

Maple 18 への更新によって、主要なカテゴリの処理効率が大幅に改善されました。処理効率の改善点は、以下のとおりです。

- 新バージョンGMP への更新により、長精度整数の多くの基本操作における[計算速度の向上](#)
- [LLVM コンパイラフレームワーク](#)の統合による、コンパイル時間の短縮とコード生成の効率化
- [多変数多項式操作](#)における、速度の大幅改善と次数範囲の拡大
- [数値線形代数](#)の効率改善
- 行列とベクトルの[線形結合](#)の高速化
- [ベッセルおよびハンケルの特殊関数操作](#)の大幅な高速化
- sort コマンドのマルチコア利用による、[ソート](#)の高速化
- グラフ理論での[グラフ描画](#)の性能改善

## 高精度計算の高速化

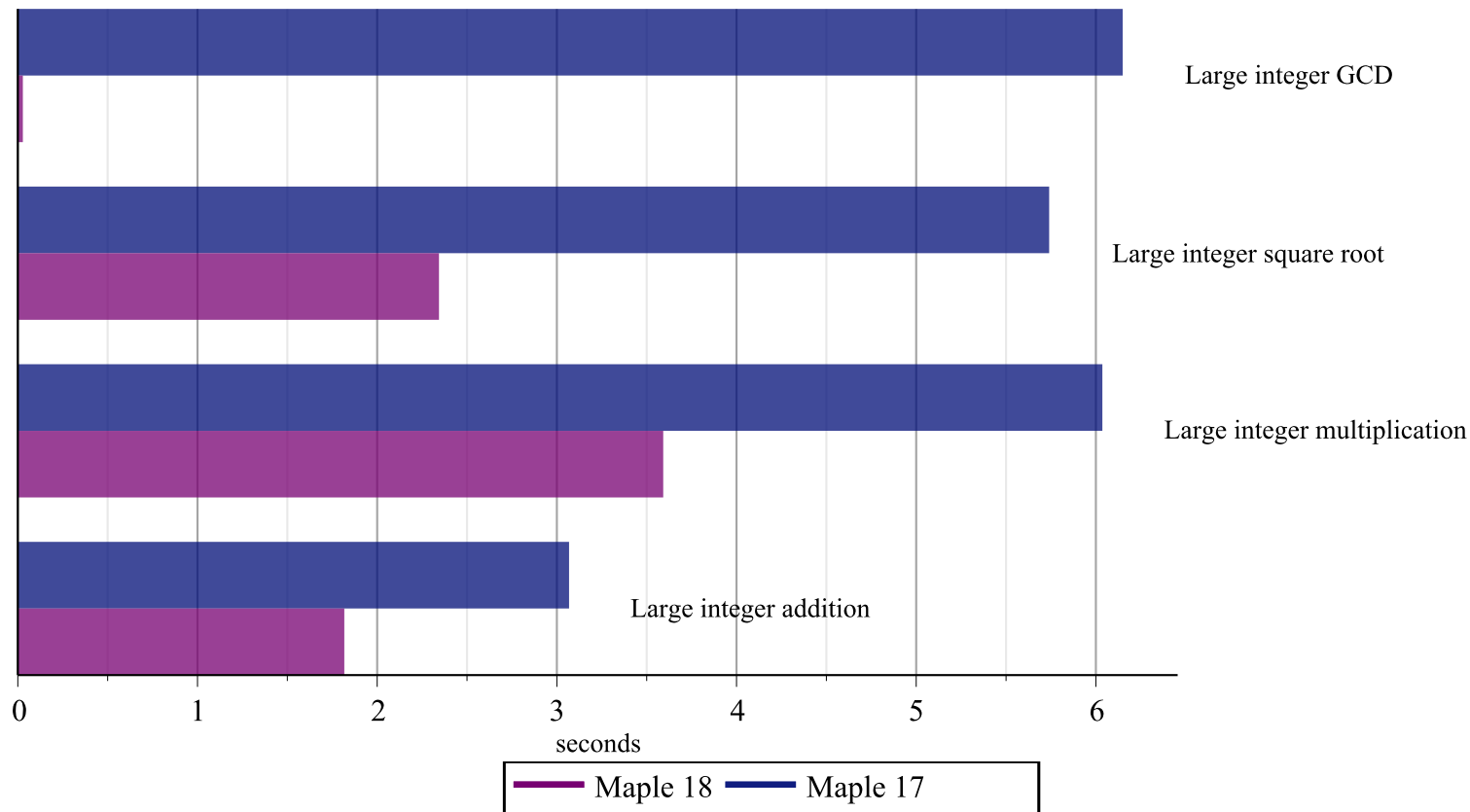
Maple 18 では、Maple エンジンで使われている GMP (GNU Multiple Precision Arithmetic Library) のバージョンを 5.1.1 にアップグレードしています。これにより、新しい CPU の性能に加え、既存のプラットフォーム (特に 64 ビットプラットフォーム) の最適化も改善されます。

```
> kernelopts( gmpversion );
```

