Parallel two-step Runge-Kutta methods

Helmut Podhaisky, podhaisky@mathematik.uni-halle.de

Martin–Luther–University Halle–Wittenberg, Germany

20 July 2006

Outline

Explicit Runge–Kutta methods for $y' = f(y)$

classical Runge–Kutta method (1904)

$$
\begin{aligned}
Y_{m1} &= y_{m-1} \\
Y_{m2} &= y_{m-1} + \frac{1}{2} h f(Y_{m1}) \\
Y_{m3} &= y_{m-1} + \frac{1}{2} h f(Y_{m2}) \\
Y_{m4} &= y_{m-1} + h f(Y_{m3}) \\
y_m &= y_{m-1} + \frac{h}{6} f(Y_{m1}) + \frac{h}{3} f(Y_{m2}) + \frac{h}{3} f(Y_{m3}) + \frac{h}{6} f(Y_{m4})
\end{aligned}
$$

Explicit Runge–Kutta methods for $y' = f(y)$

classical Runge–Kutta method (1904)

$$
\begin{aligned}
Y_{m1} &= y_{m-1} \\
Y_{m2} &= y_{m-1} + \frac{1}{2} h f(Y_{m1}) \\
Y_{m3} &= y_{m-1} + \frac{1}{2} h f(Y_{m2}) \\
Y_{m4} &= y_{m-1} + h f(Y_{m3}) \\
y_m &= y_{m-1} + \frac{h}{6} f(Y_{m1}) + \frac{h}{3} f(Y_{m2}) + \frac{h}{3} f(Y_{m3}) + \frac{h}{6} f(Y_{m4})
\end{aligned}
$$

Butcher scheme

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

- the computation is sequential

Explicit Runge–Kutta methods for $y' = f(y)$

classical Runge–Kutta method (1904)

$$
\begin{aligned}
Y_{m1} &= y_{m-1} \\
Y_{m2} &= y_{m-1} + \frac{1}{2} h f(Y_{m1}) \\
Y_{m3} &= y_{m-1} + \frac{1}{2} h f(Y_{m2}) \\
Y_{m4} &= y_{m-1} + h f(Y_{m3}) \\
y_m &= y_{m-1} + \frac{h}{6} f(Y_{m1}) + \frac{h}{3} f(Y_{m2}) + \frac{h}{3} f(Y_{m3}) + \frac{h}{6} f(Y_{m4})
\end{aligned}
$$

Butcher scheme

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

- the computation is sequential
- Y_{mk} is *not* an accurate approximation to $y(t_{m-1} + c_k h)$ leading to complex order conditions
- linear stability can be studied easily
- stepsize changes are trivial

Explicit Runge–Kutta methods, Dormand/Prince (1980)

order 5, much better accuracy than RK4

0						
$\frac{1}{5}$	$\frac{1}{5}$					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$			
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$		
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$

- widely used, Matlab: ode45
- not parallelizable

Bashforth and Adams (1883)

explicit multistep methods, e.g. order 4:

$$y^{[m]} = y^{[m-1]} + \frac{h}{24}\left(55f_{m-1} - 59f_{m-2} + 37f_{m-3} - 9f_{m-4}\right)$$

- the computation in sequential
- the order conditions are rather simple, just interpolation
- linear stability is more difficult to analyze
- implementing a robust variable stepsize–variable order code is quite a challenge, Matlab `ode113`

Implicit methods

Runge–Kutta methods

- low stage order $\Longrightarrow$ order reduction
- high stage order $\Longrightarrow$ coupled iterations (and a good chance for some parallelization, bottleneck: LU-decomposition of the Jacobian)

Multistep methods

- second Dahlquist barrier: an A-stable method cannot have order $p > 2$
- not parallelizable

Questions:

- What limitations can be overcome with more general methods? In particular: what about parallelism?
- Is it possible to be competitive in applications?

Explicit parallel peer methods

Idea: re-use stages from the last step

$$Y_m = \begin{bmatrix} Y_{m1} \\ Y_{m2} \\ \vdots \\ Y_{ms} \end{bmatrix}, \quad Y_{mk} = y(t_m + c_k h) + O(h^{p+1})$$

Scheme: A and B full $s \times s$ matrices

$$Y_m = hAf(Y_{m-1}) + BY_{m-1}$$

- the computation is completely parallel
- the order conditions are simple as long as $p \leq s - 1$ (interpolation, $B = B(A)$)
- we want zero-stability with no parasitic roots, $\sigma(B) = \{1, 0\}$
- difficult: optimizing stability and variable stepsize behaviour

order conditions for $Y_m = hAf(Y_{m-1}) + BY_{m-1}$

Taylor series expansion

With $z = h\frac{d}{dt}$, we have $Y_m = \exp(cz)y(t_m)$ and

$$\exp(cz) = Az\exp((c-1)z) + B\exp((c-1)z) + O(z^{p+1})$$

which can be satisfied for B easily if $p = s - 1$.

