General Runge-Kutta Method

- Runge-Kutta family is described by the following equation in the n^{th} step (n = 0, 1, 2, …)
- To specify a particular method, one needs to provide the integer **s** (the number of stages), and the coefficients $\mathbf{a}_{ij}$ (for $1 \le j < i \le s$), $\mathbf{b}_i$ (for i = 1, 2, ..., s) and $\mathbf{c}_i$ (for i = 2, 3, ..., s)
- The coefficients are presented in a **Butcher Tableau**

$$\boxed{y_{n+1} = y_n + h\sum_{i=1}^{s} b_i\, k_i}$$

Uniform partitioning

$$
\begin{aligned}
k_1 &= f(x_n, y_n) \\
k_2 &= f(x_n + c_2 h, y_n + h(a_{21}k_1)) \\
k_3 &= f(x_n + c_3 h, y_n + h(a_{31}k_1 + a_{32}k_2)) \\
&\;\;\vdots \\
k_s &= f(x_n + c_s h, y_n + h(a_{s1}k_1 + a_{s2}k_2 + \cdot + a_{s,s-1}k_{s-1}))
\end{aligned}
$$

$c_1 = 0$, $a_{11} = 0$

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Consistency condition

$$\sum_{j=1}^{i-1} a_{ij} = c_j \text{ for } i = 2, \cdots, s$$

Sweep spot

p	1	2	3	4	5	6	7	8
$\min s$	1	2	3	4	6	7	9	11

order of error $O(h^p)$ vs. number of stages

For p>4, number of stages is <u>more than p</u>.

General Runge-Kutta Method

- Midpoint method (**s = 2**)

Two stages: $O(h^2)$

0		
1/2	1/2	
	0	1

$$\begin{cases} y_{n+1} = y_n + h(1\,k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + 0.5\,h, y_n + h\,(0.5\,k_1)) \end{cases}$$

- Runge-Kutta 4th-order (RK4) (**s = 4**)

Four stages: $O(h^4)$

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

$$\begin{cases} y_{n+1} = y_n + h(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + 0.5\,h, y_n + h\,(0.5\,k_1)) \\ k_3 = f(x_n + 0.5\,h, y_n + h\,(0.5\,k_2)) \\ k_4 = f(x_n + 1\,h, y_n + h\,(1\,k_3)) \end{cases}$$

- Cash-Karp method (**s = 6**)

0						
1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	−9/10	6/5			
1	−11/54	5/2	−70/27	35/27		
7/8	1631/55296	175/512	575/13824	44275/110592	253/4096	
	37/378	0	250/621	125/594	0	512/1771
	2825/27648	0	18575/48384	13525/55296	277/14336	1/4

Six stages: $O(h^5)$

Six stages: $O(h^4)$

Comparison of these two results gives an estimate for the error

General Runge-Kutta Method

- Fehlberg (**s = 6**)
  - a.k.a: **RKF45**
  - Has $O(h^5)$ and $O(h^4)$

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3680/513	−845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
	16/135	0	6656/12825	28561/56430	−9/50	2/55
	25/216	0	1408/2565	2197/4104	−1/5	0

Six stages: $O(h^5)$

Six stages: $O(h^4)$

```
public Sequence fehlbergSequence(double x1) {
	return n -> {
		if(n==0)
			return y0 ;
		else {
			double x = x0 ;
			double y = y0 ;
			double h = (x1-x0)/(double)n ;
			double k1=0.0, k2=0.0, k3=0.0, k4=0.0, k5=0.0, k6=0.0 ;
			for(int i=0; i<n; i++) {
				k1 = func.value(x, y) ;
				k2 = func.value(x+1.0/4.0*h, y+h*(1.0/4.0*k1)) ;
				k3 = func.value(x+3.0/8.0*h, y+h*(3.0/32.0*k1+9.0/32.0*k2)) ;
				k4 = func.value(x+12.0/13.0*h, y+h*(1932.0/2197.0*k1-7200.0/2197.0*k2+7296.0/2197.0*k3)) ;
				k5 = func.value(x+1.0*h, y+h*(439.0/216.0*k1-8.0*k2+3680.0/513.0*k3-845.0/4104.0*k4)) ;
				k6 = func.value(x+1.0/2.0*h, y+h*(-8.0/27.0*k1+2.0*k2-3544.0/2565.0*k3+1859.0/4104.0*k4-11.0/40.0*k5)) ;
				// 5th-order
				y = y + h*(16.0/135.0*k1 + 0.0*k2 + 6656.0/12825.0*k3 +
						28561.0/56430.0*k4 - 9.0/50.0*k5 + 2.0/55.0*k6) ;
				x = x + h ;
			}
			return y ;
		}
	} ;
}
```

Implementing $O(h^5)$