

## 論文

# Visual Studio 2005 による放射線時系列データ解析 ソフトウェアの開発

芳原 新也\*、堀口 哲男\*、伊藤 眞\*

**Development of Analysis Software for Radiation Time-Series Data  
with the use of Visual Studio 2005**

Sin-ya HOHARA\*, Tetsuo HORIGUCHI\*, Shin ITO\*

Time-Series Analysis supplies a new vision that conventional analysis methods such as energy spectroscopy haven't achieved ever. However, application of time-series analysis to radiation measurements needs much effort in software and hardware development. By taking advantage of Visual Studio 2005, we developed an analysis software, "ListFileConverter", for time-series radiation measurement system called as "MPA-3". The software is based on graphical user interface (GUI) architecture that enables us to save a large amount of operation time in the analysis, and moreover to make an easy-access to special file structure of MPA-3 data. In this paper, detailed structure of ListFileConverter is fully explained, and experimental results for counting capability of MPA-3 hardware system and those for neutron measurements with our UTR-KINKI reactor are also given.

*Keywords: Visual Studio 2005, Visual C#, multi thread technique, Time-Series Analysis, MPA-3, radiation measurement*

---

\*近畿大学原子力研究所      Atomic Energy Research Institute, Kinki University

## 1. はじめに

近年、パーソナルコンピュータ (PC) は非常に一般化・高速化が進んでいる。CPUの単純な周波数向上のみならず、デュアルコア・クアッドコアCPUも登場し、PCの演算能力は数年前の大型計算機に匹敵するほどになっている。これほどの性能にも関わらずコストパフォーマンスは非常に良く、安価に購入できることから1人が数台のPCを所有する状況になっている。

それゆえ、PCの研究現場への適用は、実験・解析を中心に多くの場面で見ることが出来る。特に物理系実験のデータ解析現場では、他の分野と異なり桁外れに多くの数値データ解析を行う必要があることから物理系研究室でのPC普及率は高い。

また、そのデータ量から物理系の実験データ解析を既存の表計算ソフトウェア等で行う事は非常に非効率的であり、実質不可能に近い状況にある。この様な状況下において、物理系研究室においては、PCはプログラミング言語と一緒に導入されることが多い。

上記の様な状況において導入されるプログラミング言語は歴史的にFortran、C言語が選択される頻度が高く、特に原子力系研究分野においてはFortranが未だに根強い人気を保っている。しかしながら、これらのプログラミング言語による開発ソフトウェアは貧弱なユーザー・インタフェース(UI)しか有していない事が多く、エンドユーザーに対して使用に際し非常に多大な労力を強要する傾向にある。

原子力系研究分野においても一般的に流通しているシミュレーション・解析コードのUIの多くはユーザーフレンドリーと言い難いものが多い。それゆえ、コードを使いこなすには、非常に大きな労力と多大な時間、そして更に多くの忍耐力を必要とする。その為、研究グループからコードを使用する事が出来る「特殊技能者」が居なくなってしまうと、

研究活動自体が停止してしまうというリスクを抱え込んでしまう。まさにBen Shneidermanの言う「コンピューター・システムを放棄せざるを得ない」状況<sup>1)</sup>に陥る可能性があるわけである。

多くの研究室においては解析担当者が学生等である事が多く、せっかく開発された解析ソフトウェア・シミュレーションコードが担当学生の卒業や他研究室への進学に伴い、引継ぎの不十分さや使用マニュアルの不備等から単なるハードディスク(HDD)上のゴミとなってしまうことが多い。

このような状況を改善するためには、UIや対象ソフトウェアへのアクセスを簡易にすることにより、一旦開発されたソフトウェアの使用行為を「特殊技能」から「誰にでも簡単に出来る操作」へと切り替えることが重要となる。そのためには、開発言語の選択、インストーラの整備等が必要となり、必然的にUI開発に重点の置かれた統合開発環境を使用する事となる。現在、最も普及しているOSであるWindowsシリーズにおいては、これらの開発の為に、OS開発元のMicrosoft社が統合開発環境Visual Studioシリーズを公開している。

