

将来のスーパーコンピューティングへの挑戦

— 利用者の立場より —

産総研・池上 努



“Free lunch is over.”

以前は大型計算機に移植するだけで速くなった。

パソコン



スパコン



“Free lunch is over.”

今はもうそのままでは速くならない。

パソコン



スパコン



“Free lunch is over.”

今はもうそのままでは速くならない。

パソコン



スパコン



“Free lunch is over.”

今はもうそのままでは速くならない。

パソコン

スパコン



100万本のドリルを
使いこなす技術の開
発が必要



“Free lunch is over.”

今はもうそのままでは速くならない。

パソコン

スパコン



100万本のドリルを
使いこなす技術の開
発が必要

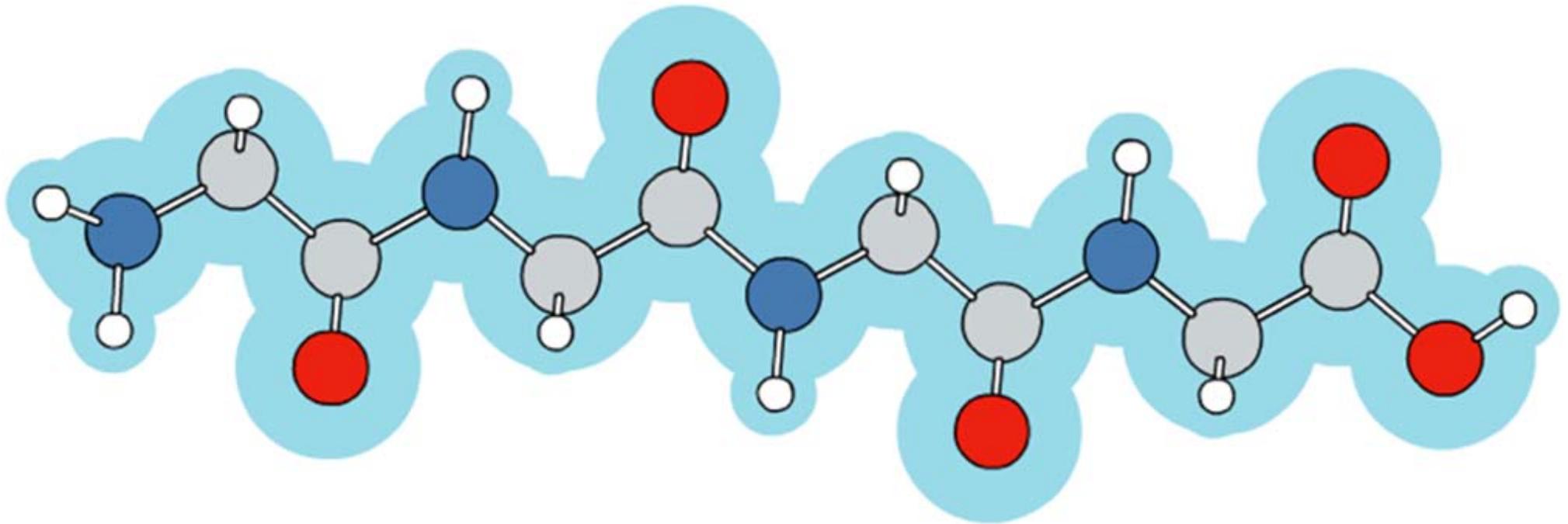


||

並列アルゴリズム開発への挑戦

フラグメント分子軌道法

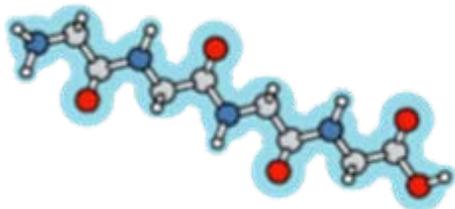
生体高分子の電子状態計算



従来法では、分子全体をまとめて解いている。

北浦(京大)・Fedorov(産総研)との共同研究

