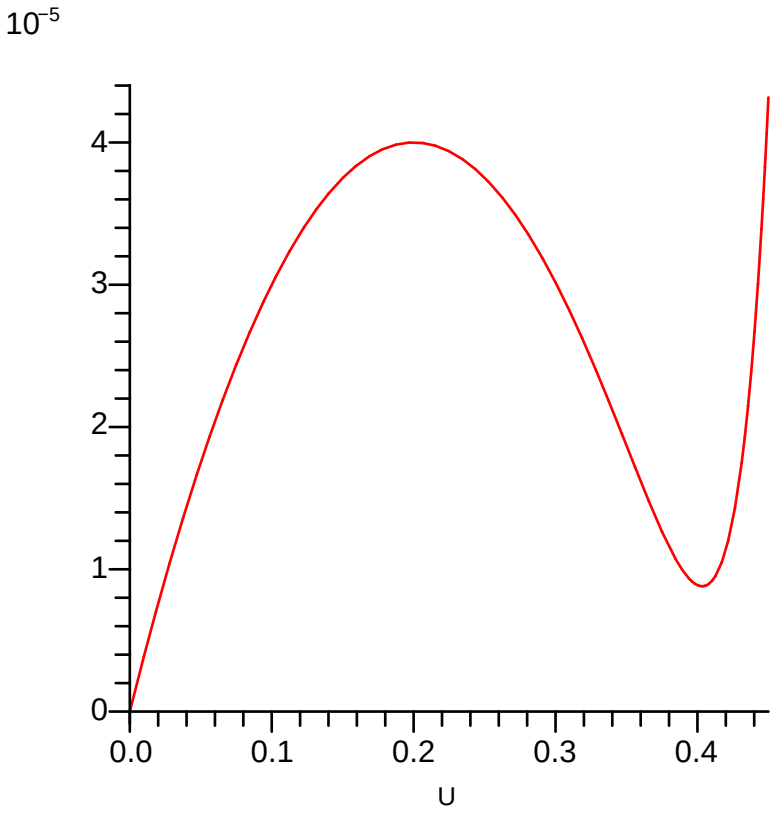
$$R := 3333.$$

$$e := 2.718281828 \tag{17.9.1}$$

```
> i:=alpha*(e^(U/beta)-1)-mu*U*(U-Gamma);
```

$$i := 1.000000000\ 10^{-12}\ 2.718281828^{40.00000000\,U} - 1.000000000\ 10^{-12} - 0.001\ U\left(U - 0.4\right) \tag{17.9.2}$$

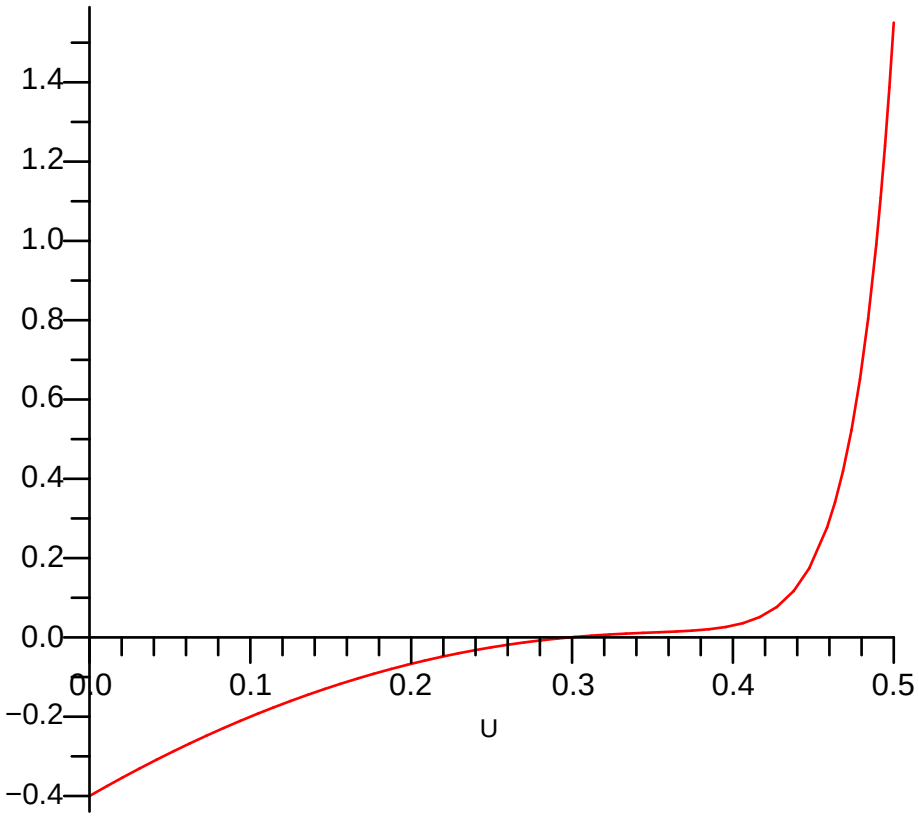
```
> plot(i,U=0.0..0.45);
```




```
> f:=i*R+U-T;
```