order conditions for $Y_m = hAf(Y_{m-1}) + BY_{m-1}$

Taylor series expansion

With $z = h\frac{d}{dt}$, we have $Y_m = \exp(cz)y(t_m)$ and

$$\exp(cz) = Az\exp((c-1)z) + B\exp((c-1)z) + O(z^{p+1})$$

which can be satisfied for B easily if $p = s - 1$.
ordered by powers of z

$$\exp(cz) \sim \begin{bmatrix} 1 & c_1 & \frac{c_1^2}{2} & \cdots & \frac{c_1^{s-1}}{s-1!} \\ 1 & c_2 & \frac{c_2^2}{2} & \cdots & \frac{c_2^{s-1}}{s-1!} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & c_s & \frac{c_s^2}{2} & \cdots & \frac{c_s^{s-1}}{s-1!} \end{bmatrix}$$

order conditions for $Y_m = hAf(Y_{m-1}) + BY_{m-1}$

Taylor series expansion

With $z = h\frac{d}{dt}$, we have $Y_m = \exp(cz)y(t_m)$ and

$$\exp(cz) = Az\exp((c-1)z) + B\exp((c-1)z) + O(z^{p+1})$$

which can be satisfied for B easily if $p = s - 1$.
ordered by powers of z

$$\exp((c-1)z) \sim \begin{bmatrix} 1 & c_1 - 1 & \frac{(c-1)_1^2}{2} & \dots & \frac{(c_1-1)^{s-1}}{s-1!} \\ 1 & c_2 - 1 & \frac{(c-1)_2^2}{2} & \dots & \frac{(c_2-1)^{s-1}}{s-1!} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & c_s - 1 & \frac{(c-1)_s^2}{2} & \dots & \frac{(c_s-1)^{s-1}}{s-1!} \end{bmatrix}$$

order conditions for $Y_m = hAf(Y_{m-1}) + BY_{m-1}$

Taylor series expansion

With $z = h\frac{d}{dt}$, we have $Y_m = \exp(cz)y(t_m)$ and

$$\exp(cz) = Az\exp((c-1)z) + B\exp((c-1)z) + O(z^{p+1})$$

which can be satisfied for B easily if $p = s - 1$.
ordered by powers of z

$$z\exp((c-1)z) \sim \begin{bmatrix} 0 & 1 & c_1 - 1 & \dots & \frac{(c_1-1)^{s-2}}{s-2!} \\ 0 & 1 & c_2 - 1 & \dots & \frac{(c_2-1)^{s-2}}{s-2!} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 1 & c_s - 1 & \dots & \frac{(c_s-1)^{s-2}}{s-2!} \end{bmatrix}$$

Explicit parallel peer methods, linear stability

Let $y' = \lambda y$ and $z = h\lambda$.
We obtain

$$Y_m = M(z)Y_{m-1} = (B + zA)Y_{m-1}$$

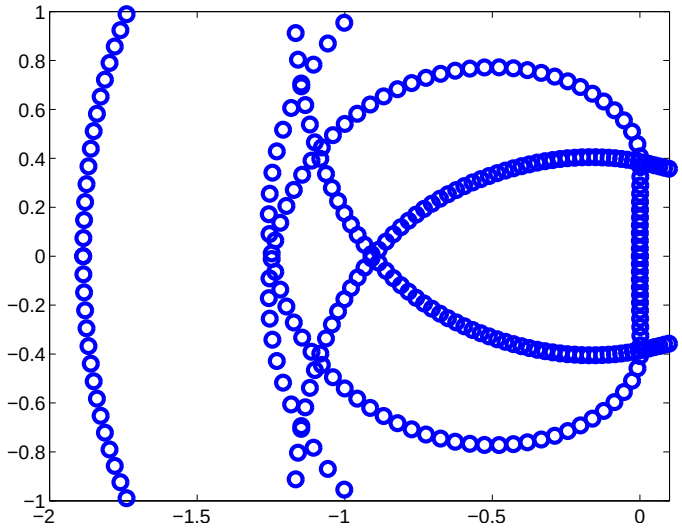
- $M(z)$ is called stability or amplification matrix.
- zero stability: $B = M(0)$ must be power-bounded
- stability domain: $\{z : M(z) \quad \text{is power-bounded}\}$

Explicit parallel peer methods, linear stability

Let $y' = \lambda y$ and $z = h\lambda$.
We obtain

$$Y_m = M(z)Y_{m-1} = (B + zA)Y_{m-1}$$

- $M(z)$ is called stability or amplification matrix.
- zero stability: $B = M(0)$ must be power-bounded
- stability domain: $\{z : M(z) \quad \text{is power-bounded}\}$
- example $s = 6$



compute boundary:
$M(z)v = \exp(\phi i)v$
leads to
generalized EV problem:
$zA = (\exp(\phi i)I - B)v$

Explicit parallel peer methods, optimization

inner loop: linear conditions

- order $p \leq s - 1$
- zero-parasitic eigenvalues: Schur-form $Q^T \hat{B} Q$ ist strictly upper triangular,
  $Q = (I + S)(I - S)^{-1}$ Cayley-transform, S – free skew matrix
- linear least squares, e.g. $\|A\|_F$ and $\|B\|_F$

outer loop: nonlinear conditions

- $\rho(M(z))$, shape of the stability domain
- higher order errors $p \geq s$, super-convergence conditions

After all, the method has to be reformulated for variable stepsizes.

Explicit parallel peer methods, optimization

inner loop: linear conditions

- order $p \leq s - 1$
- zero-parasitic eigenvalues: Schur-form $Q^T \hat{B} Q$ ist strictly upper triangular,
  $Q = (I + S)(I - S)^{-1}$ Cayley-transform, S – free skew matrix
- linear least squares, e.g. $\|A\|_F$ and $\|B\|_F$

outer loop: nonlinear conditions

- $\rho(M(z))$, shape of the stability domain
- higher order errors $p \geq s$, super-convergence conditions

After all, the method has to be reformulated for variable stepsizes.

