Splitting with Exponential Time Differencing Schemes For Reaction-Diffusion Systems

Bruce A. Wade

Department of Mathematics

bruce.wade@louisiana.edu

JMM2020

Collaborators

Dr. Emmanuel Asante-Asamani
Department of Mathematics
Hunter College

THE CITY UNIVERSITY OF NEW YORK

Dr. Andreas Kleefeld
Jülich Supercomputing Centre
Germany

JÜLICH Forschungszentrum

Nonlinear Reaction-Diffusion Systems

- Nonlinear parabolic initial-boundary value problem:

$$\begin{aligned} u_t + Au &= f(t,u) \qquad \text{in } \Omega, \qquad t \in (0,T\ ), \\ u(\cdot,0) &= u_0 \end{aligned}$$

- Ω : bounded domain in $\mathbb{R}^d$;
  $A = -D\Delta$, D is diagonal & positive definite;
  Boundary conditions are homogeneous Dirichlet, homogeneous Neumann, or periodic;
  f : nonlinear reaction.

Exact dynamics:

$$u(t) = E(t)u_0 + \int_0^t E(t-s)\, f(s, u(s))\, ds$$

$$0 < k \le k_0, t_n = nk, 0 \le n \le N.$$

Normalize over one time step t_n to t_{n+1} & use $E(t) = e^{-tA}$:

Capture the single step exact dynamics

$$u(t_{n+1}) = e^{-kA}u(t_n) + k \int_0^1 e^{-kA(1-\tau)} f(t_n + \tau k, u(t_n + \tau k))\, d\tau$$

Options: Approximate the whole integrand (quadrature) or approximate f & integrate exactly.

Single Step Exact Dynamics Exploited

For e^{-kA} we have choices: Rational approximation (Padé or other type), Krylov subspace methods...None appears easy due to $e^{-kA(1-\tau)}$ inside the integral. However, we have found a way to deal with the problem.

Two interesting rational approximations are:

- **(1,1) - Padé Scheme (Crank-Nicolson):**

$$R_{1,1}(-kA) = (I + \frac{1}{2}kA)^{-1}(I - \frac{1}{2}kA)$$

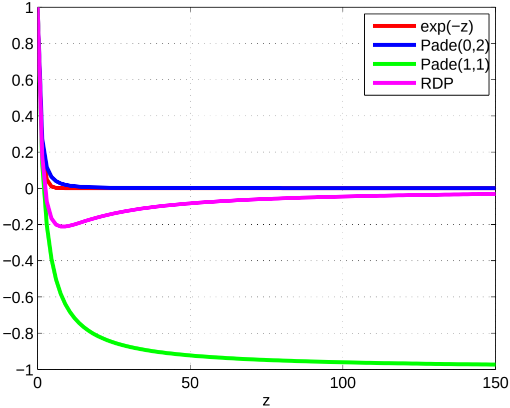
- **Real-Distinct Poles (RDP) Scheme (*L*-stable, Khaliq-Voss):**

$$r(-kA) = \left(I - \frac{5}{12}Ak\right)\left(I + \frac{1}{4}Ak\right)^{-1}\left(I + \frac{1}{3}Ak\right)^{-1}$$

Equivalent: $$= 9(I + \frac{1}{3}kA)^{-1} - 8(I + \frac{1}{4}kA)^{-1}$$

$R(z) \approx e^{-z}$ is A-acceptable if $|R(z)| < 1$ whenever $Re(z) > 0$ and L-acceptable if also $|R(z)| \to 0$ as $Re(z) \to \infty$.

RDP is L-acceptable & has a simple form: $9(1+\frac{1}{3}z)^{-1} - 8(1+\frac{1}{4}z)^{-1} \approx e^{-z}$



Semi-discrete ETD using RDP

Linear appoximation for f, integrate exactly:

$$
\begin{aligned}
u(t_{n+1}) &= e^{-Ak}u(t_n) + A^{-1}(I - e^{-Ak})f(t_n, u(t_n)) \\
&+ k^{-1}A^{-2}(e^{-kA} - I + kA)(f(t_n, u(t_{n+1}) - f(t_n, u(t_n)))
\end{aligned}
$$

& use —

$$
e^{-kA} \approx \left(I - \frac{5}{12}kA\right)\left(I + \frac{1}{4}kA\right)^{-1}\left(I + \frac{1}{3}kA\right)^{-1}
$$

