1 図の取り込み

dvipdfm$x$ で直接サポートされているフォーマットは PNG, JPEG, PDF のみです。PostScript や Encapsulated PostScript (EPS) を取り扱うこともできますが、この場合は外部プログラムを利用して PDF への変換を行ないます。ここでは、dvipdfm$x$ で画像を扱う際に生じる問題とその対処方法についてそれぞれのフォーマットごとに解説します。

1.1 PostScript

PostScript ファイルの取り込みがうまくいかない原因は、ほとんどの場合 Ghostscript のインストールの不備や誤ったオプションの指定によるものです。よくある間違いとして、誤った用紙サイズ指定により画像が一部しか表示されないという場合があります。例えば、 `dvipdfmx.cfg` で

```
D "gs -q -dNOPAUSE -dBATCH -sPAPERSIZE=a4 ..."
```

のように `-sPAPERSIZE=a4` の指定がある場合、A4 用紙に収まらない領域、つまり左下隅の座標が $(0, 0)$ で右上隅の座標が $(595, 842)$ である矩形領域 (単位はポイント) の外の領域、に描かれたものは表示されません。多くの場合は単に `a4` を `a0` に変更するといった処置で対処できますが、x, y 座標が負の値になる領域も表示させたい場合は、`epstopdf` コマンドなどによりあらかじめ PDF に変換しておき、 PostScript ではなく PDF を使うようにする必要があります。[*1]

PostScript から PDF への変換に Ghostscript を利用しているのであれば、取り込みに失敗する場合はほとんど常に Ghostscript による変換そのものが失敗しています。このような場合は `dvipdfmx.cfg` に記述してある実行オプションを使って[*2]`gs` を実行してみてください。

PostScript を扱う場合に生じる問題の多くは dvipdfm$x$ とは無関係ですので、ファイルに問題がないかの確認は dvipdfm$x$ とは完全に切り離して行なって下さい。Ghostscript 等の問題が解決されない限り PostScript ファイルの取り込みに成功することはありません。

1.2 PDF

PDF の取り込みにはまだ多くの制限がありますが、多くの場合に問題なく動作します。なるべく dvipdfm$x$ のバージョンが 20031207 以降のものを使うようにして下さい。

お使いのアプリケーションから PDF で出力する際には以下の点に注意してください。

- テキスト圧縮方式を指定できるのであれば、これを常に zlib (flate) にする。画像の圧縮方式はなんでも構いません。
- 「アスキー形式で保存」というようなオプションがある場合はこれを必ず無効にする。
- 暗号化により保護された PDF は取り扱えません。セキュリティ関連のものはすべて無効する必要があります。

[*1] PDF を使う場合はオリジナルの dvipdfm に附属の `ebb` コマンドを使い `.ebb` ファイルを作り、graphicx パッケージのオプションとして `dvipdfm` を指定します。

[*2] `%i` は入力 PostScript ファイル名に、`%o` は適当な出力 PDF ファイル名に置き換えます。

図 1　元の画像 (左、5.7K bytes) を 1/2 に縮小して表示させたもの (中央) とあらかじめ 1/2 に縮小しておいたもの (右、2.5K bytes)。この例では、Independent JPEG Group による JPEG ライブラリ配布物に含まれる `testorig.jpg` を使用した。

また、PDF バージョンを不必要に高く設定しない方が無難です。dvipdfm$x$ は設定された PDF バージョンより高いバージョンを要求する PDF ファイルは取り込まないようになっています。目安として、文書中に日本語が使われている場合は PDF 1.2 (Acrobat 3) 以上に、和文 TrueType フォントを使用している場合は PDF 1.3 以上、グラデーション (shading) や透過色が使われているものは PDF 1.4 以上に設定します。タグつき PDF にはしないでおいてください (未対応です)。基本的には、必要のない情報を書き出さずになるべく単純な PDF にしておくと取り込みに成功する可能性が高くなります。

1.3　PNG と JPEG

JPEG 形式の画像ファイルを取り扱う場合によく問題になるのはファイルサイズの問題です。デジダル・カメラなどから取得した画像では、解像度が不必要に高すぎるためにファイルサイズが非常に大きくなるという問題が生じます。このような場合はあらかじめ画像のリサイズを行なっておきます。

適切な縮小率の値は想定している出力装置の能力に依存します。一般的なディスプレイで閲覧されることを想定しているのであれば、画像が拡大表示されることを考慮しても、解像度として 150 dpi 程度を上限に設定すれば良いでしょう。dvipdfm 附属の `ebb` コマンドは 1 インチ四方あたり 100x100 ピクセルが含まれるように BoundingBox の値を調整します。[*3] もし、`\includegraphics` オプションで `scale` の値を 0.5 程度にする必要があるのであれば、あらかじめ画像を 1/2 程度に縮小しておいたほうが良いでしょう。

JPEG 画像のサイズ変更を行なう場合は画像の劣化が顕著に現れる場合があります。このような場合には、画像処理専門のプログラムを使って、フィルタやパラメータなどの調整を手動で行なって下さい。dvipdfm$x$ では JPEG 画像は「そのまま」PDF ファイルに取り込まれるので、元の JPEG 画像のサイズの違いはそのまま出力ファイル・サイズに反映されます。