$\Rightarrow$ work in progress

implicit parallel peer methods

Implicit parallel peer methods

parallel+implicit

Y_{mi} depends on $hf(Y_{mi})$

$$Y_m = Ghf(Y_m) + hAf(Y_{m-1}) + BY_{m-1}, \quad G = \mathrm{diag}(g_1, \ldots, g_s)$$

stiff accuracy

$M(z) = (I - zG)^{-1}(zA + B)$ vanishes for $z \to -\infty$ iff $A = 0$

example: method PP3, 3 stages, order 2

L-stable, stiffly accurate

$c = (-0.1, 0.7, 1)$, $G = \mathrm{diag}(1.7, 0.2670, 0.4454)$

Implicit parallel peer methods

parallel+implicit

Y_{mi} depends on $hf(Y_{mi})$

$$Y_m = Ghf(Y_m) + hAf(Y_{m-1}) + BY_{m-1}, \quad G = \mathrm{diag}(g_1, \ldots, g_s)$$

stiff accuracy

$M(z) = (I - zG)^{-1}(zA + B)$ vanishes for $z \to -\infty$ iff $A = 0$

example: method PP3, 3 stages, order 2

L-stable, stiffly accurate

$c = (-0.1, 0.7, 1)$, $G = \mathrm{diag}(1.7, 0.2670, 0.4454)$

parasitic roots: $B = \begin{bmatrix} -0.21 & 6.79 & -5.57 \\ 0.27 & -2.46 & 3.18 \\ 0.31 & -2.99 & 3.68 \end{bmatrix}$,

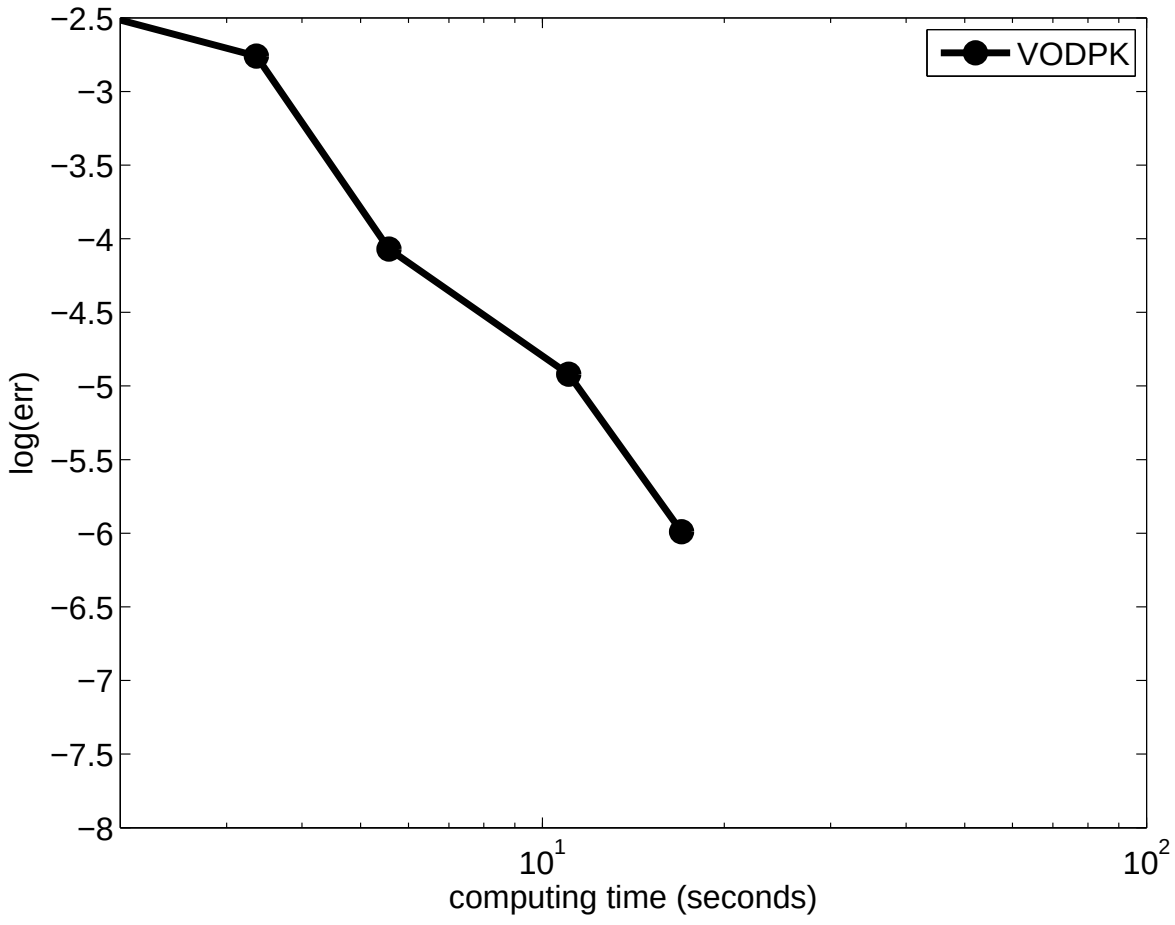
$\sigma(B) = \{1, -0.0005 \pm 0.2202i\}$ (for constant stepsize)

Implementation

- one Newton iteration per stage, linearly implicit method
- Krylov subspace $\mathcal{K} := \text{span}\{b, Tb, \ldots, T^{\kappa-1}\}$, with $T = f_y$
  Find $\bar{x}$ for $(I - \gamma h T)x = b$ with $\bar{x} \in \mathcal{K}$ and $(A\bar{x} - b) \perp \mathcal{K}$.
- Fortran 90 using OpenMP for loop parallelization
- experiments on a SunFire (shared memory)
- spatial discretizations with finite differences and finite volumes, respectively