$$f := 3.333000000\ 10^{-9}\ 2.718281828^{40.00000000\ U} - 0.4000000033 - 3.333\ U(U-0.4) + U \tag{17.9.3}$$

```
> plot(f,U=0..0.5);
```




```
> halfing(0.2,0.4,x->subs(U=x,f),0.0001,1,30);
{--> enter halfing, args = .2, .4, proc (x) options
operator, arrow; subs(U = x, f) end proc, 0.1e-3, 1, 30
```

$$aa := 0.2$$

$$bb := 0.4$$

$$\delta := 0.2$$

$$ab := 0.3000000000$$

$$\delta := 0.1000000000$$

$$ff := 0.0005324584$$

$$bb := 0.3000000000$$

$$ab := 0.2500000000$$

$$\delta := 0.05000000000$$

$$ff := -0.0249390891$$

$$aa := 0.2500000000$$

$$ab := 0.2750000000$$
$$\delta := 0.02500000000$$
$$ff := -0.0102285678$$
$$aa := 0.2750000000$$
$$ab := 0.2875000000$$
$$\delta := 0.01250000000$$
$$ff := -0.0043692648$$
$$aa := 0.2875000000$$
$$ab := 0.2937500000$$
$$\delta := 0.006250000000$$
$$ff := -0.0018014790$$
$$aa := 0.2937500000$$
$$ab := 0.2968750000$$
$$\delta := 0.003125000000$$
$$ff := -0.0006057063$$
$$aa := 0.2968750000$$
$$ab := 0.2984375000$$
$$\delta := 0.001562500000$$
$$ff := -0.0000294824$$
$$aa := 0.2984375000$$
$$ab := 0.2992187500$$
$$\delta := 0.0007812500000$$
$$ff := 0.0002532656$$
$$bb := 0.2992187500$$
$$ab := 0.2988281250$$
$$\delta := 0.0003906250000$$
$$ff := 0.0001123370$$
$$bb := 0.2988281250$$
$$ab := 0.2986328125$$
$$\delta := 0.0001953125000$$
$$ff := 0.0000415387$$
$$bb := 0.2986328125$$

$$ab := 0.2985351562$$

$$\delta := 0.00009765625000$$

$$ff := 0.0000060560$$

```
<-- exit halfing (now at top level) = .2985351562}
```