Semi-discrete $\Rightarrow$ fully discrete

Intermediate form —

$$
\begin{aligned}
U_{n+1} &= \left(I - \frac{5}{12}Ak\right)\left(I + \frac{1}{4}Ak\right)^{-1}\left(I + \frac{1}{3}Ak\right)^{-1} U_n \\
&+ \frac{k}{2}\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} f(t_n, U_n) \\
&+ \frac{k}{2}\left(I + \frac{1}{6}kA\right)\left(I + \frac{1}{4}kA\right)^{-1}\left(I + \frac{1}{3}kA\right)^{-1} f(t_{n+1}, U^*_{n+1})
\end{aligned}
$$

Partial fraction forms simplify the computation

$$\left(I - \frac{5}{12}Ak\right)\left(I + \frac{1}{4}Ak\right)^{-1}\left(I + \frac{1}{3}Ak\right)^{-1} = 9\left(I + \frac{1}{3}Ak\right)^{-1} - 8\left(I + \frac{1}{4}Ak\right)^{-1}$$

$$\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} = 4\left(I + \frac{1}{3}Ak\right) - 3\left(I + \frac{1}{4}Ak\right)^{-1}$$

$$\left(I + \frac{Ak}{6}\right)\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} = 2\left(I + \frac{1}{3}Ak\right)^{-1} - \left(I + \frac{1}{4}Ak\right)^{-1}$$

RE Efficiency with linear algebra solvers

ETD-RDP: Efficient implementation

1. Solve for the estimator U^*

$$(I + Ak)U^* = U_n + kf(U_n)$$

2. Solve for U_{n+1} (serial or parallel)

$$\left(I + \frac{1}{3}Ak\right) U_a = 9U_n + 2kf(U_n) + kf(U^*)$$
$$\left(I + \frac{1}{4}Ak\right) U_b = -8U_n - \frac{3}{2}kf(U_n) - \frac{k}{2}f(U^*)$$
$$U_{n+1} = U_a + U_b$$

Suggests parallel implementation & splitting.
Allows highly efficient linear solvers.

2D Splitting Integrating Factor & ETD-RDP Scheme

$$u_t + Au = f(u).$$

$$v = e^{Bt}u, \quad v_t = e^{Bt}f(u) - e^{Bt}Au + Be^{Bt}u.$$

If A & B commute, $\quad v_t + (A - B)v = e^{Bt}f(e^{-Bt}v).$

Now take $B = A_1$ & $\tilde{f}(v) = e^{A_1 t}f(e^{-A_1 t}v)$

Apply ETD-RDP to this PDE:

$$v_t + A_2 v = \tilde{f}(v), \qquad v(0) = u_0.$$

2D Efficient ETD-RDP-IF Algorithm

$$\begin{aligned}
(I + A_2 k)u^* &= (u_n + kf(u_n)) \\
(I + A_1 k)u^* &= u^* \\
(I + \frac{1}{3}A_1 k)a_1 &= u_n \quad \& \quad (I + \frac{1}{4}A_1 k)b_1 = u_n \\
(I + \frac{1}{3}A_1 k)a_2 &= f(u_n) \\
(I + \frac{1}{4}A_1 k)b_2 &= f(u_n) \\
c_1 &= 9a_1 - 8b_1 \qquad c_2 = 9a_2 - 8b_2 \\
(I + \frac{1}{3}A_2 k)d_1 &= 9c_1 + 2kc_2 + kf(u^*) \\
(I + \frac{1}{4}A_2 k)d_2 &= -8c_1 - \frac{3}{2}kc_2 - \frac{1}{2}kf(u^*) \\
U_{n+1} &= d_1 + d_2.
\end{aligned}$$

Example 1. *The one-dimensional approximation of the* $-\Delta \mathbf{u}$ *is given by*

$$\mathbf{B}_n = -\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad h = \frac{1}{n+1},$$

$$\mathbf{B}_n = -\frac{1}{h^2}\begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad h = \frac{1}{n-1},$$

$$\mathbf{B}_n = -\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad h = \frac{1}{n},$$

for homogeneous Dirichlet, homogeneous Neumann, and periodic boundary conditions,

Example 2. *In two dimensions,* $\mathbf{A}$ *in (3) is given by* $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$ *with*