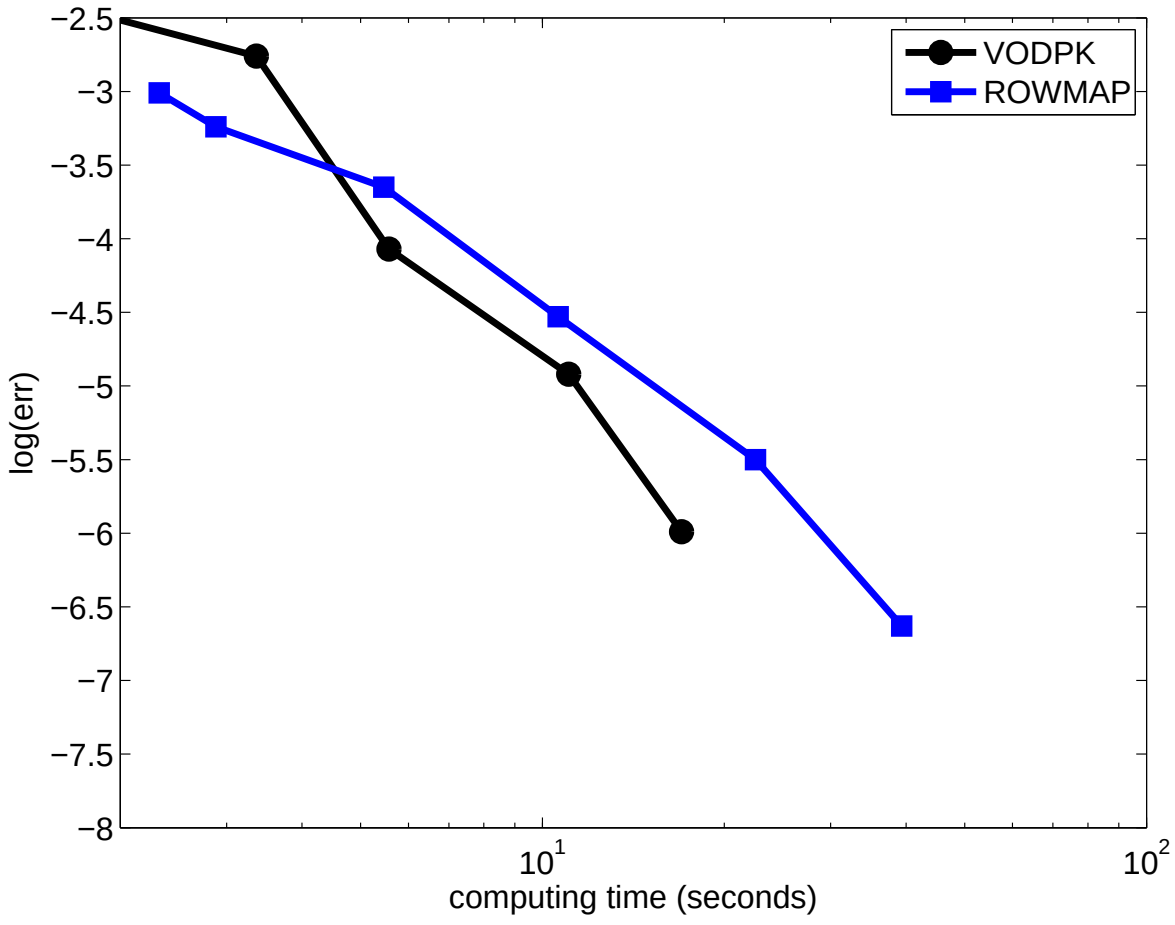
Example 1: Diffu, $u_t = u_{xx} + u_{yy} + f(t, x, y)$

with $0 \leq t, x, y \leq 1$ and exact solution
$\sin(\pi x) \sin(\pi y)(1 + 4xy \sin t)$



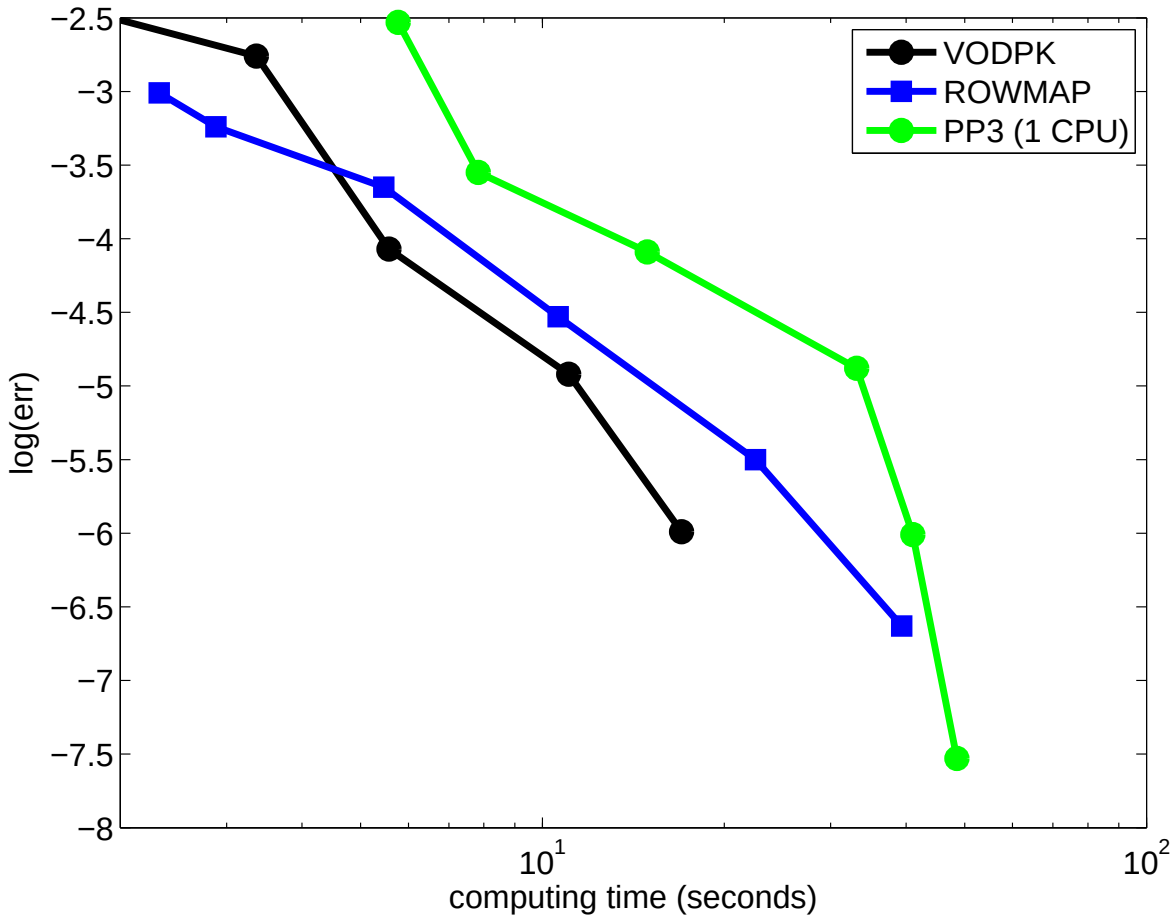
Example 1: Diffu, $u_t = u_{xx} + u_{yy} + f(t, x, y)$

