adaptive quadrature

Why adaptive quadrature? Requirements for modern quadrature methods?

- reliability: algorithm yields a result that is correct up to a prescribed tolerance
- efficiency: as little computational effort as possible (sensible work measure: e.g. number of function evalulations)
- applicability: method should work for a large class of integrands. In particular, the algorithm should detect automatically where the integrand is smooth

goal: given f and tolerance $\tau > 0$, find a partition $\{x_0, \ldots, x_N\}$ of $[a, b]$, s.t.

1. $$\left| \int_a^b f - S_{\{x_0,\ldots,x_N\}}(f) \right| \leq \tau$$

2. The number N of subintervals should be small

adaptive quadrature: basic recursive form

trapezoidal rule: $$T([a,b]) := \frac{b-a}{2}\left(f(a)+f(b)\right)$$

adapt(f,a,b,τ)

% computes $\int_a^b f(x)\,dx$ with prescribed tolerance τ

if $\left|\int_a^b f\,dx - T([a,b])\right| \le \tau$ { % desired accuracy reached ☺

  return (T([a,b])) }

else {

% desired accuracy not reached → subdivide $[a,b]$ into $[a,m]$ and $[m,b]$

$m := (a+b)/2$

$I :=$ **adapt**$(f,a,m,\tau/2)$ + **adapt**$(f,m,b,\tau/2)$

return(I) }

adaptive quadrature I

Simpson rule: $S([a,b]) := \frac{b-a}{6}\left(f(a) + 4f(m) + f(b)\right), \qquad m = \frac{a+b}{2}$

adapt(f,a,b,τ)

% computes $\int_a^b f(x)\,dx$ with prescribed accuracy τ

% h_{min} = minimal interval length; $\rho \in (0,1)$ safety factor

if $(b-a) \le h_{min}$ **return**$(T([a,b]))$ %forced termination!

if $|S([a,b]) - T([a,b])| \le \rho\tau$ { desired accuracy reached ☺

    return (T([a,b])) }

else {

% desired accuracy not reached $\to$ subdivide $[a,b]$ into $[a,m]$ and $[m,b]$

$m := (a+b)/2$

$I :=$ **adapt**$(f,a,m,\tau/2)$ + **adapt**$(f,m,b,\tau/2)$

return(I) }

adaptive quadrature II (bolder!)

adapt(f,a,b,τ)

% computes $\int_a^b f(x)\,dx$ with prescribed accuracy τ
% h_{min} = minimal interval length; $\rho \in (0,1)$ safety factor
if $(b-a) \le h_{min}$ **return**$(S([a,b]))$ %forced termination!

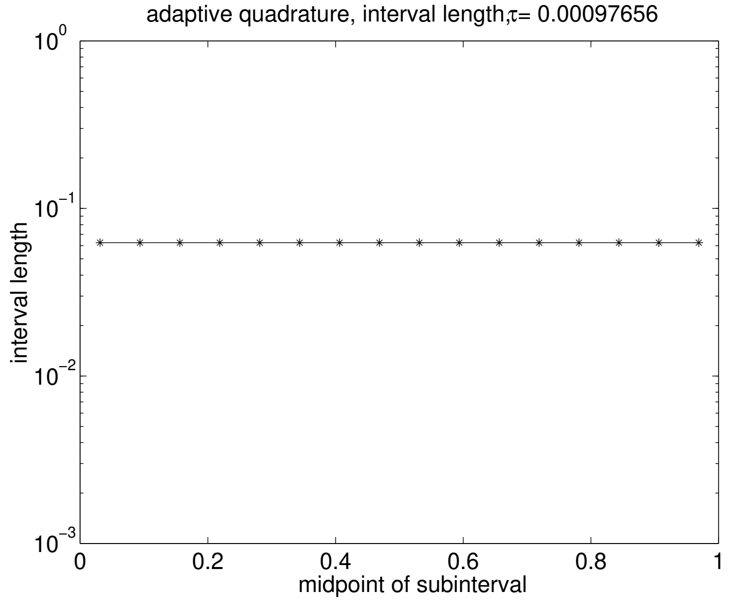
if $|S([a,b]) - T([a,b])| \le \rho\tau$ { desired accuracy reached ☺