本文では、上記のVisual Studioを用いて行った、近畿大学原子力研究所の所有する時系列測定装置MPA-3用のデータ解析ソフトウェア開発に関して述べる。

## 2. 時系列測定装置MPA-3

MPA-3はドイツのFAST ComTec社製の放射線計測用マルチパラメータデータ収集装置である。また、データストリーム内に一定間隔の時間情報(タイムスタンプ)を挿入することで時系列データ測定装置として使用する事も可能なシステムである。MPA-3へのADC接続は4ADC同時接続が可能であり、段階的に8ADC、12ADC、16ADCと拡張することも出来る。また、MPA-3は専用インターフェースボード(IF-Board)を介してPCと接続可能であ

り、接続したPCから操作する事が出来る。MPA-3システムの構成概要を図1に示す。

MPA-3に接続したADCからのデータ取得はPCからの操作で任意のADC組み合わせに対してAND計測、OR計測が可能となっている。測定データはリストデータとして制御PCのローカルHDDに保存可能であり、MPA-3からの最大データスループットは16ADC計測時に6Mイベント/秒である。

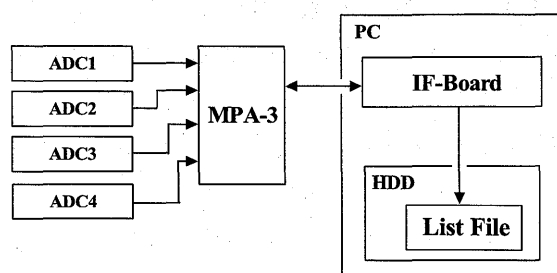


図1 MPA-3システムの構成概要

時系列測定を行う場合、データの時間精度が問題となるが、上記システムではMPA-3からデータ出力される時点で時間情報出力が最優先されるようになっている。また、リアルタイムオプションを付加する事により、出力時間情報の時間精度の向上を図ることも可能となっている。

近畿大学原子力研究所では、上記システムを原子炉施設及びトレーサー・加速器棟に導入済みで、両施設において非常に精度の良い時系列データ測定が可能となっている。両施設に導入しているMPA-3システムの構成を表1に示す。

### 3. MPA-3リストファイル構造

MPA-3から制御PCへ出力されるデータはバイナリ形式のリストファイルとしてローカルHDDへと保存される。このリストファイルのデータは通常FAST ComTec社から提供されるソフトウェアを介してのみのアクセスに限定されている。これは、FAST ComTec社の想定する解析に対しては、統一したアクセスが可能であるという点に関して非常に強力である一方、想定外の解析に対しては非常にアクセスが難しくなっている。

MPA-3リストファイルの基礎構造は前半のASCII形式部分と後半のバイナリ形式部分に分かれる。前半のASCII形式部分には接続されているADC情報・測定開始時刻等の測定諸条件が書き込まれており、後半のバイナリ形式部分にはMPA-3からの出力情報が書き込まれている。したがって、後半のバイナリ形式部分が解析の主要対象である。

MPA-3リストファイルのバイナリ形式部分は独自のデータフォーマットを持っており、これが第三者からのアクセスを難しくしている。このデータフォーマットを解析した結果、基本的に時間情報コード（タイムスタンプ）とADCデータコード及びそのヘッダーコードにより構成されることがわかった。表2にシングルADCの場合の書き込みコード一覧を示す。表2に示したコードは図2に示す書き込み則に従い、バイナリ形式部分内に並んでいる。

表1：近畿大学原子力研究所の所有するMPA-3システム

	ADCタイプ	ADC数	IF-Board バス	PCのOS
原子炉施設	ウィルキンソン型 (7074 : FAST ComTec)	4	ISA	Windows 2000
トレーサー・加速器棟	逐次比較型 (7072T : FAST ComTec)	4	PCI	Windows XP

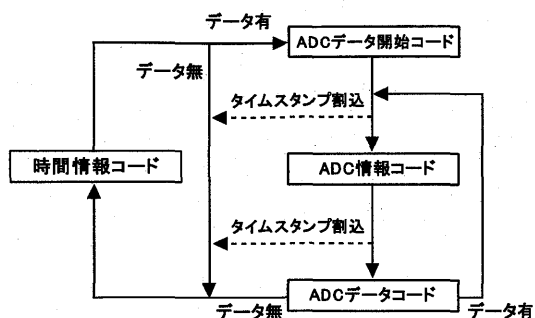


図2 リストファイルの書き込み規則

#### 4. 解析ソフトウェアの設計概要

上記のMPA-3リストファイルのバイナリ形式部分のASCII形式への変換および時系列解析を

目的として解析ソフトウェア（開発コード名：ListFileConverter）を開発した。開発にはコードメンテナンスの容易化も見据えて、オブジェクト指向言語であるVisual C# 2005を用いた。