PNG の場合も基礎的な事柄は JPEG と同じですが、8-bit インデックス・カラーを使うことで画像ファイルを小さくすることができます。また、透過度情報が不要な場合 (特に α-channel) にはこれを使わないようにしておいた方が良いでしょう。α-channel (PDF では Soft Mask) が存在する場合、出力 PDF ファイルのサイズとビューアの描画速度に大きな違いがでることがあります。特定の色を完全透明にするだけという場合にはファイル・サイズにほとんど影響しません。

[*3] 角藤さんの W32TEX に含まれるものでは解像度を自由に指定できるようになっており、またデフォルトの値はオリジナルのものとは異なります。

図 2　透過度情報を持つ PNG 画像の例。24-bit RGBA カラー (左、 26K bytes) と 8-bit インデックス・カラー (中央、9.0K bytes)。8-bit インデックス・カラーへの変換には pngquant をもちいた。右は JPEG 形式で保存したもの (4.5K bytes)。画像は米 Kodak 社のウェブ・サイト から入手可能なサンプル画像を加工したもの。

2　出力形式の選択

幾つかのアプリケーションでは図を製作する際に dvipdfm$x$ に適した出力形式を選択することで問題を回避することができます。

2.1　Gnuplot の場合

最近の Gnuplot は PDF[*4]や METAPOST 出力をサポートしています。数式や日本語を使わない場合は PDF への出力で多くの場合は足りるでしょう。数式を使う場合、dvipdfm$x$ での利用を前提とするのであれば METAPOST での出力が良いでしょう[*5]。

Gnuplot で METAPOST 出力を行なうには出力デバイスに `mp` を指定します。例えば、以下のような内容のファイルを `fig1.gp` として保存しておきます。

```
set term mp latex 'min10' 10
set output 'fig1.mp'
set nokey
set title '特性温度が $T_v$ である振動モードの比熱に対する寄与.'
set xlabel '$2T/T_v$'
set ylabel '$\frac{RT_v^2}{4T^2\sinh^2(\frac{T_v}{2T})}$'
set ytics ('$0$' 0, '$0.5R$' 0.5, '$R$' 1)
set xrange [0:2.1]
set yrange [0:1.05]
plot (1/x)*1.0/sinh(1.0/x)
```

[*4] PDF 出力には PDFlib が必要です。

[*5] dvipdfm x の 20031207 以前のバージョンでは日本語 METAPOST に関連した部分にバグがあるものがあります。なるべく最新のものを使うようにして下さい。

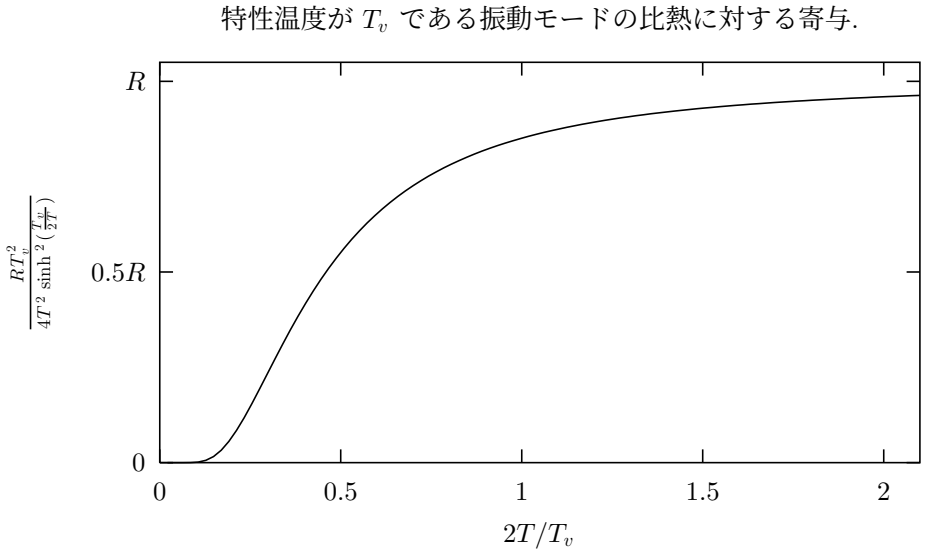

図 3 Gnuplot による METAPOST 出力例。

これを gnuplot で処理します。

```
% gnuplot fig1.gp
```

これで fig1.mp というファイルができるので pTEX 対応版 METAPOST[*6]で処理します[*7]。

```
% jmpost --tex=platex fig1.mp
```

fig1.0 という名前のファイルが生成さるので、これを例えば fig1.mps という名前にかえ、\includegraphics などで取り込みます。

dvipdfm$x$ で METAPOST を使う利点は、ラベルなどに使われる文字列が本文のものととほとんど同じように処理されることです。上記の例では 図 3 のようになります。日本語部分の文字コードには pTEX および METAPOST のコンパイル時に指定したものを使います。詳しい使い方については Gnuplot のドキュメントなどを参照して下さい。

[*6] 鈴木秀幸さんによる日本語 METAPOST のページから入手できます

[*7] Gnuplot の mp 出力ドライバではラベルに TEX コマンドを使うことができます。LATEX の場合は必ずドライバ・オプションに latex を指定し、また jmpost のコマンドライン・オプションに--tex=platex を指定します。