"5.1.1"

(1.1

以下の図に、GMP を活用する基本操作における Maple 17 と Maple 18 の性能の違いを示します。これは、Intel 64 ビット i7 プロセッサで操作されています。



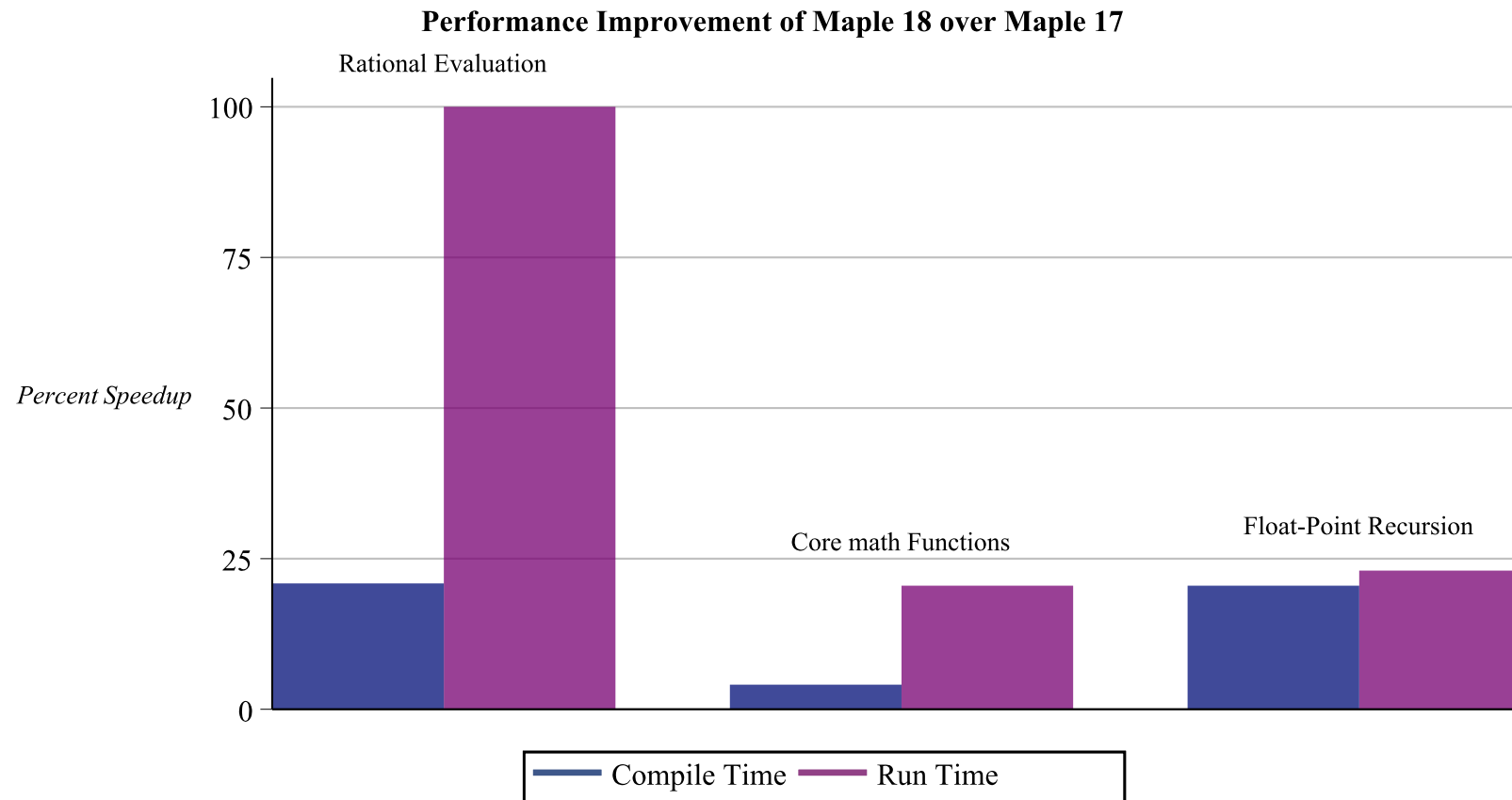
## LLVM の統合

Maple 18 では LLVM コンパイラインフラストラクチャを組み込むことで、[Compiler:-Compiler](#) コマンドのネイティブコードコンパイル機能を即時に利用可能なソリューションを提供します。LLVM を追加することで、従来のリリースに比べ、多くの利点があります。

LLVM は、Maple から呼び出されてコンパイル作業を行う個別のプロセスとして存在するのではなく、Maple カーネルに直接組み込まれています。従来のリリースと比べて、このコンパイラの緊密な統合は最大の長所と言えます。