with $0 \leq t, x, y \leq 1$ and exact solution
$\sin(\pi x) \sin(\pi y)(1 + 4xy \sin t)$



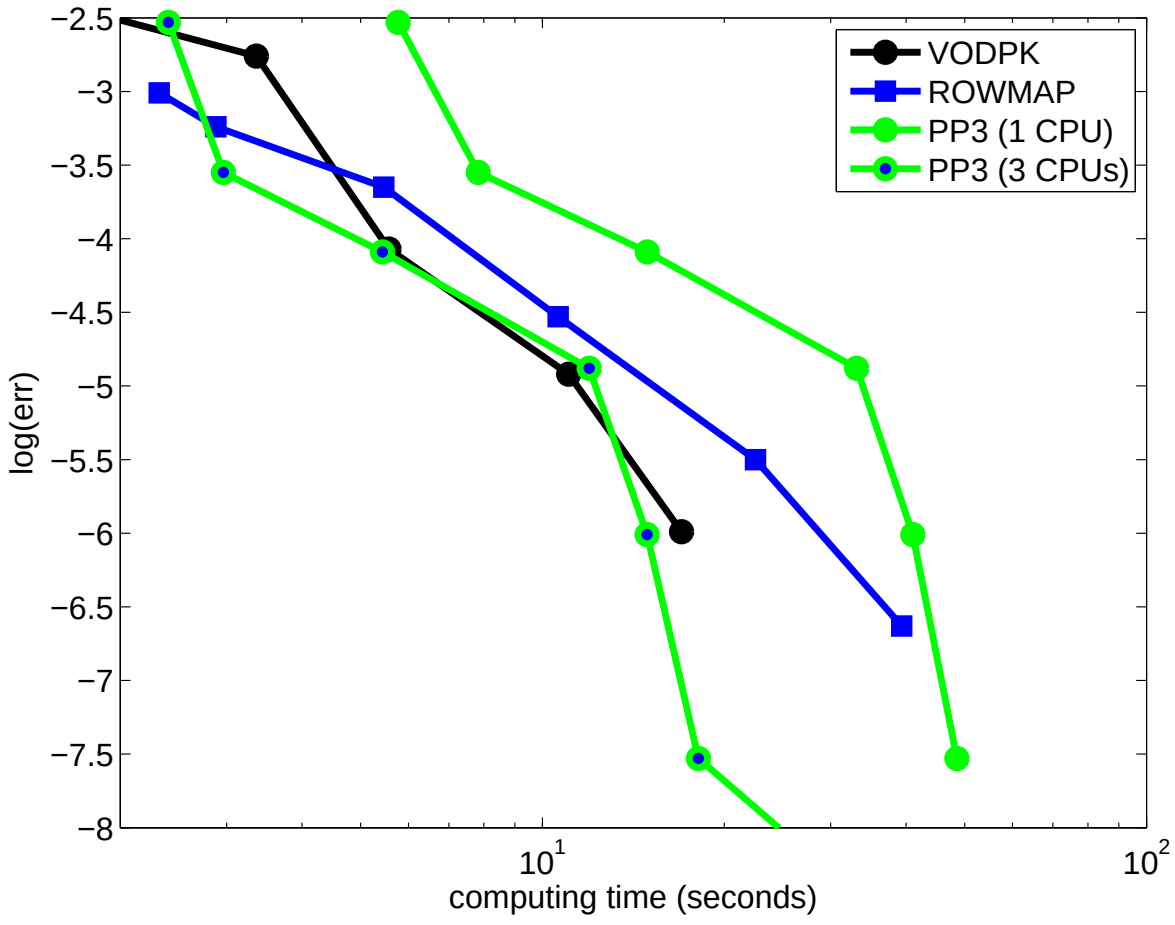
Example 1: Diffu, $u_t = u_{xx} + u_{yy} + f(t, x, y)$

with $0 \leq t, x, y \leq 1$ and exact solution
$\sin(\pi x)\sin(\pi y)(1 + 4xy \sin t)$



Example 1: Diffu, $u_t = u_{xx} + u_{yy} + f(t, x, y)$

with $0 \leq t, x, y \leq 1$ and exact solution
$\sin(\pi x)\sin(\pi y)(1 + 4xy \sin t)$