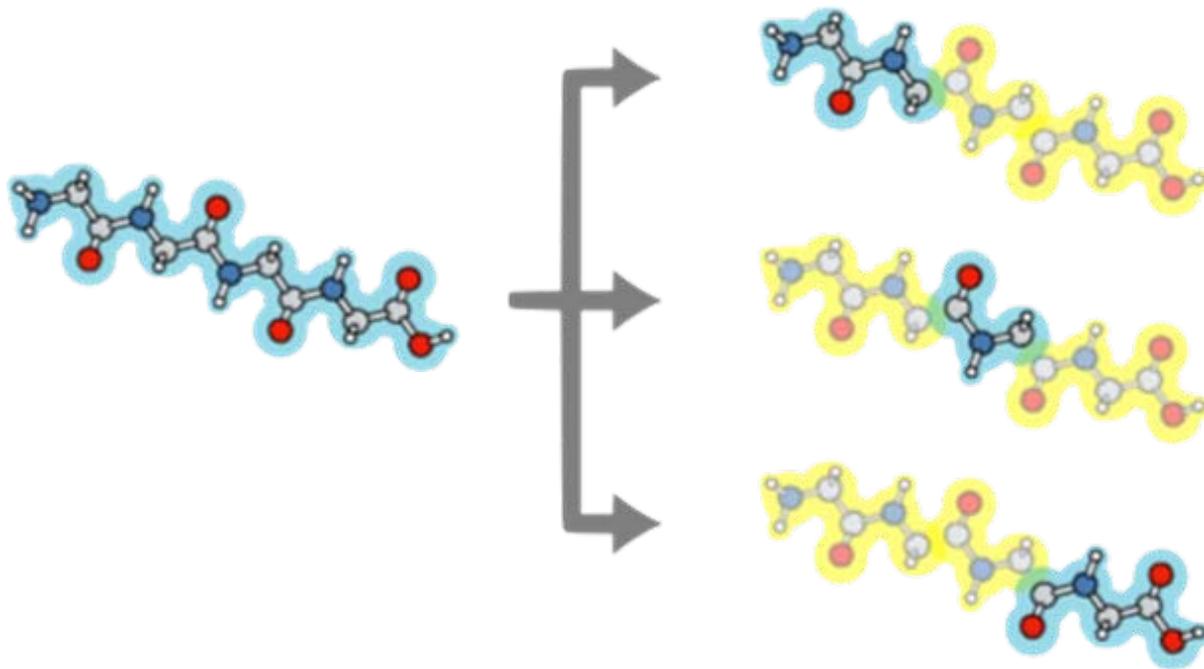
フラグメント分子軌道法



分子を断片化

北浦(京大)・Fedorov(産総研)との共同研究

フラグメント分子軌道法



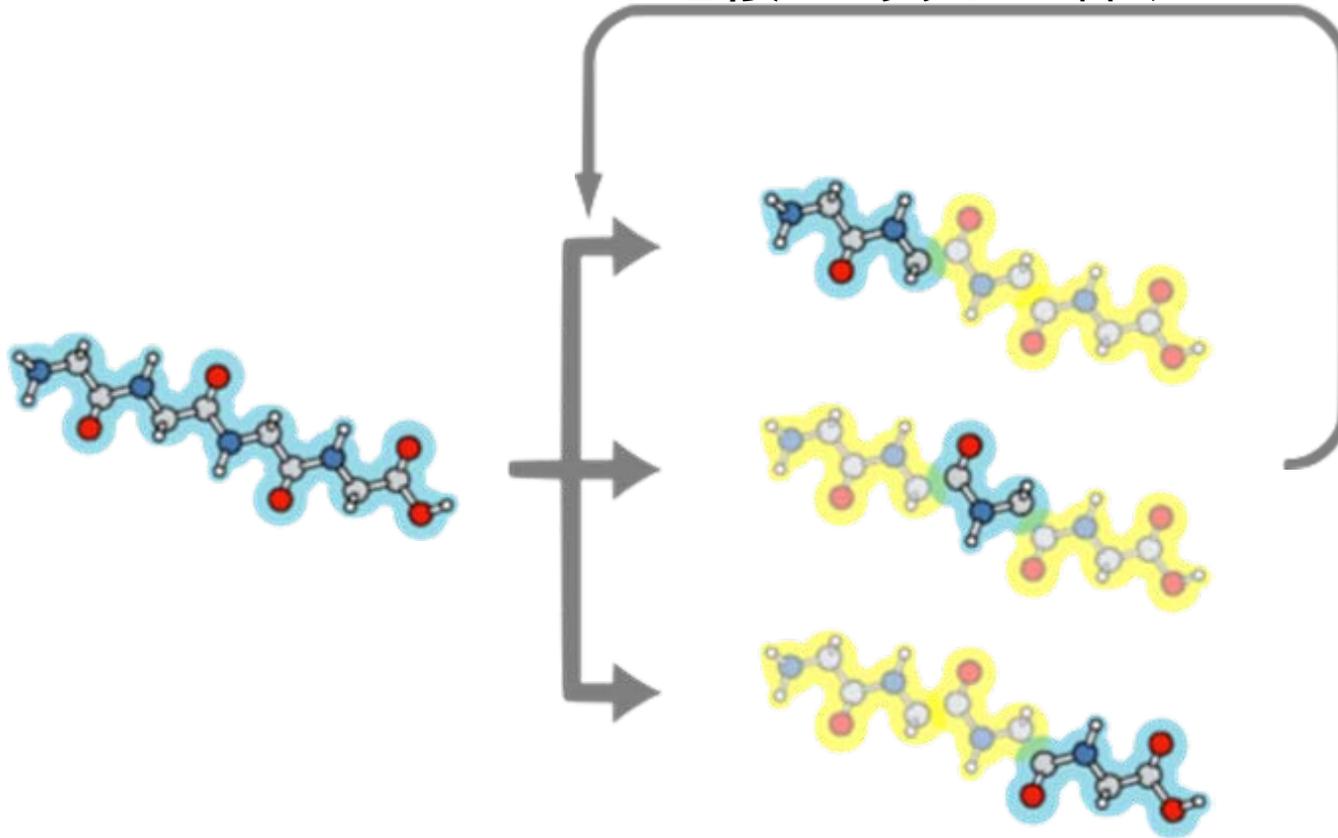
分子を断片化

断片毎に電子状態計算

北浦(京大)・Fedorov(産総研)との共同研究

フラグメント分子軌道法

辻褄があうまで繰り返し



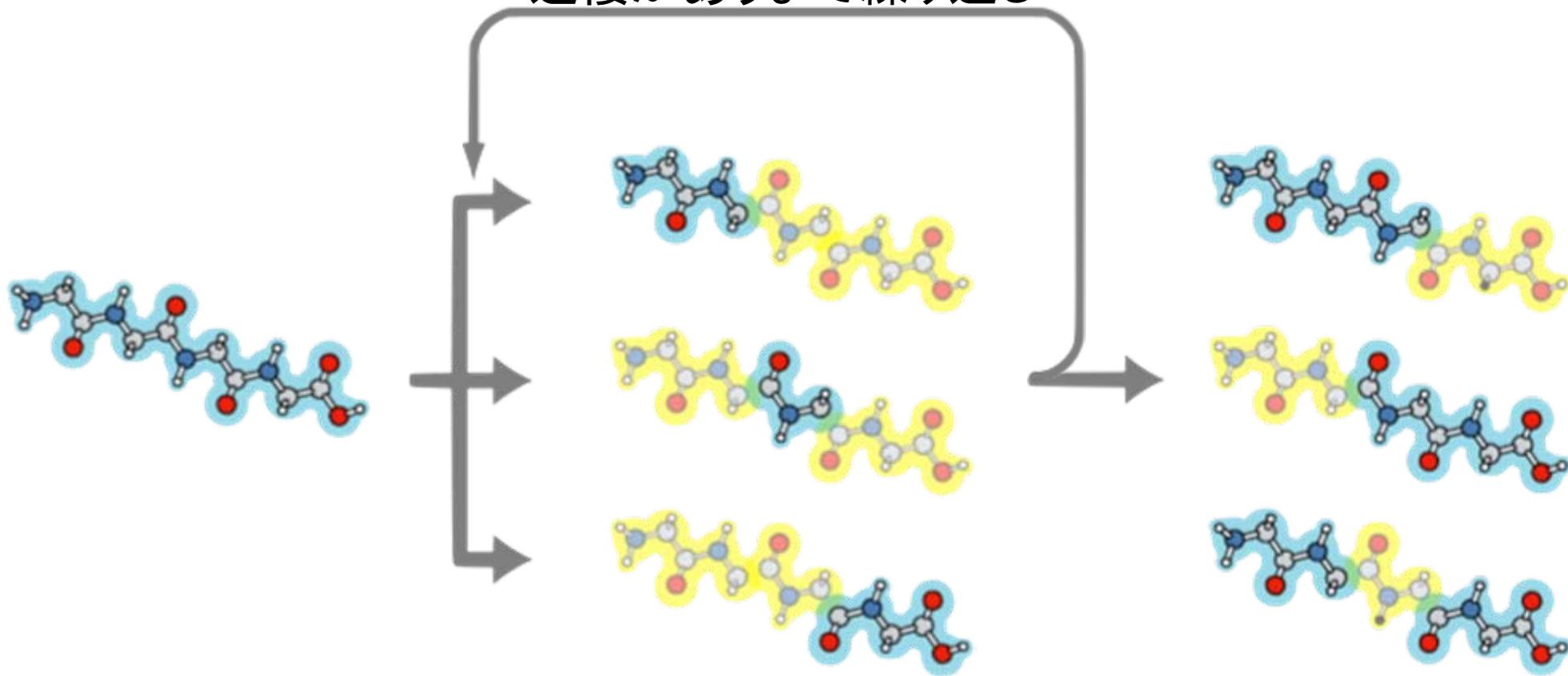
分子を断片化

断片毎に電子状態計算

北浦(京大)・Fedorov(産総研)との共同研究