具体的には、2 次プロセス作成のオーバーヘッドを削除できるため、このストリームラインは `Compile` をコールします。プロシージャをコンパイルすると、結果のマシン語が直接メインメモリに生成されます。マシン語はここで実行することができます。従来のリリースでは、Maple では動的にリンクされたライブラリが作成され、実行前に読み込まれました。LLVM の統合によって、コストのかかるディスクアクセスを減少させることができます。

さらに、LLVMはこのシチュエーションで使用されるように設計されており、このため、より迅速かつ軽快に効率的なマシン語を生成することができます。

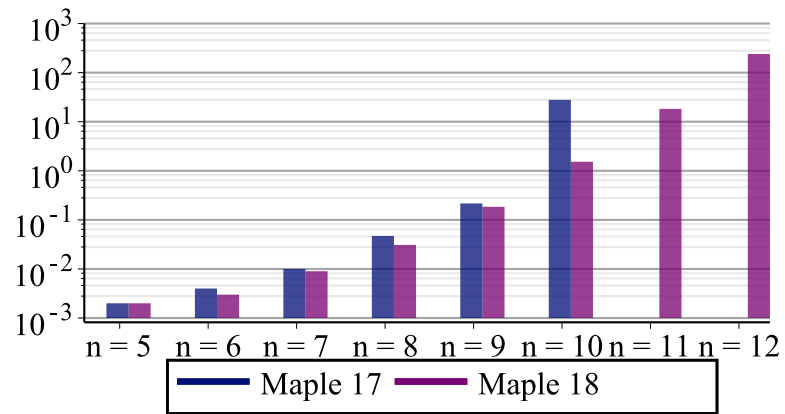
以下の図では、Maple 18 の改善により、コンパイル時間と実行時間を Maple 17 (64 ビット版 Windows) と比較して表示します。