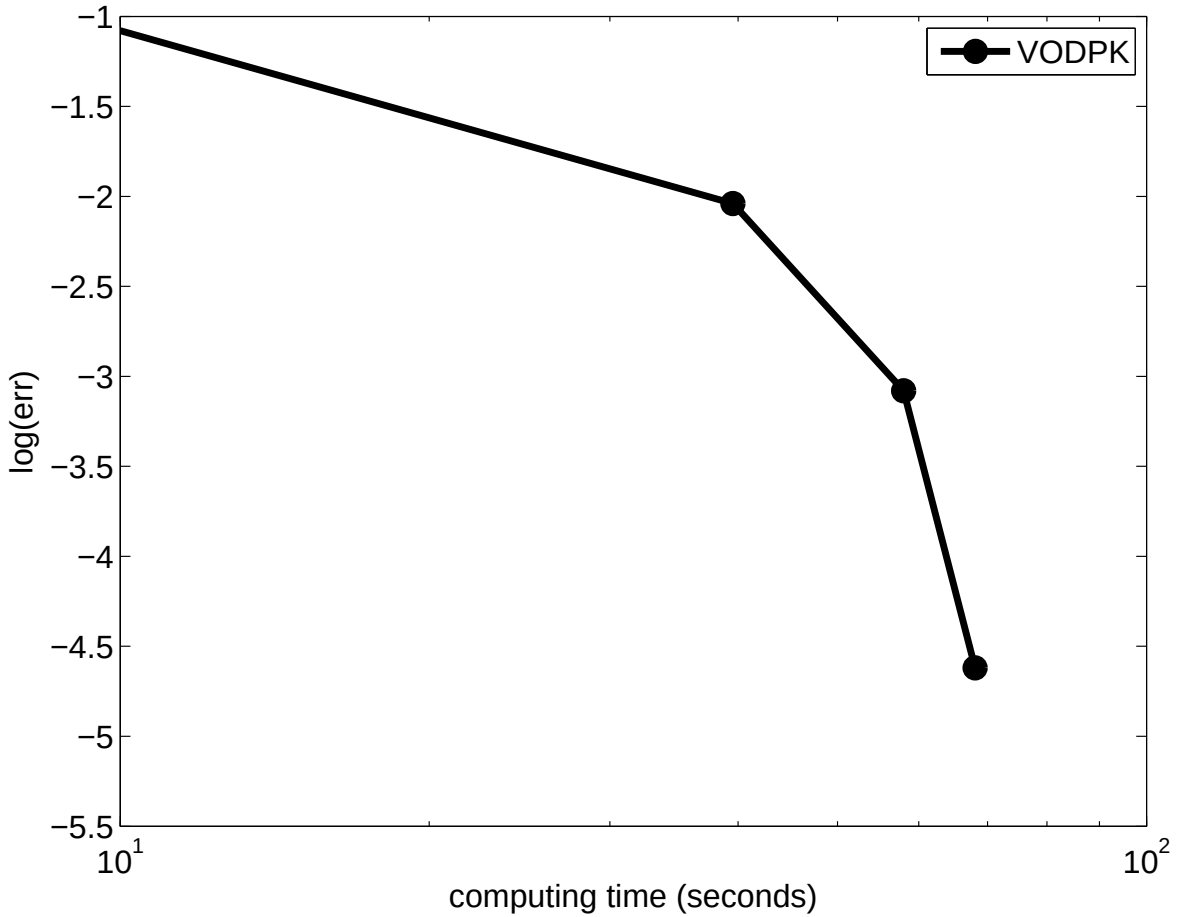
Example 2: Radiation-Diffusion

Mousseau, Knoll & Rider (2000), Verwer, Hundsdorfer (2003)

$$E_t = \nabla \cdot (D_r \nabla E) + \sigma(T^4 - E), \quad D_r = (3\sigma + (1/E)|\partial E/\partial x|)^{-1},$$

$$T_t = \nabla \cdot (D_t \nabla T) - \sigma(T^4 - E), \quad D_t = kT^{5/2}, \quad k = 0 \text{ or } k = 0.1$$