今回のコード開発ではシングルADC測定 of MPA-3リストファイルに限定し、特に炉物理計測に特化した解析機能を付加した。ListFileConverterに実装した機能を表3に示す。

表3に示す機能は大きく分けて、ファイルアクセスとデータ表示を主な処理とする機能に大別することが出来る。これらの機能の内、ファイルアクセスは非常に重い処理となる為、スレッドを立てその中にファイルアクセス処理を切り離し、本体スレッドの処理を軽くした。また、データ表示に関しては常

表2：リストファイルのコード一覧

	コード（16進表記）
時間情報コード（タイムスタンプ）	FF FF 00 40 or FE FF 00 40
ADCデータ開始コード	FF FF FF FF
ADC情報コード	01 00 00 80
ADCデータコード	FF FF ** **

表3：ListFileConverterの実装機能一覧

実装機能	処理分類
解析データに関する情報の作成	ファイルアクセス
リストファイルのASCII書き換え	ファイルアクセス
エネルギー分布の作成	ファイルアクセス
分散対平均比の計算	ファイルアクセス
計数率の時間トレンドの作成	ファイルアクセス
データの一部の切り出し	ファイルアクセス
エネルギー分布の表示	データ表示
分散対平均比の表示	データ表示
計数率の時間の表示	データ表示

表4：解析結果ファイルのフォーマット

ファイルの種類	ファイルフォーマット					
データ情報	1行目		総計測時間 [ms]			
	2行目		総計数 [counts]			
	3行目		総計数率 [cps]			
	4行目		ADC-Dead率 [%]			
リストデータ (ASCII形式)	1列目		2列目		3列目	
	相対時刻 [ms]		ADCの Dead情報 0 : Live、1 : Dead		ADCの波高情報	
エネルギー分布	1列目			2列目		
	チャンネル番号 [ch]			カウント数 [counts/ch]		
分散対平均比*)	1行目			設定ULD [ch]		
	2行目			設定 LLD [ch]		
	5行目 以降	1列目	2列目	3列目	4列目	5列目
		ゲート 時間 [ms]	ゲート 数 [個]	分散	平均	分散対 平均比
計数率の時間トレンド	1行目		設定ULD [ch]			
	2行目		設定 LLD [ch]			
	5行目以降	1列目		2列目		
		相対時刻 [sec]		計数率 [counts / unit time]		

\*) 原子炉雑音解析手法の一つであるファインマン $\alpha$ 法を用いれば、異なる計測時間幅（ゲート時間）に対する分散対平均比を解析することで、原子炉の炉物理定数である即発中性子減衰定数と「 $\varepsilon \cdot (\beta - \rho)^{-2}$ 」（ $\varepsilon$ ：全核分裂あたりの計数効率、 $\beta$ ：全遅発中性子割合、 $\rho$ ：反応度）が求まる。これらは、炉の反応度や中性子寿命等を決定する為の足掛かりとなる定数である。

