ROMBERG INTEGRATION

Background :

for smooth $f(x)$, with $I(f) = \int_a^b f(x)dx$, using series,

- Trapezoidal rule $T_n(f)$ has error expansion

$$I(f) = T_n(f) + c_1h^2 + c_2h^4 + c_3h^6 + \cdots$$

for constants $c_1,\ c_2, \ldots$ independent of h.

- Midpoint rule $M_n(f)$ has error expansion

$$I(f) = M_n(f) + d_1h^2 + d_2h^4 + d_3h^6 + \cdots$$

for constants $d_1,\ d_2, \ldots$ independent of h.

Extrapolation: Richardson extrapolation can be used

- **Romberg Integration**: let $R_{1,1} = T_1(f), h_k = \frac{b-a}{2^{k-1}}$,

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1}+h_{k-1}\sum_{i=1}^{2^{k-2}} f(a+(2i-1)h_k)\right] \equiv T_{2^{k-1}}(f),\ k = 2, 3, \ldots$$

NUMERICAL INTEGRATION CONT.

Extrapolation formulas

$$R_{k,2} = R_{k,1} + (R_{k,1} - R_{k-1,1})/(4-1), \quad k = 2, 3, \ldots$$

$$\vdots$$

$$R_{k,j} = R_{k,j-1} + (R_{k,j-1} - R_{k-1,j-1})/(4^{j-1} - 1), \; k = j, j+1, \ldots$$

- Romberg Integration Table: computed row by row

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$	$O(h^{10})$	$\leftarrow$ Errors
$R_{1,1} = T_1(h)$					
$R_{2,1} = T_2(h)$	$R_{2,2}$				
$R_{3,1} = T_4(h)$	$R_{3,2}$	$R_{3,3}$			
$R_{4,1} = T_8(h)$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
$R_{5,1} = T_{16}(h)$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Note: $R_{k,2} = S_{2^{k-1}}(f)$, Simpsons' rule;
but could also use $R_{k,1} = M_{2^{k-1}}(f)$.

NUMERICAL INTEGRATION CONT.

Romberg Examples :

- $\int_{-.5}^{0} x \ln(x+1)dx = 0.0525698072900205.$

  Romberg Table

.0866434				
.0613018	.0528546			
.0547688	.0525911	.052573503		
.0531206	.0525712	.052569893	.052569836	
.0527076	.0525699	.052569809	.052569808	.052569807

- $\int_{0}^{1} e^{-x^2}dx = 0.746824132812427.$

  Romberg Table

.6839397				
.7313703	.7471804			
.7429841	.7468554	.746833710		
.7458656	.7468261	.746824170	.746824018	
.7465846	.7468243	.746824133	.746824133	.746824133