strongly nonlinear, conservative second-order discretization



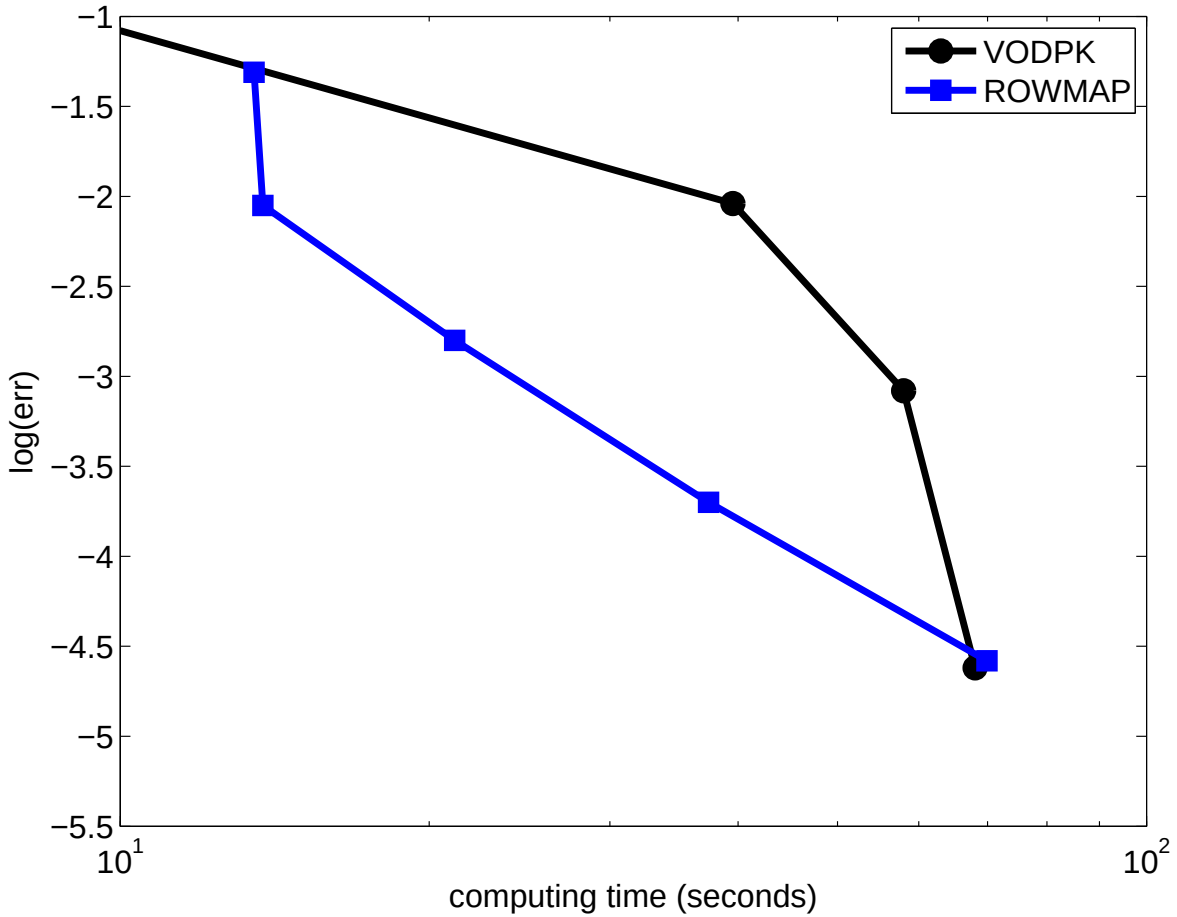
Example 2: Radiation-Diffusion

Mousseau, Knoll & Rider (2000), Verwer, Hundsdorfer (2003)

$$E_t = \nabla \cdot (D_r \nabla E) + \sigma(T^4 - E), \quad D_r = (3\sigma + (1/E)|\partial E/\partial x|)^{-1},$$

$$T_t = \nabla \cdot (D_t \nabla T) - \sigma(T^4 - E), \quad D_t = kT^{5/2}, \quad k = 0 \text{ or } k = 0.1$$

strongly nonlinear, conservative second-order discretization



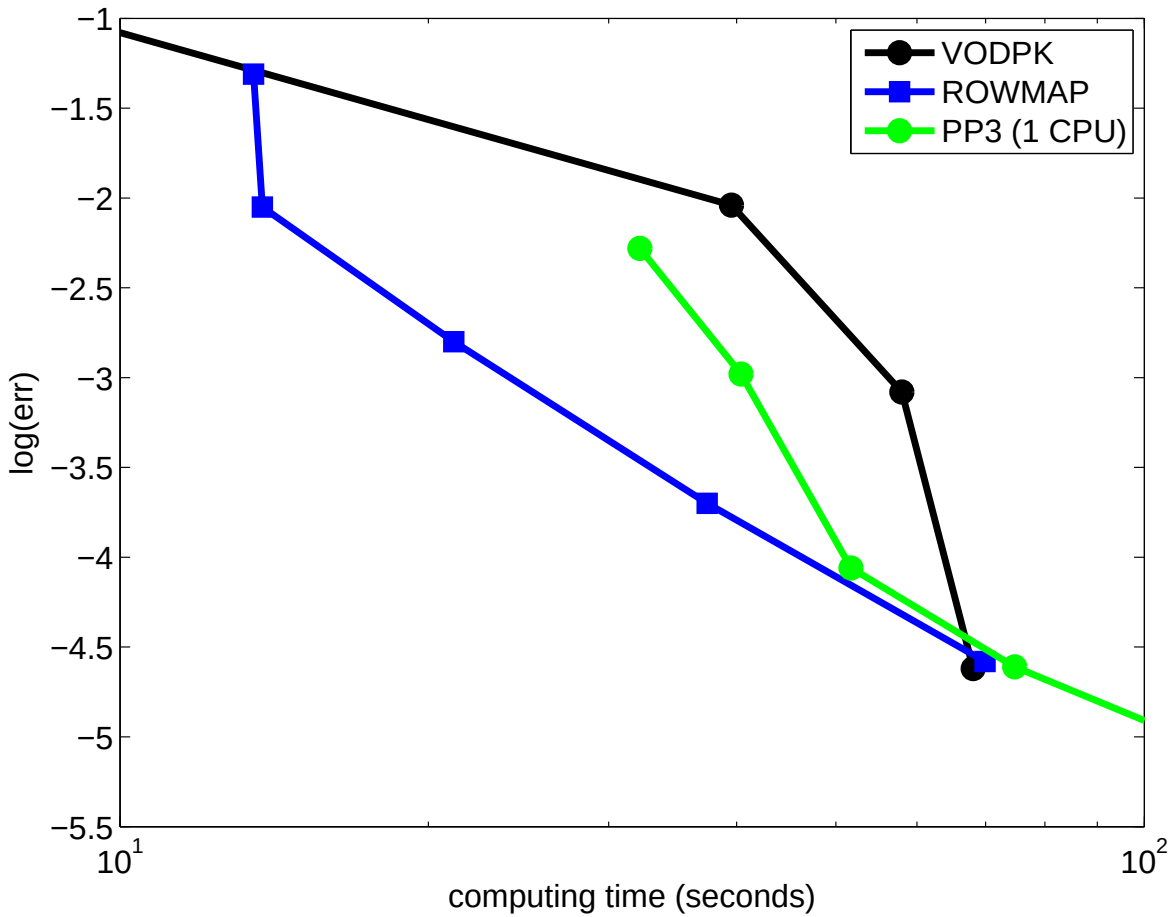
Example 2: Radiation-Diffusion

Mousseau, Knoll & Rider (2000), Verwer, Hundsdorfer (2003)