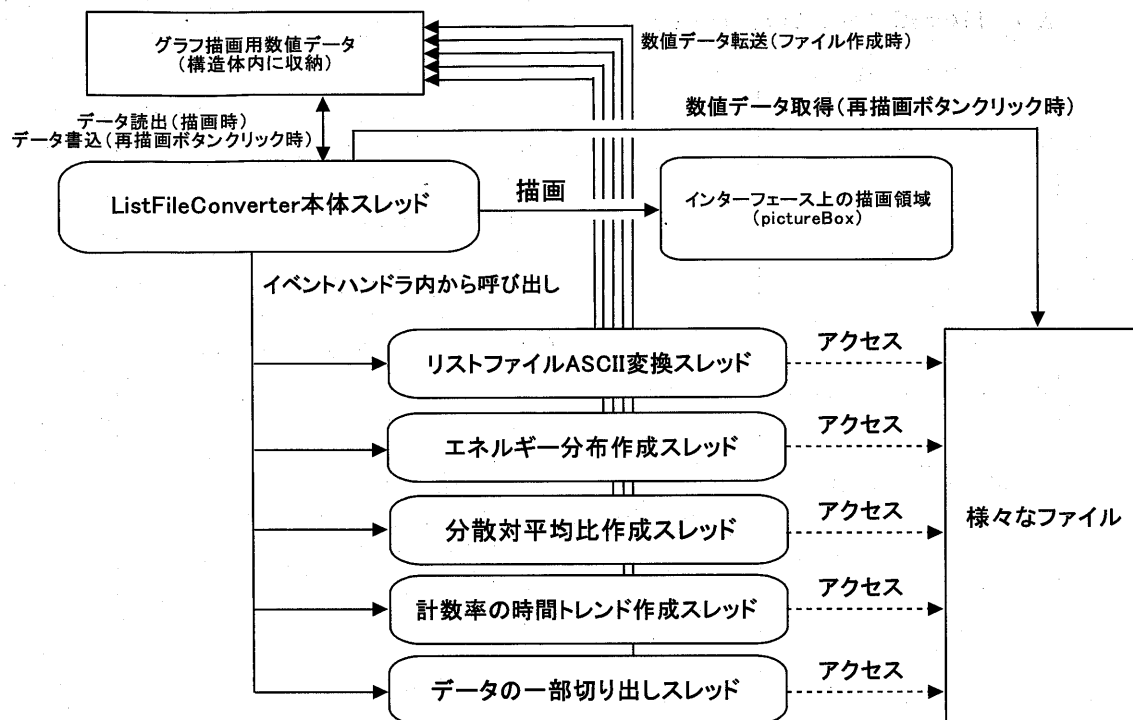


図3 ListFileConverterの設計概要

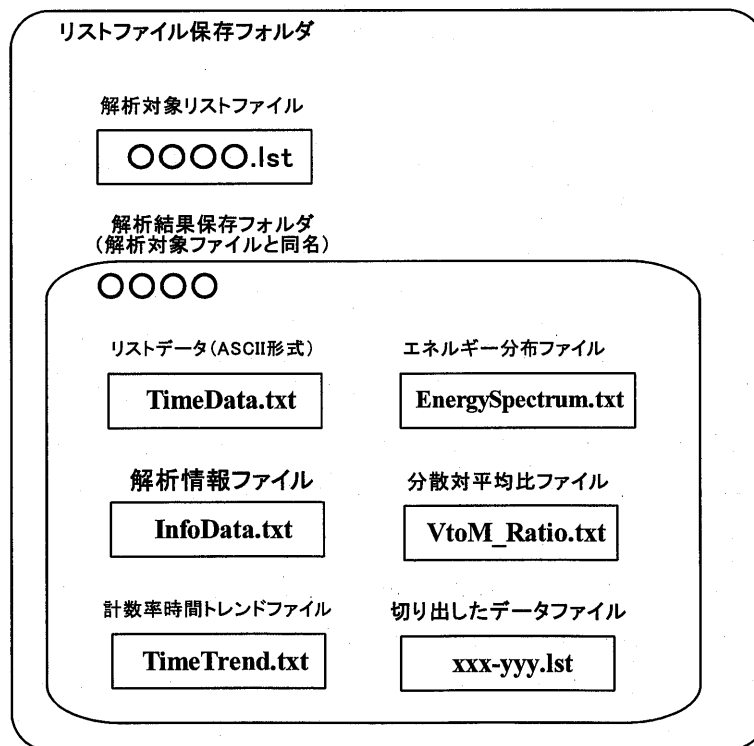


図4 解析結果ファイルの保存場所

時表示が必要であり、表示毎のファイルアクセスを避ける為に、名前空間内に設置した構造体に表示用数値データをバッファする設計にした。

ListFileConverterの設計概要を図3に示す。

さらに、解析結果等は他の表計算ソフト等からのアクセスを容易にする為に、選択したリストファイルに対し同名のフォルダを作成し、その中に解析結果ファイルを保存する設計にした。解析結果ファイルの種類とフォーマットを表4に、ファイル場所の概要を図4に示す。