$$0.2985351562 \tag{17.9.4}$$

```
> T:=0.6; R:=12000.;
```

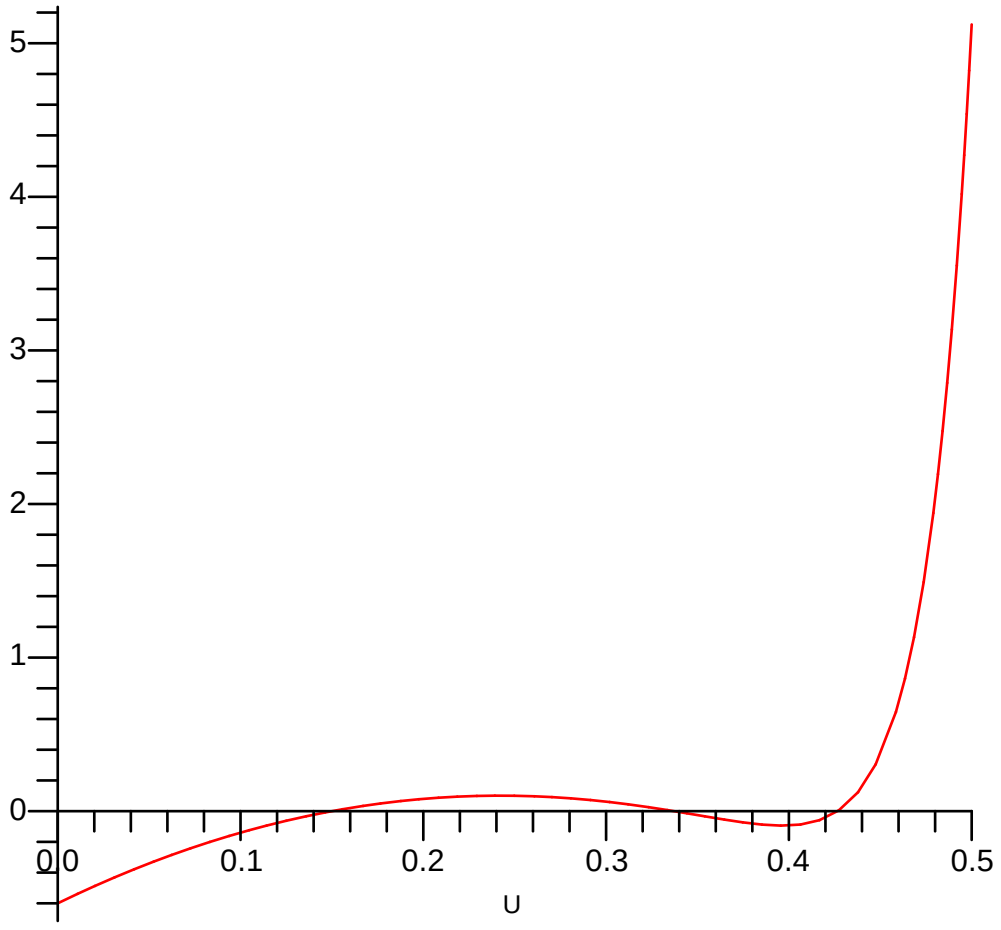
$$T := 0.6$$

$$R := 12000. \tag{17.9.5}$$

```
> f:=i*R+U-T;
```

$$f := 1.200000000\ 10^{-8}\ 2.718281828^{40.00000000\,U} - 0.6000000120 - 12.000\,U(U-0.4) + U \tag{17.9.6}$$

```
> plot(f,U=0..0.5);
```




```
> ff:=x->subs(U=x,f);
```

$$ff := x \to subs(U = x,\ f) \tag{17.9.7}$$

```
> halfing(0.2,0.4,ff,0.0001,1,30);
{--> enter halfing, args = .2, .4, ff, 0.1e-3, 1, 30
```

$$aa := 0.4$$

$$bb := 0.2$$

$$\delta := 0.2$$

$$ab := 0.3000000000$$

$$\delta := 0.1000000000$$

$$ff := 0.0619530455$$

$$bb := 0.3000000000$$

$$ab := 0.3500000000$$

$$\delta := 0.05000000000$$

$$ff := -0.0255687606$$

$$aa := 0.3500000000$$

$$ab := 0.3250000000$$

$$\delta := 0.02500000000$$

$$ff := 0.0228089487$$

$$bb := 0.3250000000$$

$$ab := 0.3375000000$$

$$\delta := 0.01250000000$$

$$ff := -0.0006220156$$

$$aa := 0.3375000000$$

$$ab := 0.3312500000$$

$$\delta := 0.006250000000$$

$$ff := 0.0113480785$$

$$bb := 0.3312500000$$

$$ab := 0.3343750000$$

$$\delta := 0.003125000000$$

$$ff := 0.0054197928$$

$$bb := 0.3343750000$$

$$ab := 0.3359375000$$

$$\delta := 0.001562500000$$

$$ff := 0.0024121203$$

$$bb := 0.3359375000$$

$$ab := 0.3367187500$$

$$\delta := 0.0007812500000$$

$$ff := 0.0008982338$$

$$bb := 0.3367187500$$

$$ab := 0.3371093750$$

$$\delta := 0.0003906250000$$

$$ff := 0.0001388883$$

$$bb := 0.3371093750$$

$$ab := 0.3373046875$$

$$\delta := 0.0001953125000$$

$$ff := -0.0002413709$$

$$aa := 0.3373046875$$

$$ab := 0.3372070312$$

$$\delta := 0.00009765625000$$

$$ff := -0.0000511928$$

```
<-- exit halfing (now at top level) = .3372070312}
```