フラグメント分子軌道法

辻褄があうまで繰り返し



分子を断片化

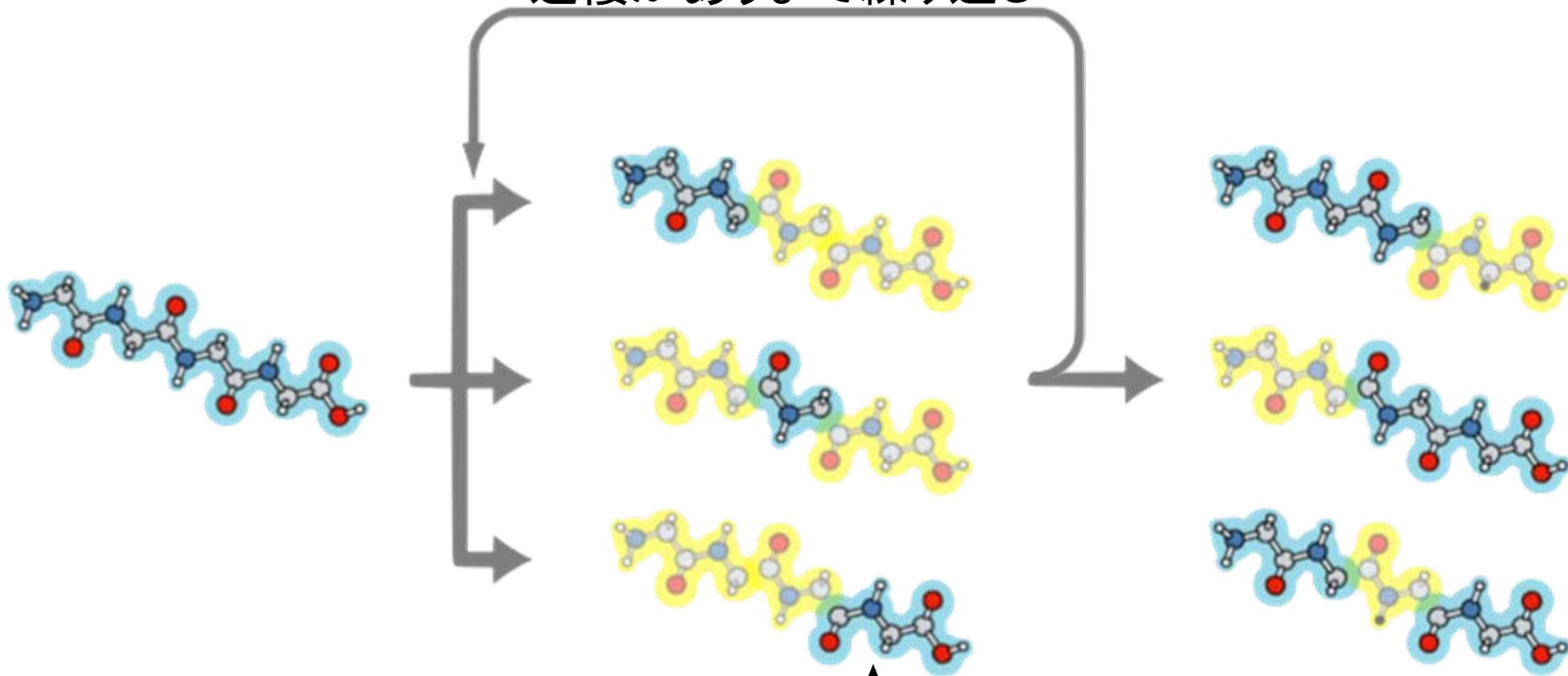
断片毎に電子状態計算

切断面の補修

北浦(京大)・Fedorov(産総研)との共同研究

フラグメント分子軌道法

辻褄があうまで繰り返し



分子を断片化

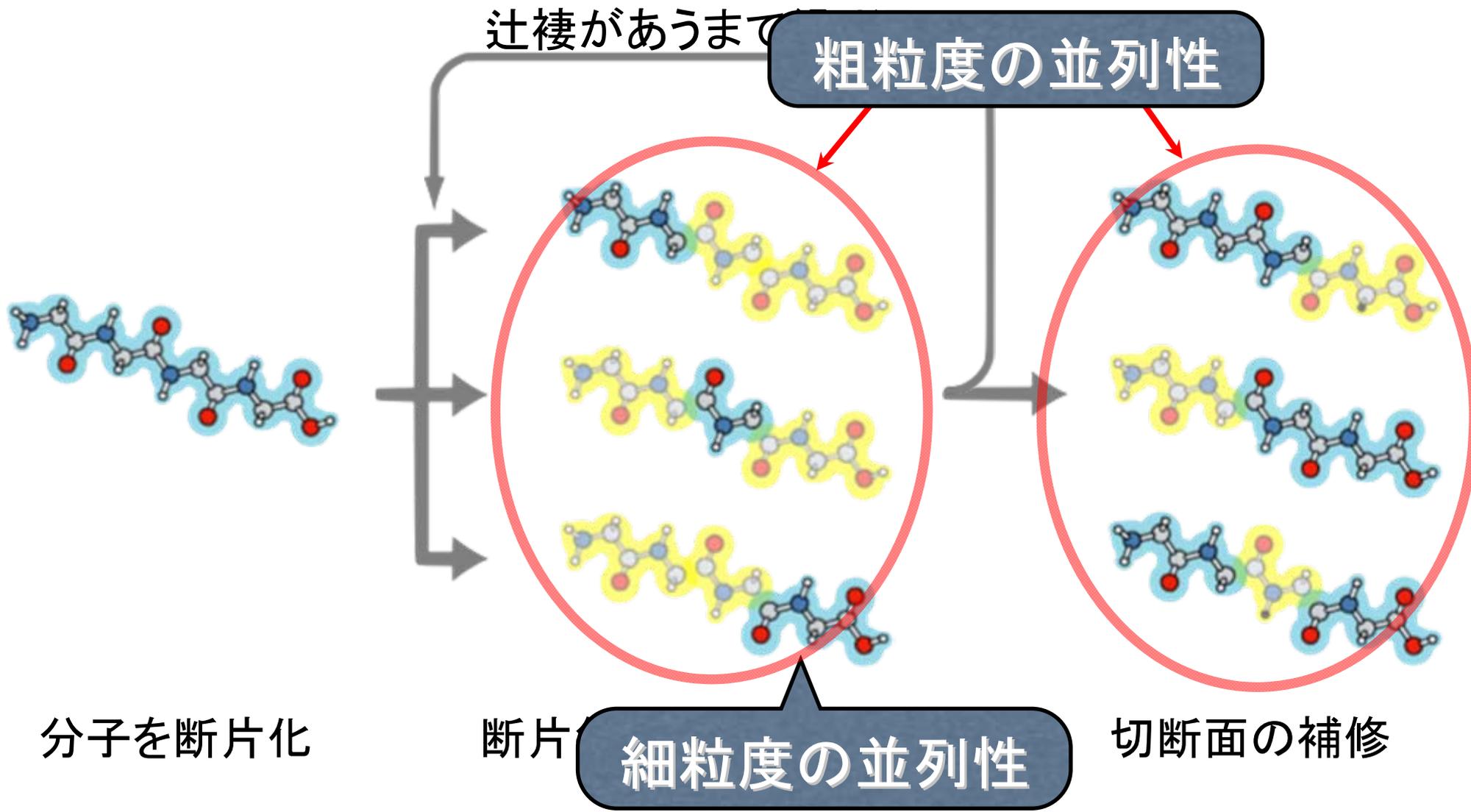
断片

細粒度の並列性

切断面の補修

北浦(京大)・Fedorov(産総研)との共同研究

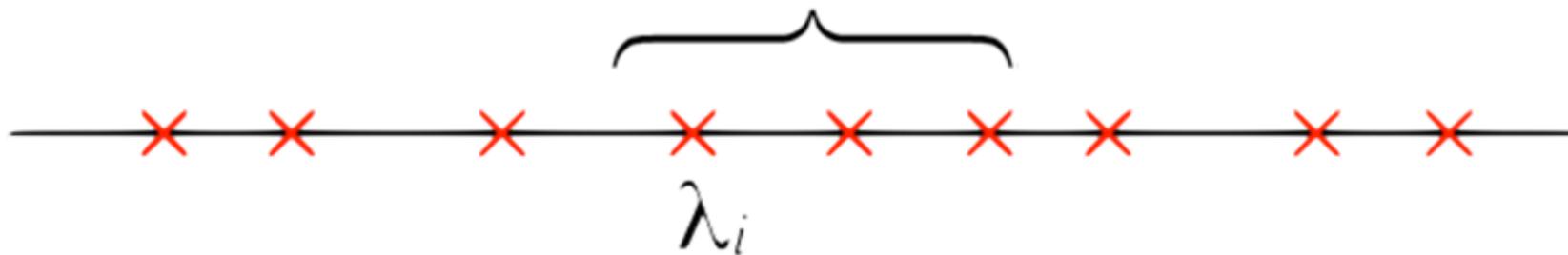