## 5. 解析ソフトウェアのユーザー・インタフェース

ListFileConverterでは特に学生による使用を主眼に置き、ユーザー・インタフェースの設計を行った。ListFileConverterのウィンドウには必要最低限の操作ボタンと情報のみを表示し、プルダウンメニューは不採用とした。また、ファイル操作とデータ表示との切り替えを容易にするため、タブを採用した。インターフェースウィンドウの例を図5に示す。

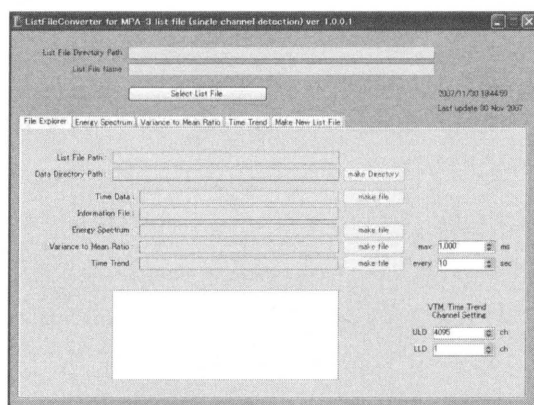


図5 ListFileConverterのインターフェース  
ウィンドウ例

必要最低限の操作ボタンと情報窓のみである。

## 6. 解析ソフトウェアのコーディングとインストーラの作成

ここでは、ListFileConverterのコーディングの

うち、ポイントとなる一部とインストーラ作成に関して述べていく。先述したように、開発にはVisual studio 2005 / Visual C# 2005を用いた。

### 6-1. ボタンのEnabled制御

今回の開発コードでは、ファイルアクセス・ファイル作成はリストファイルに近い場所からアクセス優先順位が生じることとなる。これは、一旦リストファイルからASCIIファイルを作成し、これから解析結果が派生するように設計したからである。このファイルアクセス優先順位を明示的に示し、ユーザーからの操作ミスをなくす為に、ボタンのEnabled制御を関数にまとめ、各イベントハンドラ内で呼び出すこととした。disable\_buttons関数でListFileConverterの全ボタンを使用不可にし、enable\_buttons関数で、ファイル・フォルダの有無を分岐条件にして、ボタンの使用可／不可化を実装した。

### 6-2. スレッドによるファイルアクセス処理の切り離し

各ファイル作成処理はボタンクリックを合図として行うようにデザインした。この時、実際の処理は別スレッドに切り分けた為、各ファイル作成ボタンのクリックイベントハンドラ内はボタンのEnabled制御、スレッド呼び出しおよび状況のリストボックスへの表示のみとした。

スレッド (thread) とは「プログラムの一つの流れ」を意味する言葉である。マルチスレッド技術は一つのプログラム内で複数の処理を並列に処理することの出来る技術で、Microsoft社製OSではWindowsシリーズ以降でマルチスレッド対応となっている。

切り離された処理は別スレッド内に記述され、スレッド処理が終わったらその旨をリストボックスへ表示するようにした。

### 6-3. 解析結果の表示

解析結果の可視化には様々な選択肢があるが、今回は簡単なグラフ表示のみに限定し、ピクチャーボックスにより実装した。グラフ表示を1秒間で2回更新するためにTimerコンポーネントを用い、表示処理はTimer\_tickイベントハンドラ内に記述した。

表示に必要な数値データはファイル処理時、あるいはグラフ表示ボタンをクリックした時のみに用意された構造体内へ読み込むようにし、通常時は構造体内の数値を参照することとした。

この時、ピクチャーボックス内の座標は左上端が(0, 0)となり、x座標は右方向、y座標は下方向に向かって数値が大きくなる仕様なので、数値変換処理をグラフ描画前に記述した。

### 6-4. インストーラの作成

様々なPCで作成したソフトウェアを動作させるには、ソフトウェアの様々な依存関係を解消する必要がある。これを行う為にVisual studioによりインストーラの作成を行った。ListFileConverterソリューションにインストーラの為のプロジェクトを追加し、インストーラの作成を行った。

## 7. MPA-3システムの計数率特性評価

上記のソフトウェアにより、MPA-3リストデータファイルへのアクセスが簡略化された。このソフトウェアを使用し、MPA-3システムの計数率特性評価の実験・解析を行った。