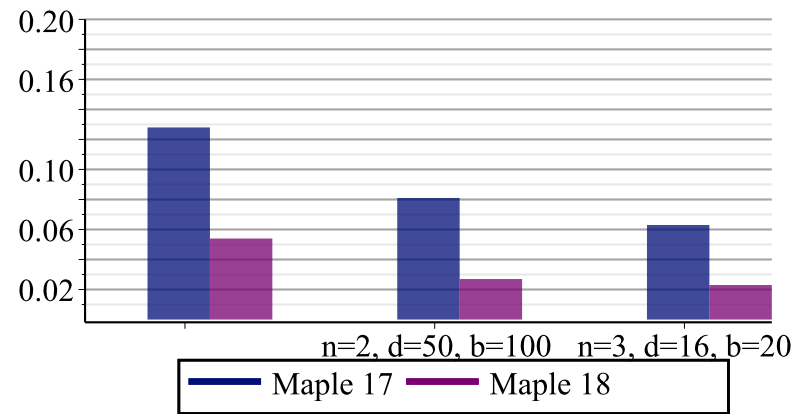
## ▼ 多項式の改善

Maple 18 での効率改善によって、大規模な多項式演算の速度も大幅に向上させることができます。詳細は、[多項式操作の改善](#)を参照してください。

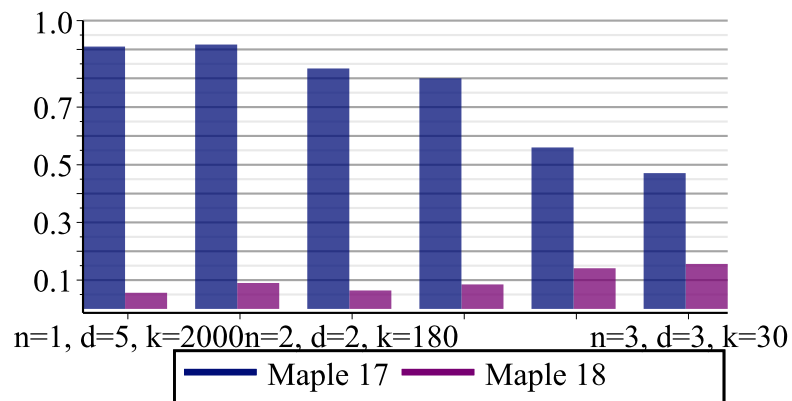
Vandermonde Determinant (seconds)



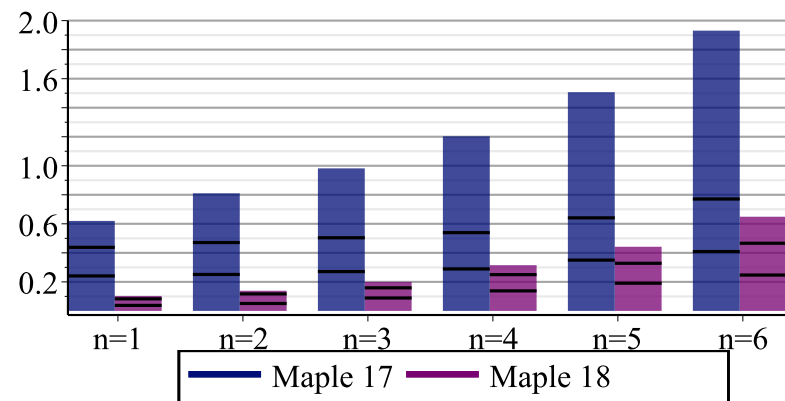
Multiply and Divide (seconds)



Sparse Powers (seconds)