$$0.3372070312 \tag{17.9.8}$$

```
> halfing(0.1,0.2,ff,0.0001,1,30);
{--> enter halfing, args = .1, .2, ff, 0.1e-3, 1, 30
```

$$aa := 0.1$$

$$bb := 0.2$$

$$\delta := 0.1$$

$$ab := 0.1500000000$$

$$\delta := 0.05000000000$$

$$ff := 0.0000048291$$

$$bb := 0.1500000000$$

$$ab := 0.1250000000$$

$$\delta := 0.02500000000$$

$$ff := -0.0624982310$$

$$aa := 0.1250000000$$

$$ab := 0.1375000000$$

$$\delta := 0.01250000000$$

$$ff := -0.0293720757$$

$$aa := 0.1375000000$$

$$ab := 0.1437500000$$

$$\delta := 0.006250000000$$

$$ff := -0.0142149917$$

$$aa := 0.1437500000$$

$$ab := 0.1468750000$$

$$\delta := 0.003125000000$$

$$ff := -0.0069879271$$

$$aa := 0.1468750000$$

$$ab := 0.1484375000$$

$$\delta := 0.001562500000$$

$$ff := -0.0034622611$$

$$aa := 0.1484375000$$

$$ab := 0.1492187500$$

$$\delta := 0.0007812500000$$

$$ff := -0.0017213940$$

$$aa := 0.1492187500$$

$$ab := 0.1496093750$$

$$\delta := 0.0003906250000$$

$$ff := -0.0008564520$$

$$aa := 0.1496093750$$

$$ab := 0.1498046875$$

$$\delta := 0.0001953125000$$

$$ff := -0.0004253538$$

$$aa := 0.1498046875$$

$$ab := 0.1499023438$$

$$\delta := 0.00009765625000$$

$$ff := -0.0002101478$$

```
<-- exit halfing (now at top level) = .1499023438}
```

$$0.1499023438 \qquad (17.9.9)$$

```
> halfing(0.4,0.5,ff,0.0001,1,30);
{--> enter halfing, args = .4, .5, ff, 0.1e-3, 1, 30
```

$$aa := 0.4$$

$$bb := 0.5$$

$$\delta := 0.1$$

$$ab := 0.4500000000$$

$$\delta := 0.05000000000$$

$$ff := 0.3679196153$$

$$bb := 0.4500000000$$

$$ab := 0.4250000000$$

$$\delta := 0.02500000000$$

$$ff := -0.0126405798$$

$$aa := 0.4250000000$$

$$ab := 0.4375000000$$

$$\delta := 0.01250000000$$

$$ff := 0.1185223994$$

$$bb := 0.4375000000$$

$$ab := 0.4312500000$$

$$\delta := 0.006250000000$$

$$ff := 0.0417181162$$

$$bb := 0.4312500000$$

$$ab := 0.4281250000$$

$$\delta := 0.003125000000$$

$$ff := 0.0120865678$$

$$bb := 0.4281250000$$

$$ab := 0.4265625000$$

$$\delta := 0.001562500000$$

$$ff := -0.0008505494$$

$$aa := 0.4265625000$$

$$ab := 0.4273437500$$

$$\delta := 0.0007812500000$$

$$ff := 0.0054698773$$

$$bb := 0.4273437500$$

$$ab := 0.4269531250$$

$$\delta := 0.0003906250000$$

$$ff := 0.0022732358$$

$$bb := 0.4269531250$$

$$ab := 0.4267578125$$

$$\delta := 0.0001953125000$$

$$ff := 0.0007023107$$

$$bb := 0.4267578125$$

$$ab := 0.4266601562$$

$$\delta := 0.00009765625000$$

$$ff := -0.0000763680$$

```
<-- exit halfing (now at top level) = .4266601562}
```

$$0.4266601562 \qquad (17.9.10)$$

*17.32. A legkisebb négyzetek módszere.

*17.33. Simítás.

*17.34. Feladat.

*17.35. Feladat.

*17.36. Feladat.

```
> ;
```

> ;
> ;

> ;
> ;

>

>

>

>

>

>

>

>

>

▶ 18. Fourier–transzformáció

VI. Variációszámítás

▶ 19. Az Euler–Lagrange–egyenletek

VII. Közönséges differenciálegyenletek