実験では近畿大学原子力研究所が原子炉施設、トレーサー・加速器棟の両施設に所有する2台のMPA-3に対して行った。実験では、MPA-3システムのADCにパルサーで信号を入力し、ADCの数え落とし及びシステムの数え落としの計数率特性に関する調査を行った。実験にはBNC社のModel PB-5を使用し、ADCに10、50、100、200 [kHz]の信号

を入力した。

図6に入力信号の周波数変化に対するADCの数え落とし係数を示す。横軸は入力信号の波高、縦軸は後述する計数の数え落とし係数である。実線はトレーサー・加速器棟に設置されたMPA-3システムの実験結果、破線は原子炉施設に設置されたMPA-3システムの実験結果である。ADCは典型的な非マヒ型不感時間を持つ装置であり、不感時間より短い周期を持つ繰り返し信号を測定した場合、観測計数は実際の入力計数の整数分の1になる。(数え落とし係数) 図6において、逐次比較型ADCでは計測された計数率が全波高領域において変わらないのに対して、ウィルキンソン型ADCでは計測された計数率が波高に依存して整数分の1に落ちている様子が示されており、ウィルキンソン型ADCの不感時間が波高に比例して変化する様子を観測できた。

次にMPA-3システムの数え落としの計数率特性を図7に示す。横軸は入力信号の波高、縦軸はMPA-3システムによる計数の数え落とし率である。実線はトレーサー・加速器棟に設置されたMPA-3システムの実験結果、破線は原子炉施設に設置されたMPA-3システムの実験結果である。図7では、10、50、100 [kHz]において、トレーサー・加速器棟のシステムが原子炉施設のシステムに比べ数え落とし率が低い事が示されている。200 [kHz]において両システムの数え落とし率が反転しているが、これは原子炉施設のシステムではシステム上流側のADCがハードウェア的に数え落としすることにより、結果としてシステム全体がソフトウェア的に感じる計数率が下がり、見かけ上の数え落とし率が下がっていると考えられる。

以上の実験により、原子炉施設、トレーサー・加速器棟に設置されているMPA-3システムは10 [kHz]程度までなら問題なく動作すること、システムパフォーマンスはトレーサー・加速器棟に設置されているシステムの方が優れていることがわかった。



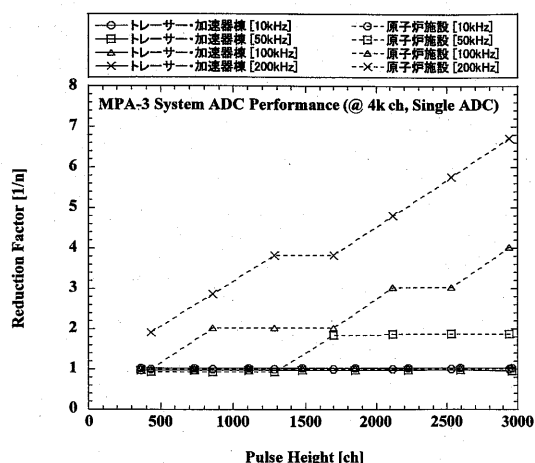


図6 MPA-3システムにより観測された計数率の減衰係数

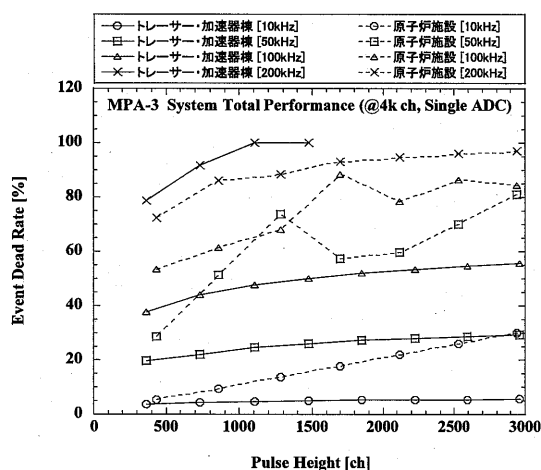


図7 MPA-3システムでの計数の数え落とし率

## 8. 原子炉からの漏洩中性子線の測定

更に原子炉施設に設置されているMPA-3システムを用い、原子炉からの漏洩中性子線を測定し、時系列解析を行った。

実験では測定検出器として $\text{BF}_3$ 比例計数管を用いた。 $\text{BF}_3$ 計数管は近畿大学原子炉 (UTR-KINKI) の上蓋上に設置され、原子炉からの漏洩中性子線を測定した。実験回路系は基本的な構成のものを使用した。