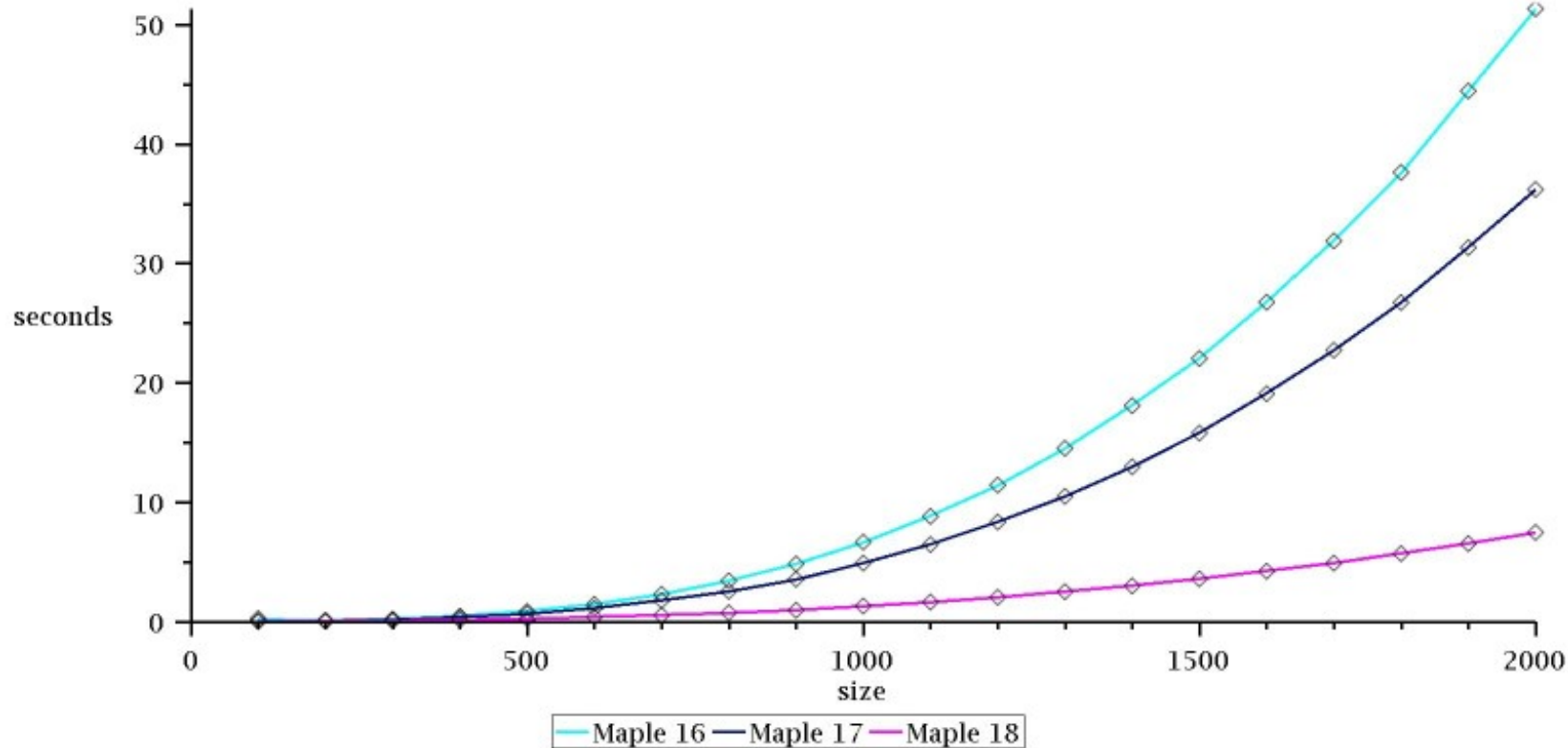


Small Polynomials Mod p (seconds)



行列の特異値分解の浮動小数点の計算は、[CLAPACK](#) の DGESDD ルーチンに基づき、分割統治法による統合を使用するようアップグレードされています。これにより、コマンド [SingularValues](#) および [LeastSquares](#) の処理が向上しました。