- $\int_{-1}^{1} \sin(1+e^{3x})dx = 2.50080911033617.$

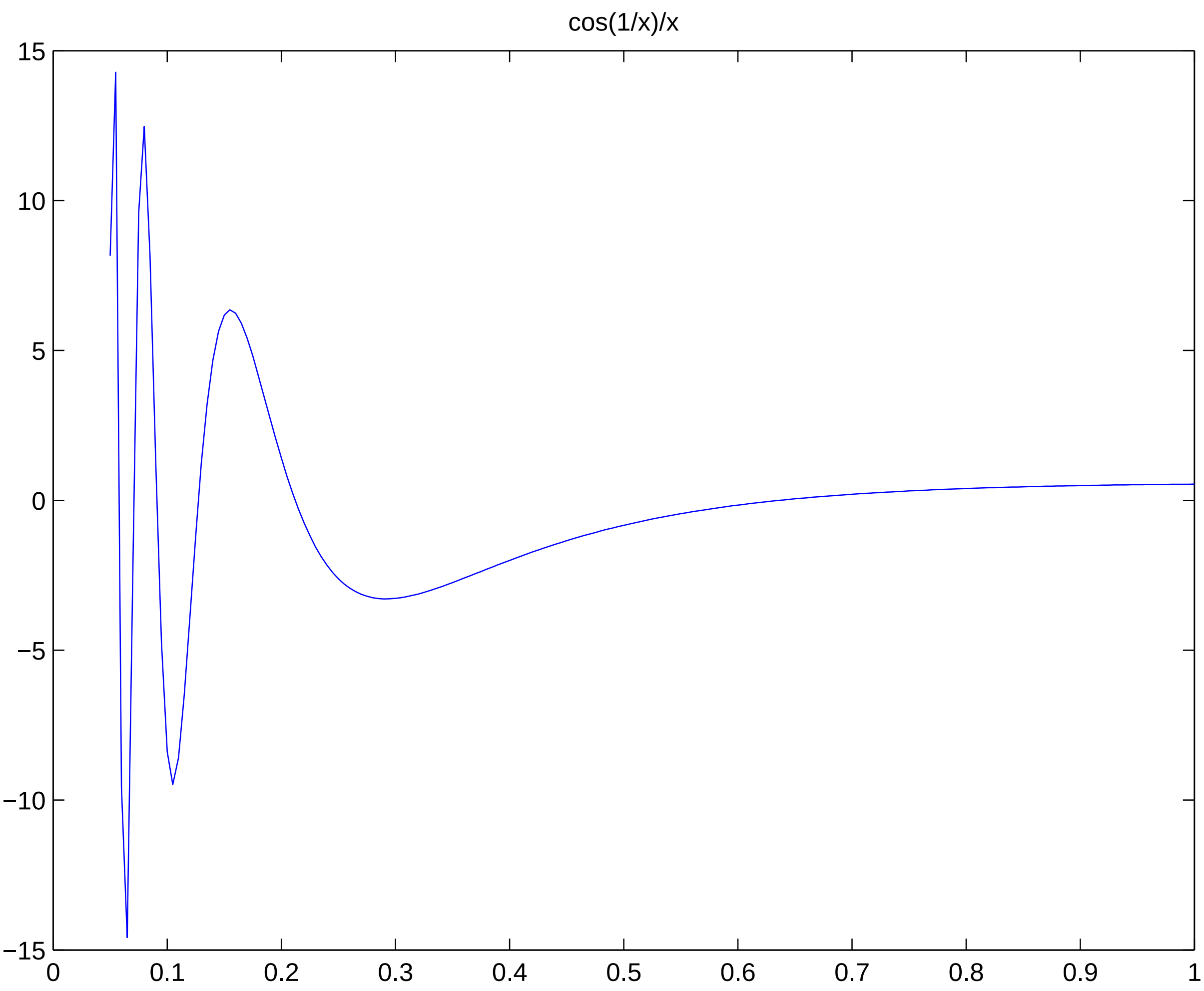
  Romberg Table

2.99424					
3.33859	3.45337				
2.29318	1.94471	1.84414			
2.48454	2.54832	2.58856	2.60038		
2.66899	2.73048	2.74262	2.74507	2.74564	
2.51149	2.45899	2.44089	2.43610	2.43488	2.43458
2.50257	2.49959	2.50230	2.50327	2.50354	2.50360
2.50122	2.50077	2.50085	2.50082	2.50081	2.50081

ADAPTIVE NUMERICAL INTEGRATION

Adaptive Numerical Integration Background

- Motivation: some numerical integration problems involve integrands with varying smoothness in different parts of integration interval $[a, b]$; efficient computation of $I(f)$ requires the total number of integrand evaluations to be minimized.



Romberg method for $\int_{.05}^{1} \cos(1/x)/x dx \approx -.29298$ requires 1025 f values.

- Main Idea: subdivide integration interval $[a, b]$ nonuniformly, adaptively concentrating work in subintervals where integrand has more variation.

ADAPTIVE NUMERICAL INTEGRATION

Local Adaptive Integration, Basic Algorithm

1. input error tolerance ϵ, $[a, b]$, $f(x)$
2. use integration rule $R(f, a, b)$, with error estimate E
3. if $E > \epsilon$, apply algorithm to $f(x)$
   on $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$ with error tolerances $\epsilon/2$;
   and return sum of results for two halves
   otherwise, return $R(f, a, b)$

Produces "tree" of subintervals, results at each tree leaf.

ADAPTIVE NUMERICAL INTEGRATION

Local Adaptive Algorithm as recursive function Int

Int(f, a, b, ϵ)

compute R(f, a, b) and E(f, a, b);

if E(f, a, b) $< \epsilon$, return R(f, a, b),

else, return Int($f, a, \frac{a+b}{2}, \frac{\epsilon}{2}$) + Int($f, \frac{a+b}{2}, b, \frac{\epsilon}{2}$);

end Int.