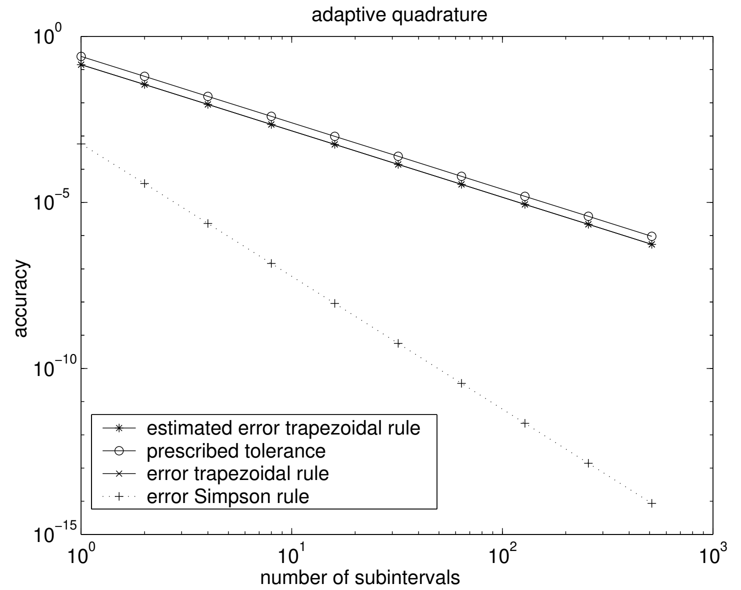
    return (S([a,b])) }

else {

% desired accuracy not reached → subdivide $[a,b]$ into $[a,m]$ and $[m,b]$
$m := (a+b)/2$
$I :=$ **adapt**$(f,a,m,\tau/2)$ + **adapt**$(f,m,b,\tau/2)$
return(I) }

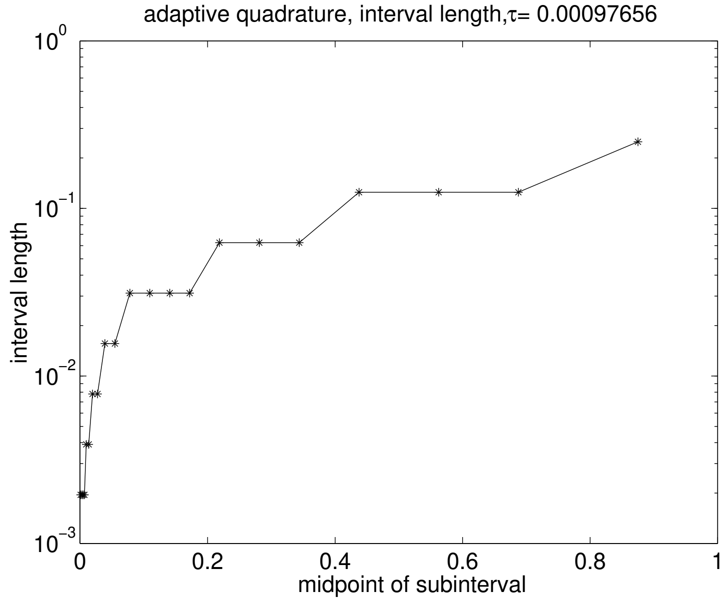
adaptive quadrature for $\int_0^1 e^x \, dx$

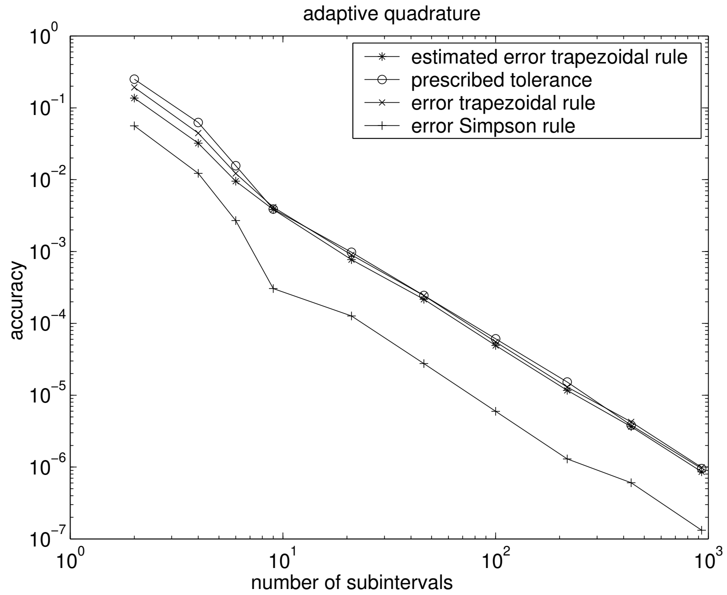




prescribed tolerance τ	# intervals	estimated error	error trapezoidal rule	error Simpson rule
4^{-1}	1	1.4028e-01	1.4086e-01	5.7932e-04
4^{-2}	2	3.5612e-02	3.5649e-02	3.7013e-05
4^{-3}	4	8.9377e-03	8.9401e-03	2.3262e-06
4^{-4}	8	2.2366e-03	2.2368e-03	1.4559e-07
4^{-5}	16	5.5929e-04	5.5930e-04	9.1027e-09
4^{-6}	32	1.3983e-04	1.3983e-04	5.6897e-10
4^{-7}	64	3.4958e-05	3.4958e-05	3.5560e-11
4^{-8}	128	8.7396e-06	8.7396e-06	2.2227e-12
4^{-9}	256	2.1849e-06	2.1849e-06	1.3900e-13
4^{-10}	512	5.4623e-07	5.4623e-07	8.6597e-15