以下のグラフは、実行環境が Intel i5 760 2.80Ghz マシン上の 64 ビット Linux で、倍精度 `datatype=float[8]` 行列でのオプション `method=SVD` および単一の右辺ベクトルの [LeastSquares](#) コマンドを用いて作成されています。



## 線形結合効率

Maple 18 では、行列またはベクトルの線形結合の高速演算を実現するなど、性能が更新されています。以下の表では、12 のサンプル計算における性能の変化を示します。

	1	2	3	4	5	6	7	8	9	10	11	12
Maple 17	41.858	4.748	2.704	0.204	9.112	7.120	1.112	0.016	78.548	29.741	32.430	29.753
Maple 18	0.280	0.272	0.004	0.008	0.001	0.284	0.832	0.004	0.236	0.256	0.256	0.252
高速化の要因	149	17	676	26	9112	25	1.3	4	333	116	127	118



### Code for testing

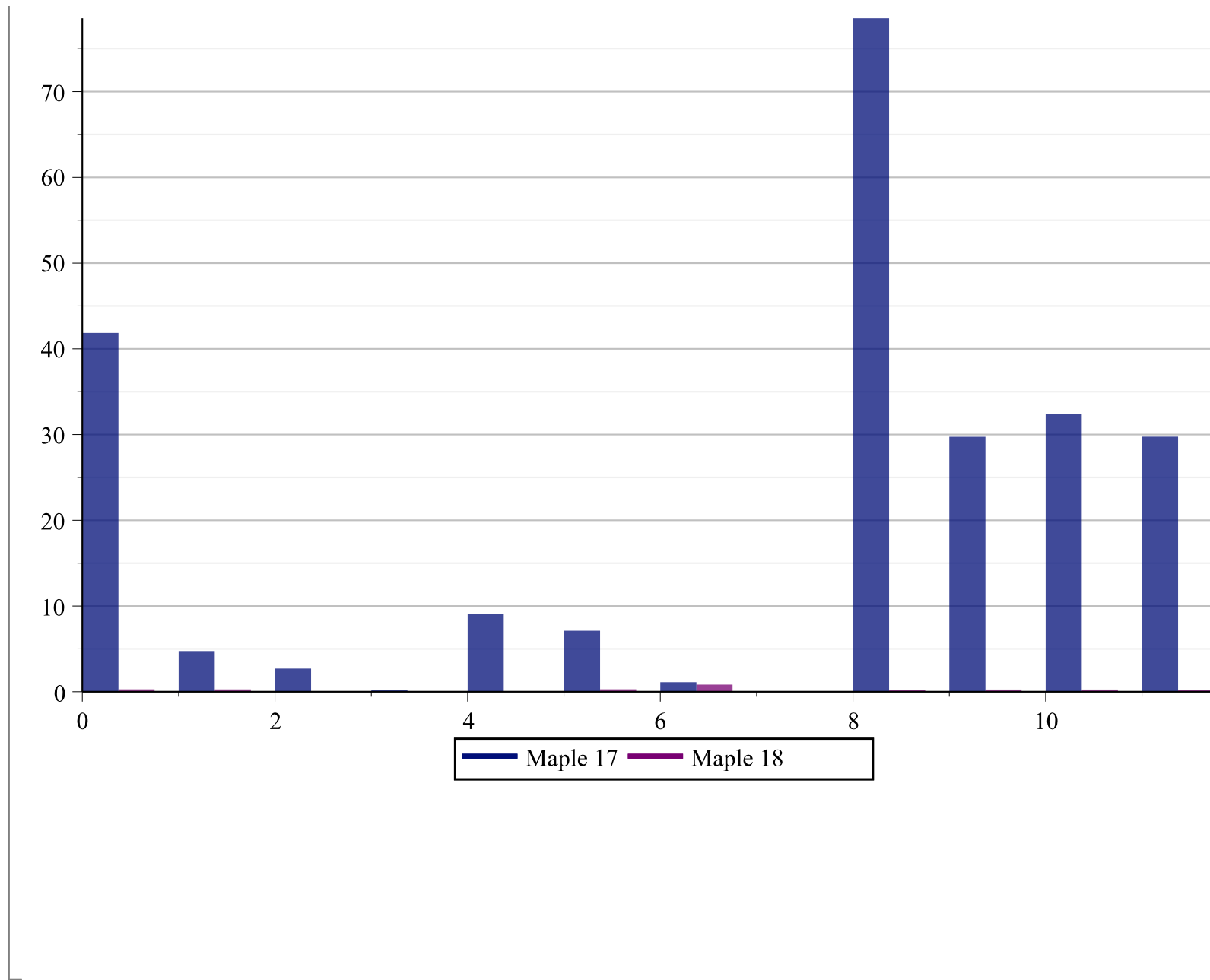
```
> Maple17Values := [ 41.858, 4.748, 2.704, 0.204, 9.112, 7.120, 1.112, 0.016, 78.548, 29.741, 32.430, 29.753 ]:
```

```
> Maple18Values := [ 0.280, 0.272, 0.004, 0.008, 0.001, 0.284, 0.832, 0.004, 0.236, 0.256, 0.256, 0.252 ]:
```

```
> SpeedUpFactors := Maple17Values /~ Maple18Values
```

```
SpeedUpFactors := [ 149.4928571, 17.45588235, 676.0000000, 25.50000000, 9112.000000, 25.07042254, 1.336538462, 4.000000000, 332.8305085, 116.1757812, 126.6796875, 118.0674603 ] (5.1)
```

```
> Statistics:-ColumnGraph([ Maple17Values, Maple18Values ], legend = ["Maple 17", "Maple 18"]);
```