フラグメント分子軌道法



固有値の並列解法

$$(\lambda B - A) v = 0$$

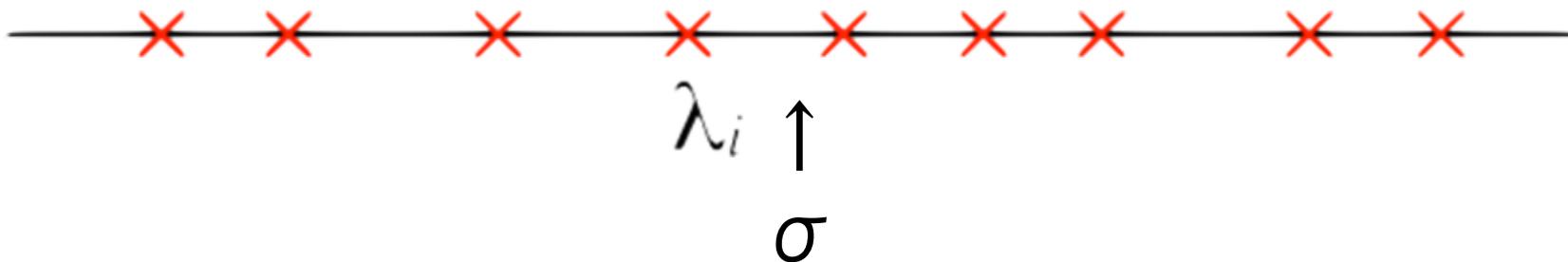
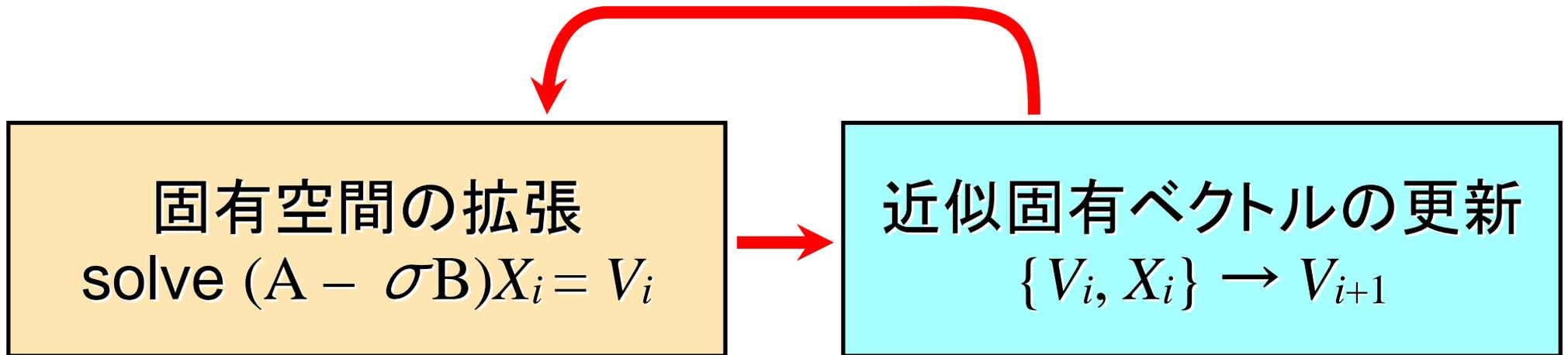
ここら辺の固有値
だけ欲しい



固有値の並列解法

$$(\lambda B - A) v = 0$$

従来型の解法:



固有値の並列解法

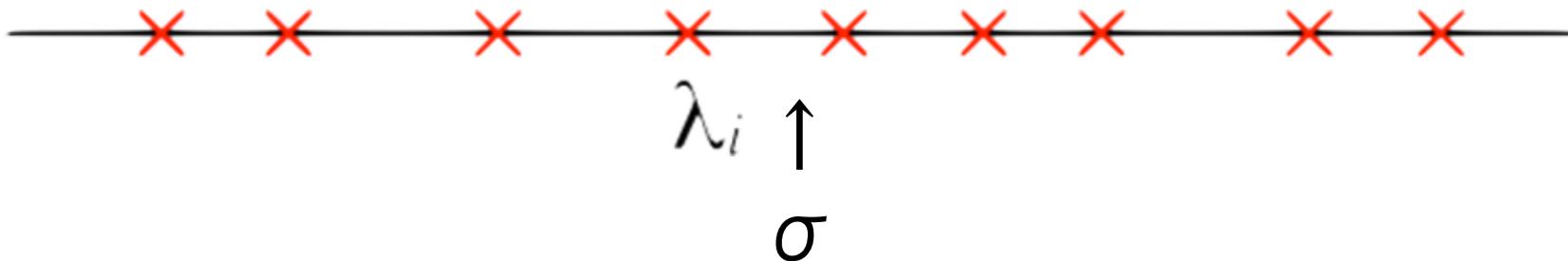
$$(\lambda B - A) v = 0$$

逐次的に更新

従来型の解法:

固有空間の拡張
solve $(A - \sigma B)X_i = V_i$

近似固有ベクトルの更新
 $\{V_i, X_i\} \rightarrow V_{i+1}$



固有値の並列解法

$$(\lambda B - A) v = 0$$

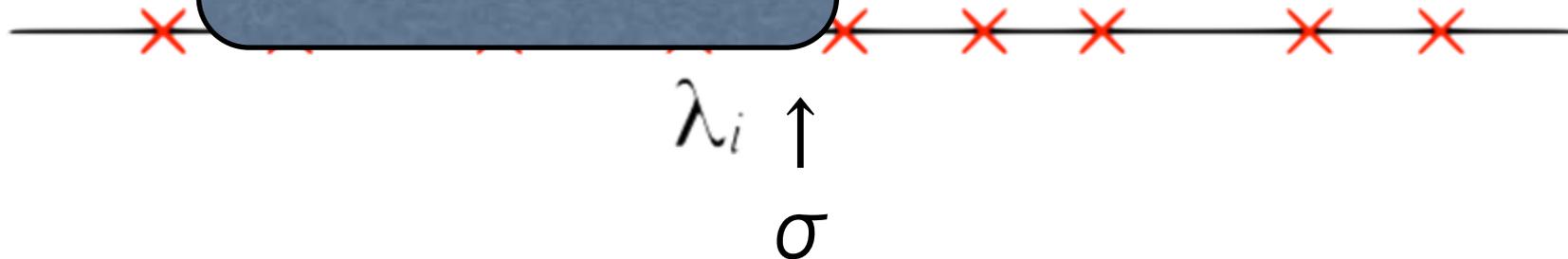
逐次的に更新

従来型の解法:

固有空間の拡張
solve $(A - \sigma B)X_i = V_i$

近似固有ベクトルの更新
 $\{V_i, X_i\} \rightarrow V_{i+1}$

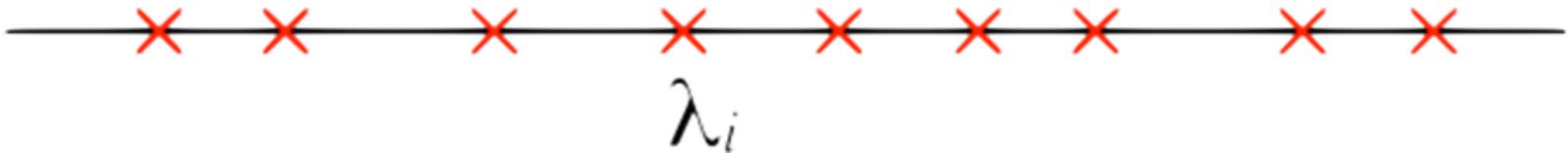
細粒度の並列性



固有値の並列解法

$$(\lambda B - A) v = 0$$

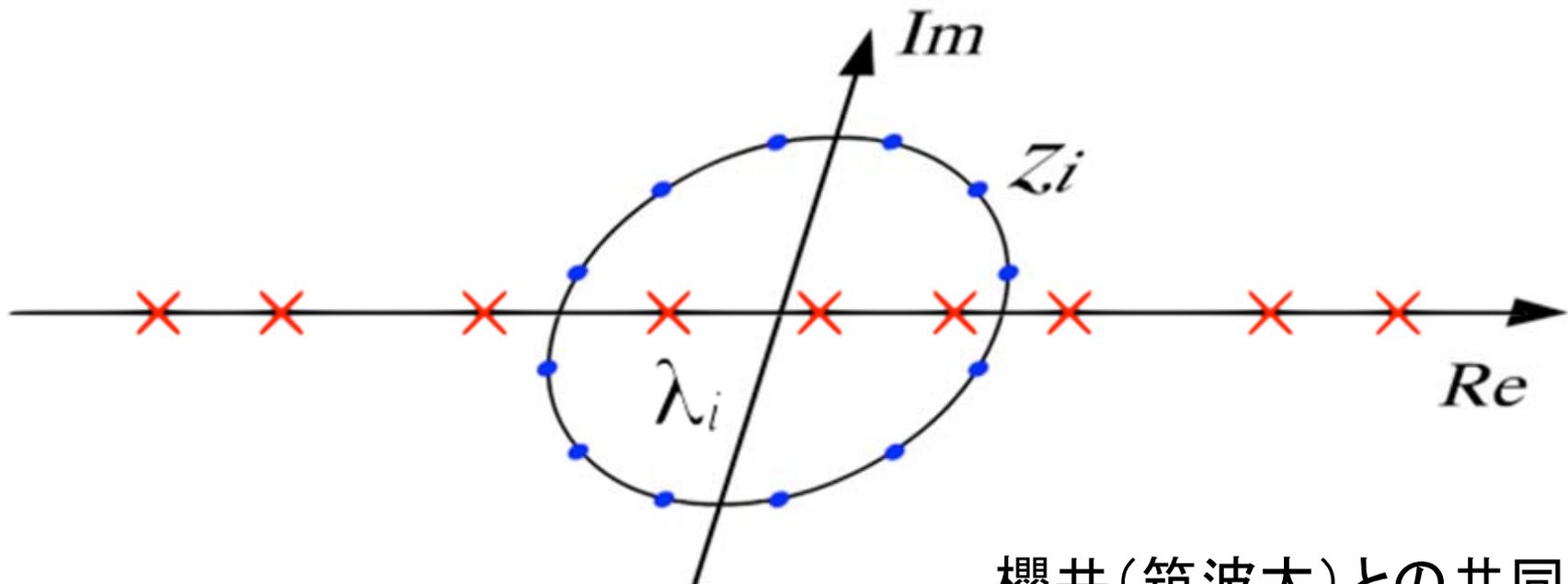
櫻井・杉浦法:



固有値の並列解法

$$(\lambda B - A) v = 0$$

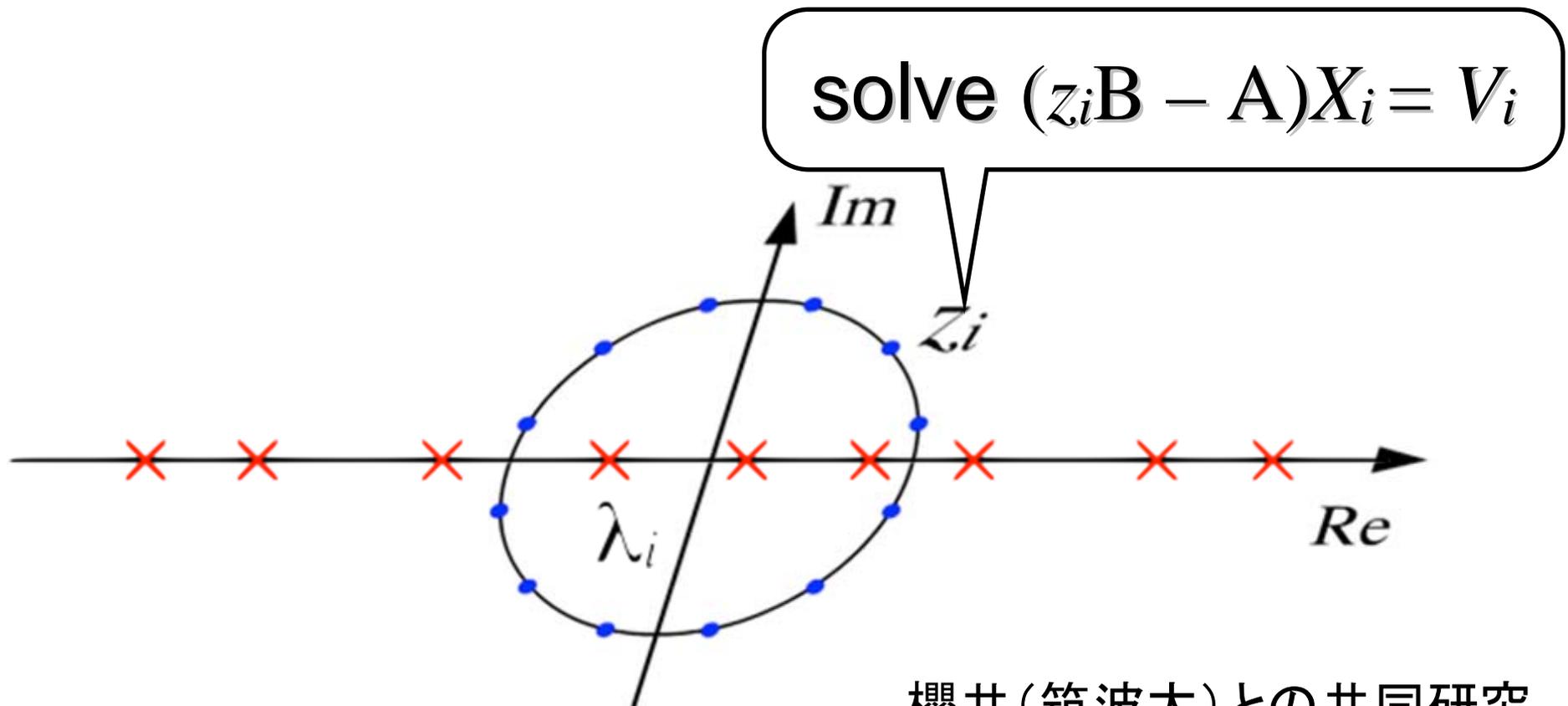
櫻井・杉浦法:



固有値の並列解法

$$(\lambda B - A) v = 0$$

櫻井・杉浦法:



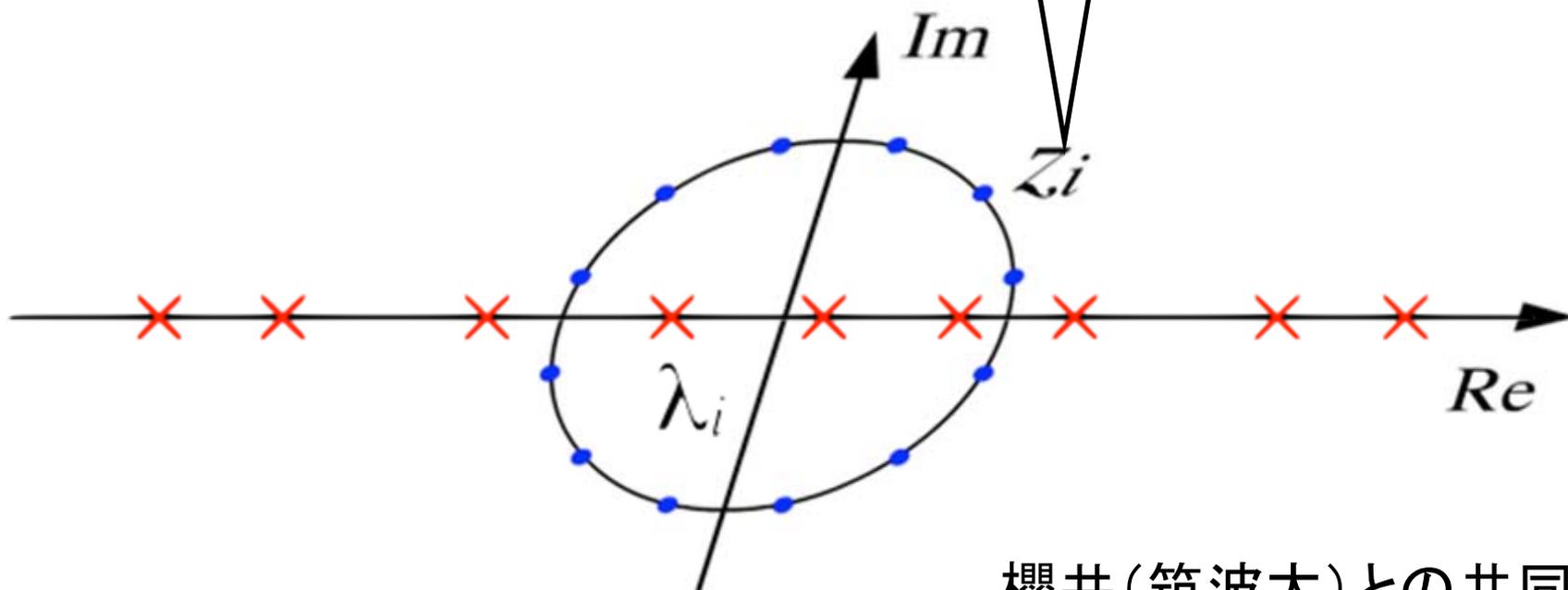
固有値の並列解法

$$(\lambda B - A)v = 0$$

櫻井・杉浦法:

細粒度の並列性

$$\text{solve } (z_i B - A)X_i = V_i$$



櫻井(筑波大)との共同研究

固有値の並列解法

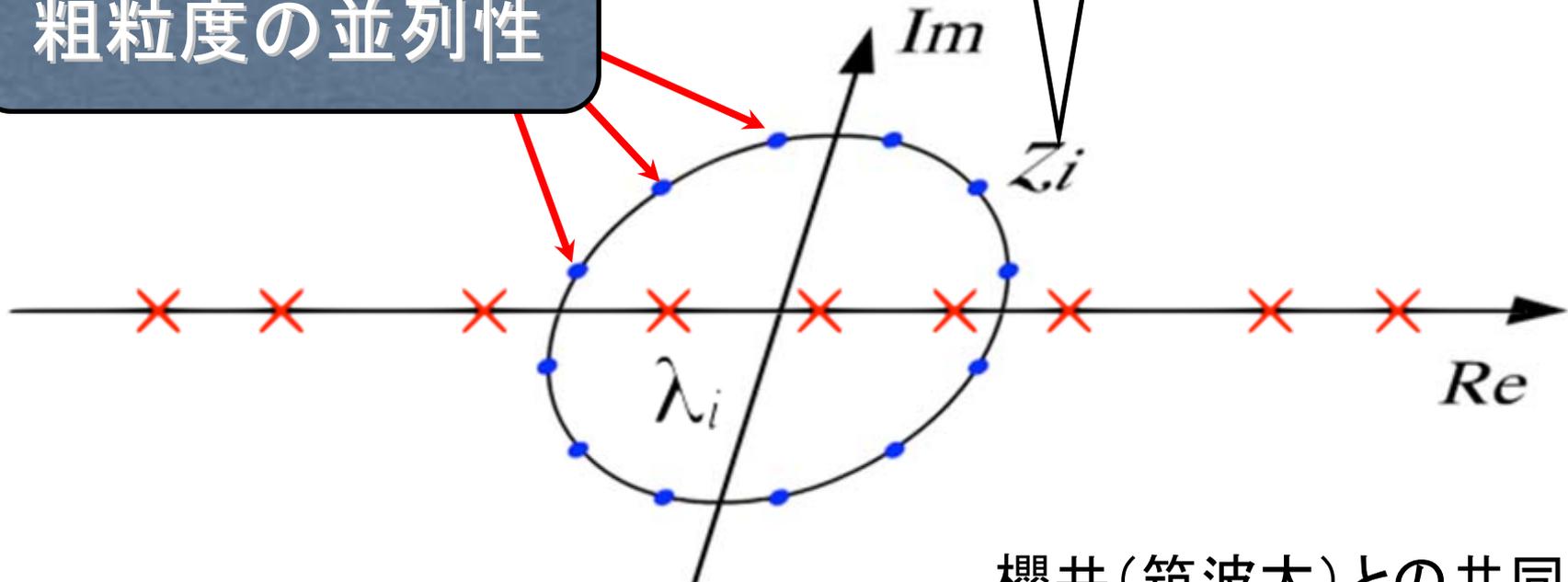
$$(\lambda B - A) v = 0$$

櫻井・杉浦法:

細粒度の並列性

$$\text{solve } (z_i B - A) X_i = V_i$$

粗粒度の並列性



将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
- 通信環境の再考
- 多様な計算資源の活用

将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
 - 電力効率から、GPU マシンになりそうだが...
 - “Free lunch” はあきらめるけどフルコースも勘弁
 - ソフトウェアの長寿命化
- 通信環境の再考
- 多様な計算資源の活用

将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
- 通信環境の再考
 - 現状既に、ノード内 \Leftrightarrow ノード間で階層的
 - ノード間通信が階層的でもかまわない
 - 100 ノード内で超低レイテンシみたいなのが欲しい
- 多様な計算資源の活用

将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
- 通信環境の再考
- 多様な計算資源の活用
 - ドリル 1000 本くらいなら、クラウドからも調達可能
 - 相互に使い分けできる環境が望ましい
 - 仮想化の導入も選択肢のひとつ

将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
 - 電力効率から、GPU マシンになりそうだが...
 - “Free lunch” はあきらめるけどフルコースも勘弁
 - ソフトウェアの長寿命化
- 通信環境の再考
 - 現状既に、ノード内 ⇔ ノード間で階層的
 - ノード間通信が階層的でもかまわない
 - 100 ノード内で超低レイテンシみたいなのが欲しい
- 多様な計算資源の活用
 - ドリル 1000 本くらいなら、クラウドからも調達可能
 - 相互に使い分けできる環境が望ましい
 - 仮想化の導入も選択肢のひとつ

将来の HPC 環境に望むこと

- 汎用性のあるプログラム環境
 - 電力効率から、GPU マシンになりそうだが...
 - “Free lunch” はあきらめるけどフルコースも勘弁
 - ソフトウェアの長寿命化

京速の実行効率 93% は福音
おめでとうございます

- 仮想化の導入も選択肢のひとつ