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^2) \times (s \cdot n^2)} , \\ \mathbf{A}_2 &= \mathbf{B}_n \otimes \mathbf{I}_n \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^2) \times (s \cdot n^2)} , \end{aligned}$$

where $\mathbf{I}_n$ *denotes the identity matrix of size* n *(see also pp. 1345–1346 in [8] for the Dirichlet and Neumann case). The extension to three dimensions reads* $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3$ *with*

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{I}_n \otimes \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^3) \times (s \cdot n^3)} , \\ \mathbf{A}_2 &= \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{I}_n \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^3) \times (s \cdot n^3)} , \\ \mathbf{A}_3 &= \mathbf{B}_n \otimes \mathbf{I}_n \otimes \mathbf{I}_n \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^3) \times (s \cdot n^3)} . \end{aligned} \tag{4}$$

Note that the extension to four dimensions or even higher follows an obvious pattern. Precisely, we have $\mathbf{A} = \sum_{i=1}^{d} \mathbf{A}_i$ *with*

$$\mathbf{A}_i = \underbrace{\mathbf{I}_n \otimes \cdots \otimes \mathbf{I}_n}_{(d-i)\,times} \otimes \underbrace{\mathbf{B}_n}_{pos\ d-i+1} \otimes \underbrace{\mathbf{I}_n \otimes \cdots \otimes \mathbf{I}_n}_{(i-1)\,times} \otimes \mathbf{D} \in \mathbb{R}^{(s \cdot n^d) \times (s \cdot n^d)} . \tag{5}$$

Example of proof of commutativity

$$\begin{aligned}
\mathbf{A}_1\mathbf{A}_2 &= (\mathbf{I}_n \otimes \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{D})(\mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{I}_n \otimes \mathbf{D}) \\
&= \mathbf{I}_n\mathbf{I}_n \otimes \mathbf{I}_n\mathbf{B}_n \otimes \mathbf{B}_n\mathbf{I}_n \otimes \mathbf{D}^2 \\
&= \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{B}_n \otimes \mathbf{D}^2
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}_2\mathbf{A}_1 &= (\mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{I}_n \otimes \mathbf{D})(\mathbf{I}_n \otimes \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{D}) \\
&= \mathbf{I}_n\mathbf{I}_n \otimes \mathbf{I}_n\mathbf{B}_n \otimes \mathbf{B}_n\mathbf{I}_n \otimes \mathbf{D}^2 \\
&= \mathbf{I}_n \otimes \mathbf{B}_n \otimes \mathbf{B}_n \otimes \mathbf{D}^2 \, .
\end{aligned}$$

Unwinding to a fully discrete scheme

ETD-RDP-IF scheme in three dimensions

For $n = 0, 1, 2, \ldots, T/k - 1$ compute

$$\begin{aligned}
\mathbf{U}_{n+1} &= \left(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_3\right)^{-1}\left[\left\{9\left(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_2\right)^{-1} - 8\left(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_2\right)^{-1}\right\}\right. \\
&\bullet \left.\left\{9\left(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_1\right)^{-1} - 8\left(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_1\right)^{-1}\right\}\{9\mathbf{U}_n + 2k\mathbf{f}(\mathbf{U}_n)\} + k\mathbf{f}(\mathbf{U}^*_{n+1})\right] \\
&- \left(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_3\right)^{-1}\left[\left\{9\left(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_2\right)^{-1} - 8\left(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_2\right)^{-1}\right\}\right. \\
&\bullet \left.\left\{9\left(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_1\right)^{-1} - 8\left(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_1\right)^{-1}\right\}\left\{8\mathbf{U}_n + \frac{3k}{2}\mathbf{f}(\mathbf{U}_n)\right\} + \frac{k}{2}\mathbf{f}(\mathbf{U}^*_{n+1})\right] \\
\mathbf{U}^*_{n+1} &= (\mathbf{I}_m + k\mathbf{A}_3)^{-1}(\mathbf{I}_m + k\mathbf{A}_2)^{-1}(\mathbf{I}_m + k\mathbf{A}_1)^{-1}(\mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n))
\end{aligned} \tag{23}$$

with $\mathbf{U}_0 = \mathbf{U}(0)$.