## ▼ ハードウェア関数と特殊関数

ベッセル関数 [BesselJ](#)、[BesselY](#)、[BesselI](#)、および [BesselK](#)、およびハンケル関数 [HankelH1](#) および [HankelH2](#) は、[evalhf](#) 環境で使用するために、コン

パイルされた特殊関数の Maple のコアライブラリに加えられています。結果、ベッセル関数を含む式のプロットなどの操作効率が大幅に改善されます。以下の例を確認してください。

```
> V := Array(0..200, i → -100. +  $\frac{i \cdot 3.14159}{3}$ , datatype = float[8]):
```

```
> X := Array(0..200, i → -100. +  $\frac{i \cdot 2.7172}{2.8}$ , datatype = float[8]):
```

```
> st := time( ) : Array(0..200, 0..200, (i, j) → BesselJ(V[i], X[j]), datatype = complex[8]) : time( ) - st;
```

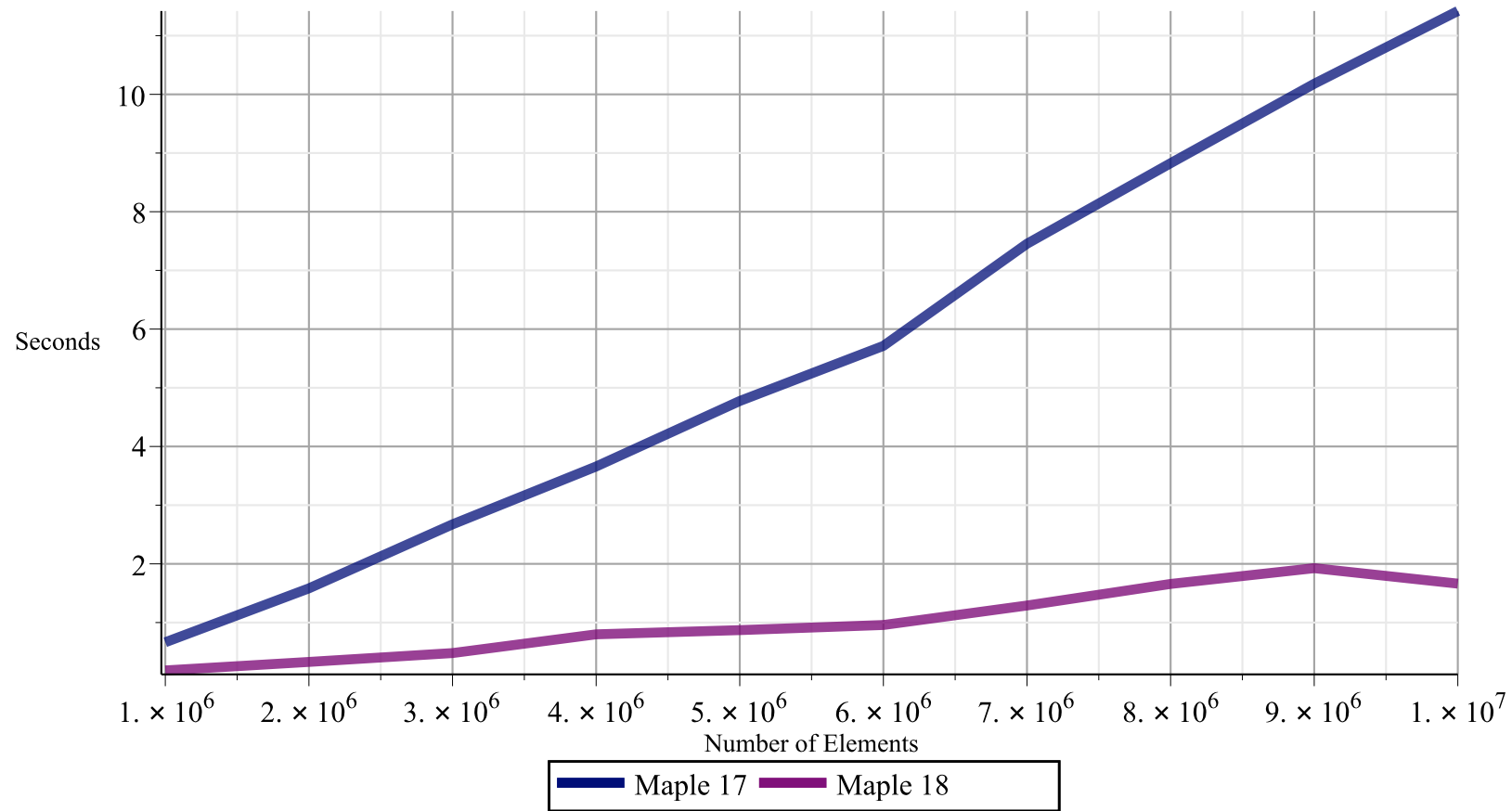
1.780

(6.1

## ▼ sort コマンドの改善

Maple 18 では、ソートアルゴリズムとキーソートオプションの並列化によって、ソートを迅速に行うことができます。以下のグラフでは、Maple 17 と Maple 18 で実行されたランダム順列のソート時間の差を示しています。





ソートの改善点の詳細は、[言語とプログラミング](#)更新ページを参照してください。

## ▼ グラフ理論

`GraphTheory` パッケージへの更新によって、大きいグラフ用の `DrawGraph` コマンドの性能が改善されました。これは、サブルーチン `GetEdgesColor` および `GetEdgesThickness` で、グラフが疎であるときに、疎行列を使用した結果です。以下の表に、一部の例のタイミングを示します。詳細は、[グラフ理論](#)ページを参照してください。

## 性能比較

グラフ	Maple 18	Mathematica® 9
CycleGraph(10 <sup>3</sup> )	0.124s	0.235s
CycleGraph(10 <sup>4</sup> )	1.140s	2.226s
CycleGraph(10 <sup>5</sup> )	18.10s	27.59s

### ▼ 参照

[Maple 18 の新機能](#)

▶ Pages That Link to This Page