Matlab Adaptive quadrature with Simpson's rule:

```
function I = adaptr( a, b, f, ep )
%
%  Estimates integral of f(x) over [a,b] with
%   estimated accuracy < ep, using recrusive
%   adaptive integration based on Simpson's rule.
% Example:
%  disp( adaptr( -1,1,@(x)1+sin(exp(3*x)),5e-5 ) )
%
  I = adint( f, a, b, ep, f(a), f((a+b)/2), f(b) );
end
% end adaptr
function I = adint( f, a, b, ep, fa, fm, fb ),
  h = (b-a)/2; m = (a+b)/2;
  fl = f((a+m)/2); L = h*(fa+4*fl+fm)/6; % left
  fr = f((m+b)/2); R = h*(fm+4*fr+fb)/6; % right
  er = ( L + R - h*(fa+4*fm+fb)/3 )/15;  % error
  if abs(er) < ep, I = L + R + er;
  else,
   I = adint( f, a,m,ep/2, fa,fl,fm ) ...
     + adint( f, m,b,ep/2, fm,fr,fb );
  end
% end adint
```

ADAPTIVE NUMERICAL INTEGRATION

Basic Integration Rule with Error Estimate

- Use Trapezoidal Rule on $[a, b]$, $[a, \frac{a+b}{2}]$, $[\frac{a+b}{2}, b]$.

$$|I(f) - T(a, \frac{a+b}{2}) - T(\frac{a+b}{2}, b)| \approx \frac{1}{3}|T(a, b) - T(a, \frac{a+b}{2}) - T(\frac{a+b}{2}, b)|.$$

- Use Simpson's Rule on $[a, b]$, $[a, \frac{a+b}{2}]$, $[\frac{a+b}{2}, b]$.

$$|I(f) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)| \approx \frac{1}{15}|S(a, b) - S(a, \frac{a+b}{2}) - S(\frac{a+b}{2}, b)|.$$

ADAPTIVE NUMERICAL INTEGRATION

NonRecursive Adaptive Integration Algorithm :
uses list of results, and reuses f's

- *input* error tolerance ϵ, $[a, b]$, $f(x)$
- *initialize*
  set $I = 0,\ k = 1,\ a_1 = a,\ h_1 = \frac{b-a}{2}$,
  $fa_1 = f(a),\ fm_1 = f(a + h_1),\ fb_1 = f(b)$,
  $S_1 = \frac{h_1}{3}(fa_1 + 4fm_1 + fb_1)$.
- while $k > 0$, *subdivide last interval from list*
  set $h_k = h_k/2,\ l = f(a_k + h_k),\ r = f(a_k + 3h_k)$;
  $L = \frac{h_k}{3}(fa_k + 4l + fm_k),\ R = \frac{h_k}{3}(fm_k + 4r + fb_k)$;
  if $|S_k - L - R|/15 < 2\epsilon h_k/(b-a)$, *check error*
  set $I = I + L + R + \frac{L+R-S_k}{15},\ k = k-1$; *save or*
  else *udpate list with new subintervals*
  set $S_{k+1} = R,\ a_{k+1} = a_k + 2h_k,\ h_{k+1} = h_k$, *right*
  $fa_{k+1} = fm_k,\ fm_{k+1} = r,\ fb_{k+1} = fb_k$;
  set $S_k = L,\ fb_k = fm_k,\ fm_k = l$; *left*
  set $k = k+1$;
  endif
  endwhile
- *return* $I \approx \int_a^b f(x)dx$.

ADAPTIVE NUMERICAL INTEGRATION

Example $I(f) = \int_{.05}^{1} cos(1/x)/x \; dx$, $\epsilon = 10^{-4}$.

```
k      I           f's / Subinterval list
1  0.98237602     3
         0.05            1
2 -0.34307544     5
         0.05        0.525
        0.525            1
3 -0.33945425     7
         0.05        0.525
        0.525       0.7625
       0.7625            1
2 -0.33940061     9
         0.05        0.525
        0.525       0.7625
3 -0.33920396    11
         0.05        0.525
        0.525      0.64375
      0.64375       0.7625
2 -0.33919885    13
         0.05        0.525
        0.525      0.64375
1 -0.33919213    15
         0.05        0.525
2  0.71815553    17
         0.05       0.2875
       0.2875        0.525
```

. . .

```
  k     I          f's / Subinterval list
  1 -0.53523399   91
         0.05     0.079687
  2 -0.30308683   93
         0.05     0.064844
     0.064844     0.079688
  3 -0.30805689   95
         0.05     0.064844
     0.064844     0.072266
     0.072266     0.079687
  4 -0.30815647   97
         0.05     0.064844
     0.064844     0.072266
     0.072266     0.075977
     0.075977     0.079687
...

         0.05     0.051855
     0.051855     0.052783
  1 -0.29297842  147
         0.05     0.051855
  2 -0.29298195  149
         0.05     0.050928
     0.050928     0.051855
  1 -0.29298206  151
         0.05     0.050928
  0 -0.29298219  153
```

Graphs of $f(x)$ for different numbers of function values