Schematic: 3D, parallel, multi-threaded PC

Time

<table>
<tr><th>1st thread</th><th>2nd thread</th><th>3rd thread</th></tr>
<tr><td colspan="3">$\mathbf{f}(\mathbf{U}_n)$</td></tr>
<tr><td colspan="3">$w^* = \mathbf{U}_n + k\mathbf{f}(\mathbf{U}_n)$</td></tr>
<tr><td>$(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_1)a_1 = \mathbf{U}_n$</td><td>$(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_1)b_1 = \mathbf{U}_n$</td><td>$(\mathbf{I}_m + k\mathbf{A}_1)w^{**} = w^*$</td></tr>
<tr><td colspan="3">$c_1 = 9a_1 - 8b_1$</td></tr>
<tr><td>$(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_1)a_2 = \mathbf{f}(\mathbf{U}_n)$</td><td>$(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_1)b_2 = \mathbf{f}(\mathbf{U}_n)$</td><td>$(\mathbf{I}_m + k\mathbf{A}_2)w^{***} = w^{**}$</td></tr>
<tr><td colspan="3">$c_2 = 9a_2 - 8b_2$</td></tr>
<tr><td>$(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_2)a_3 = c_1$</td><td>$(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_2)b_3 = c_1$</td><td>$(\mathbf{I}_m + k\mathbf{A}_3)\mathbf{U}^*_{n+1} = w^{***}$</td></tr>
<tr><td colspan="3">$c_3 = 9a_3 - 8b_3$</td></tr>
<tr><td>$(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_2)a_4 = c_2$</td><td>$(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_2)b_4 = c_2$</td><td></td></tr>
<tr><td colspan="3">$c_4 = 9a_4 - 8b_4$</td></tr>
<tr><td colspan="3">$f^* = \mathbf{f}(\mathbf{U}^*_{n+1})$</td></tr>
<tr><td colspan="3">$f_1 = 9c_3 + 2kc_4 + kf^*$
$f_2 = -8c_3 + \frac{3}{2}kc_4 + \frac{k}{2}f^*$</td></tr>
<tr><td>$(\mathbf{I}_m + \frac{k}{3}\mathbf{A}_3)d_1 = f_1$</td><td>$(\mathbf{I}_m + \frac{k}{4}\mathbf{A}_3)d_2 = f_2$</td><td></td></tr>
<tr><td colspan="3">$\mathbf{U}_{n+1} = d_1 + d_2$</td></tr>
</table>

Enzyme Kinetics

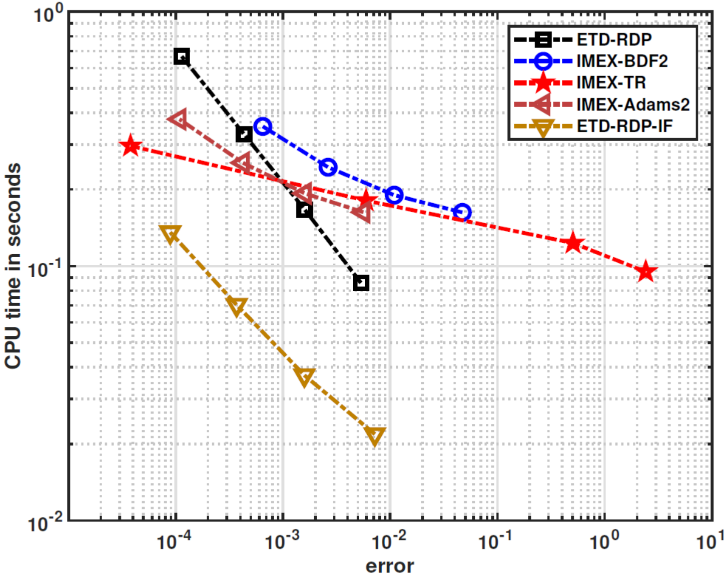
$$\frac{\partial u}{\partial t} = \gamma(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) - \frac{u}{(1+u)} \;\; 0 < x, y < 1, t > 0$$

with homogeneous Dirichlet boundary conditions and initial condition

$$u(x, y, 0) = 1 \;\; 0 \leq x, y \leq 1.$$

Incompatible IC-BC, gives difficulties with many schemes, but ETD-RDP damps out spurious oscillations.

Snapshot of convergence



Multi-thread machine parallel performance

$k = h$	Matlab	Fortran (1 thread)	Fortran (3 threads)
$1/100$	0.16	0.14	0.06
$1/200$	1.20	1.19	0.48
$1/400$	9.76	9.87	4.54
$1/800$	83.26	73.80	35.81

Figure: Efficiency, ETD-RDP-IF factor of 2.0-2.5 speedup.