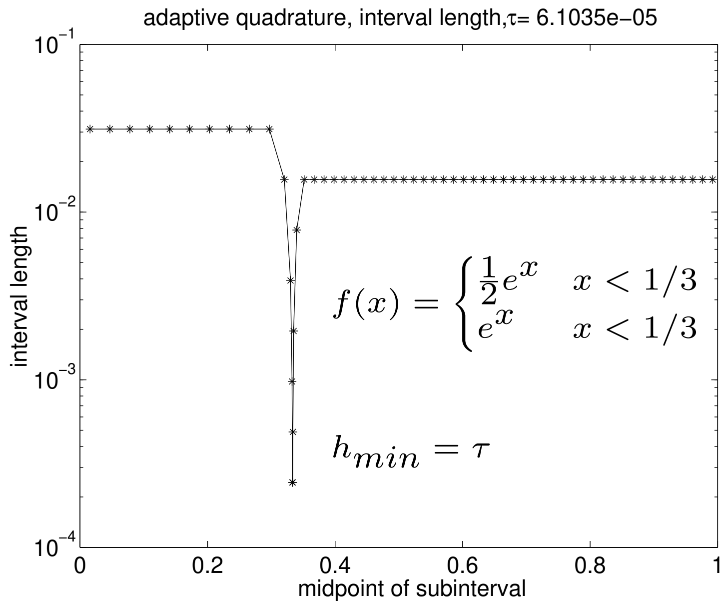
adaptive quadrature for $\int_0^1 x^{0.1}\,dx$

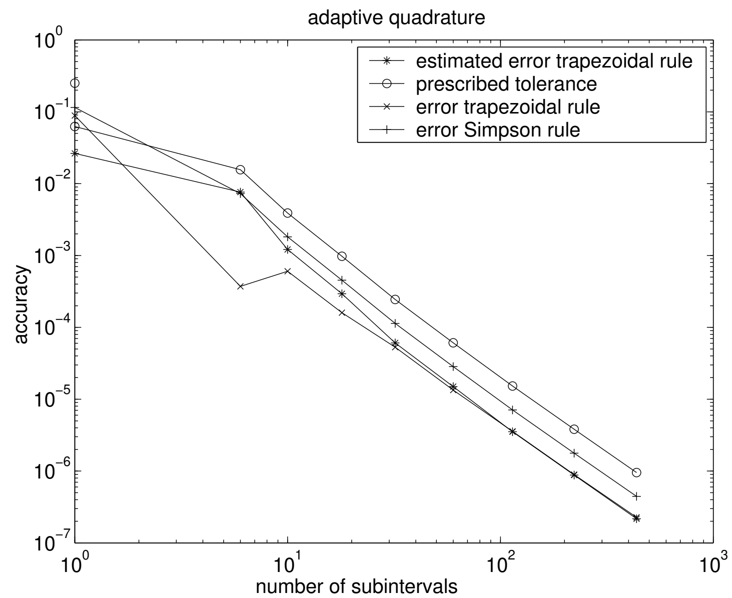




prescribed tolerance τ	# intervals	estimated error	error trapezoidal rule	error Simpson rule
4^{-1}	2	1.3639e-01	1.9257e-01	5.6188e-02
4^{-2}	4	3.2188e-02	4.4444e-02	1.2256e-02
4^{-3}	6	9.5107e-03	1.2205e-02	2.6942e-03
4^{-4}	9	3.8428e-03	4.1473e-03	3.0448e-04
4^{-5}	21	7.6991e-04	8.9684e-04	1.2693e-04
4^{-6}	46	2.1629e-04	2.4381e-04	2.7521e-05
4^{-7}	100	4.9215e-05	5.5191e-05	5.9760e-06
4^{-8}	217	1.1682e-05	1.2981e-05	1.2999e-06
4^{-9}	434	3.6523e-06	4.2586e-06	6.0630e-07
4^{-10}	930	8.5841e-07	9.9036e-07	1.3195e-07

integrand piecewise smooth, jump at $x = 1/3$





prescribed tolerance τ	# intervals	estimated error	error trapezoidal rule	error Simpson rule
4^{-1}	1	2.6387e-02	8.8665e-02	1.1505e-01
4^{-2}	1	2.6387e-02	8.8665e-02	1.1505e-01
4^{-3}	6	7.6415e-03	3.7153e-04	7.2700e-03
4^{-4}	10	1.2147e-03	6.0281e-04	1.8175e-03
4^{-5}	18	2.9397e-04	1.6035e-04	4.5432e-04
4^{-6}	32	6.0648e-05	5.2929e-05	1.1358e-04
4^{-7}	60	1.4965e-05	1.3429e-05	2.8394e-05
4^{-8}	114	3.5327e-06	3.5658e-06	7.0985e-06
4^{-9}	222	8.7993e-07	8.9469e-07	1.7746e-06
4^{-10}	436	2.1669e-07	2.2696e-07	4.4365e-07