測定データの計数率の時間トレンドを図8に示す。横軸は時刻、縦軸は単位時間当たりの計数であ

る。図8では、 $\text{BF}_3$ 計数管の捉えた中性子数が、原子炉炉頂での中性子源 (NS) の動き、原子炉出力の変動を反映していることがわかる。

1.0[W] 自動臨界時の分散対平均比 (図8のグレー領域) を図9に示す。横軸はゲート時間幅、縦軸は分散対平均比である。図11では分散対平均比は1からほとんど動かず、UTR-KINKIの漏洩中性子線が未臨界度情報を失い、ほぼホワイトなポアソン線源と同様になっていることが見て取れる。

これらの解析結果は今回開発した ListFileConverter で解析したが、上記の時間トレンドからのデータ切り出し・再解析を含めた数値データを得るまでの解析にかかった時間は10分程度であった。

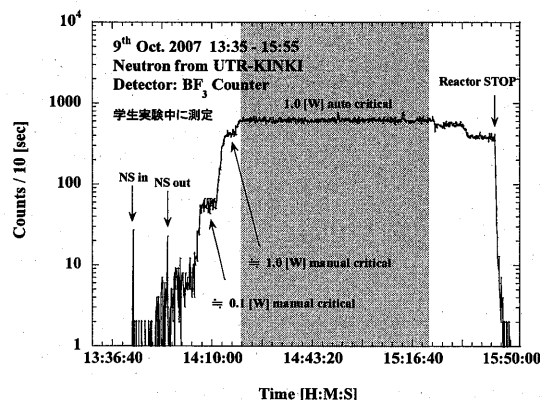


図8 原子炉炉頂部における相対中性子線量の時間トレンド

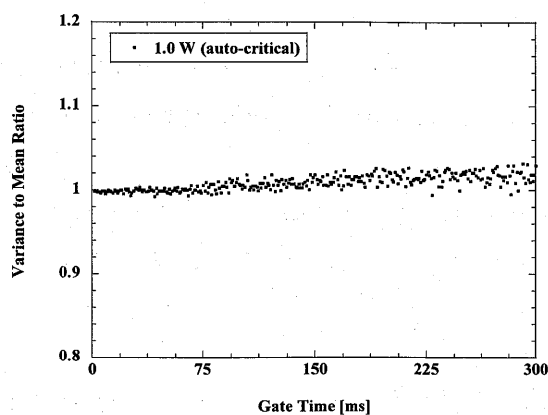


図9 原子炉からの漏洩中性子線の分散対平均比

## 9. さいごに

上記の解析ソフトウェア開発により、解析の手順を単純化し、測定終了から解析結果取得までの時間を大幅に短縮する事に成功した。今回の実験において取得したリストファイルサイズは最大250MB程度で、結果解析における所要時間は短いもので2分、長いものでも10分程度であった。今回のソフトウェア開発は放射線の時系列計測・解析を「誰にでも簡単に出来る操作」へと切り替えることに成功し、様々な応用実験への間口を広げたと考えている。

簡易な統合開発環境を導入することにより手順の単純化を図る事は、特に学生間における技術・手順の継承を容易にし、解析時間の短縮、解析上の誤りの削減、更に解析結果の物理的本質を考察する時間の確保に役立つと大いに期待している。

今後は、ListFileConverterの対応ADC数を増やし、研究室外にも公開していくことで、当研究所のMPA-3システムの利用促進を図っていく予定である。

最後にVisual studio 2005 / Visual C# 2005を用いた開発に関して非常に有用な参考文献を挙げる。<sup>2), 3), 4)</sup>

- 4) Visual C# 2005 逆引き大全500の極意：池谷京子、増田智明、国本温子著（2006、株式会社秀和システム）

## 参考文献

- 1) ユーザー・インタフェースの設計：Ben Shneiderman著、東基衛、井関治監訳（1987、日経BP社）
- 2) Microsoft Visual C# 2005 アプリケーション開発入門：植田政美（EISプランニング）、チーム・エムツー著（2006、日経BPソフトプレス）
- 3) 猫でもわかるC# プログラミング：桑井康孝著（2007、ソフトバンククリエイティブ株式会社）