2D Brusselator

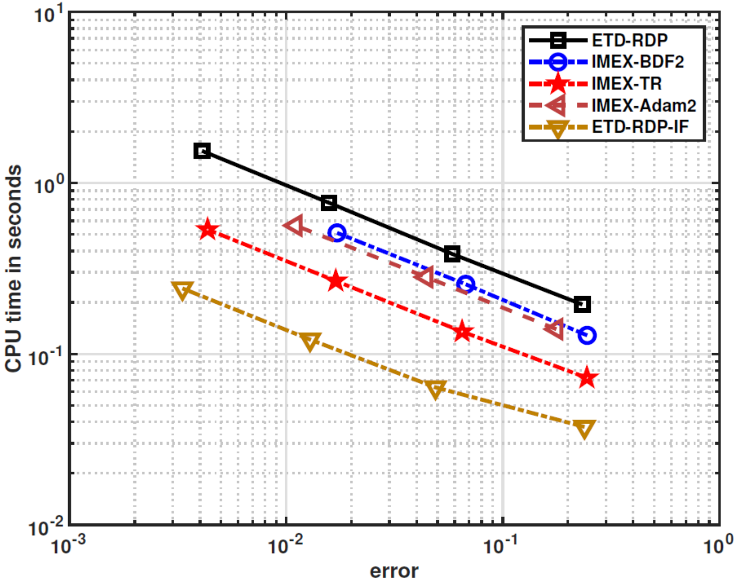
$$\frac{\partial u_1}{\partial t} = \epsilon_1 \Delta u_1 + u_1^2 u_2 - (A+1)u_1 + B$$
$$\frac{\partial u_2}{\partial t} = \epsilon_2 \Delta u_2 - u_1^2 u_2 + A u_1$$

$$\epsilon_1 = \epsilon_2 = 2.10^{-3}, A = 1.0, B = 3.4.$$

At the boundary of the domain homogeneous Neumann conditions are imposed. Initial conditions:

$$u(x, y, 0) = \frac{1}{2} + y, \quad v(x, y, 0) = 1 + 5x$$

Snapshot of convergence



Multi-thread machine parallel performance

$k = h$	Matlab	Fortran (1 thread)	Fortran (3 threads)
$1/100$	0.51	0.36	0.16
$1/200$	3.87	3.32	1.60
$1/400$	34.05	28.99	13.92
$1/800$	286.06	190.75	114.19

Figure: Efficiency, ETD-RDP-IF factor of 1.7-2.5 speedup.

3D

$$\frac{\partial u}{\partial t} = u + (1 + i\alpha)\Delta u - (1 + i\beta)u|u|^2$$

$$\alpha = 0 \; \& \; \beta = 1.3$$

At the boundary of the domain periodic conditions are imposed. Initial conditions: Normal random field (mean 0, st Dev 1).

$$\Omega = [0, 200]^3, \quad t \in (0, 150).$$

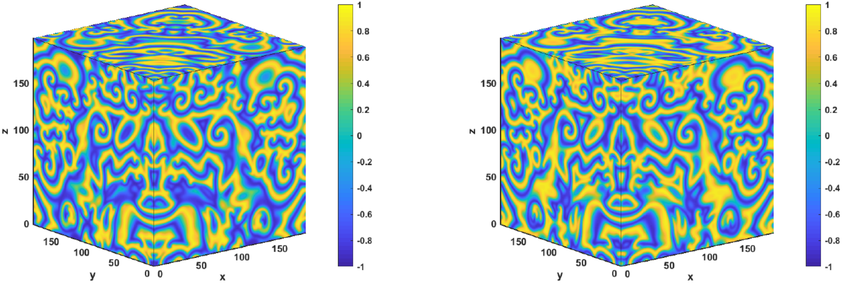


Figure 9: Real (left image) and imaginary part (right image) of the solution of the complex Ginzburg-Landau equation on the periodic domain $\Omega = [0, 200]^3$ for time $T = 150$ with the parameters $\alpha = 0$, $\beta = 1.3$, $N = 256$, and $k = 1/2$ using a series of Gaussian pulses field as initial condition.

N	h	Matlab	Fortran (1 thread)	Fortran (3 threads)
64	$25/8$	43.07	32.33	20.85
128	$25/16$	391.52	319.63	208.58
256	$25/32$	4010.79	2820.82	1872.22

Figure: Efficiency, ETD-RDP-IF factor of 1.5-2.0