$$E_t = \nabla \cdot (D_r \nabla E) + \sigma(T^4 - E), \quad D_r = (3\sigma + (1/E)|\partial E/\partial x|)^{-1},$$

$$T_t = \nabla \cdot (D_t \nabla T) - \sigma(T^4 - E), \quad D_t = kT^{5/2}, \quad k = 0 \text{ or } k = 0.1$$

strongly nonlinear, conservative second-order discretization



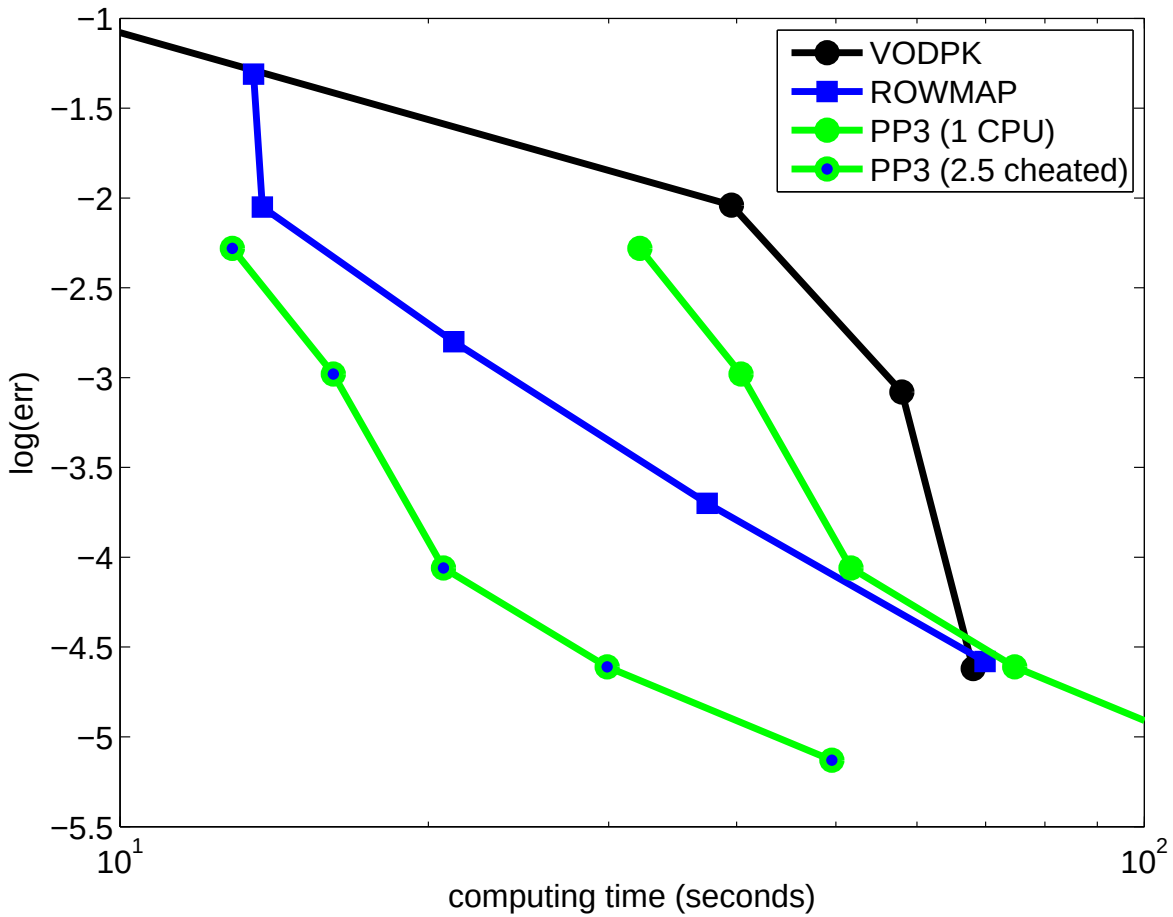
Example 2: Radiation-Diffusion

Mousseau, Knoll & Rider (2000), Verwer, Hundsdorfer (2003)

$$E_t = \nabla \cdot (D_r \nabla E) + \sigma(T^4 - E), \quad D_r = (3\sigma + (1/E)|\partial E/\partial x|)^{-1},$$

$$T_t = \nabla \cdot (D_t \nabla T) - \sigma(T^4 - E), \quad D_t = kT^{5/2}, \quad k = 0 \text{ or } k = 0.1$$

strongly nonlinear, conservative second-order discretization



Summary

- classical methods cannot be parallelized, we need multi-stage, multi-step methods
- methods with parallel stages scale can be implemented easily on shared memory computers (with, say, 2 to 8 CPUs/cores)
- order conditions can be satisfied by interpolation, however, super-convergence is more difficult to achieve
- the main difficulty lies in finding robust methods with small coefficients and moderate error constants
- *sequential* general linear methods are worth to be studied, too

  John Butcher (NZ), Adrian Hill (UK),
  Caren Tischendorf (Matheon/Colone),
  group in Halle, . . .