CME 345: MODEL REDUCTION

Methods for Nonlinear Systems

David Amsallem & Charbel Farhat
Stanford University
cfarhat@stanford.edu

Outline

- Full-Order Nonlinear Model:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}). \end{aligned}$$

  - $\mathbf{x} \in \mathbb{R}^n$: vector state variables
  - $\mathbf{u} \in \mathbb{R}^p$: vector of input variables, typically $p \ll n$
  - $\mathbf{y} \in \mathbb{R}^q$: vector output variables, typically $q \ll n$
- Usually, there is no closed form solution for $\mathbf{x}(t) = \phi(t, \mathbf{u}; t_0, \mathbf{x}_0)$

- Approximation of the state

$$\mathbf{x}(t) \approx \mathbf{V}\mathbf{x}_r(t)$$

- Resulting nonlinear ODE

$$\mathbf{V}\dot{\mathbf{x}}_r = \mathbf{f}(\mathbf{V}\mathbf{x}_r, \mathbf{u}) + \mathbf{r}(t)$$

- Enforce orthogonality of the residual $\mathbf{r}$ to a left basis $\mathbf{W}$:

$$\mathbf{W}^T\mathbf{V}\dot{\mathbf{x}}_r = \mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r, \mathbf{u})$$

- If $\mathbf{W}^T\mathbf{V}$ is nonsingular

$$\dot{\mathbf{x}}_r = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r, \mathbf{u})$$

- Reduced-order system of nonlinear ODEs

$$\dot{\mathbf{x}}_r = \mathbf{f}_r(\mathbf{x}_r, \mathbf{u})$$

$$\mathbf{y} = \mathbf{g}_r(\mathbf{x}_r, \mathbf{u})$$

  - vector of reduced coordinates: $\mathbf{x} \in \mathbb{R}^k$
  - reduced-order nonlinear dynamical operator:

$$\begin{array}{rcl} \mathbf{f}_r : \mathbb{R}^k \times \mathbb{R}^p & \rightarrow & \mathbb{R}^k \\ (\mathbf{x}_r, \mathbf{u}) & \longmapsto & (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r, \mathbf{u}) \end{array}$$

  - nonlinear output operator expressed in function of the reduced coordinates:

$$\begin{array}{rcl} \mathbf{g}_r : \mathbb{R}^k \times \mathbb{R}^p & \rightarrow & \mathbb{R}^q \\ (\mathbf{x}_r, \mathbf{u}) & \longmapsto & \mathbf{g}(\mathbf{V}\mathbf{x}_r, \mathbf{u}) \end{array}$$

- $\mathbf{x}_r^{(n)}$ denotes here an approximation to $\mathbf{x}_r(t^{(n)})$
- Example: forward Euler scheme

$$
\begin{aligned}
\mathbf{x}_r^{(n+1)} &= \mathbf{x}_r^{(n)} + (t^{(n+1)} - t^{(n)})\mathbf{f}_r(\mathbf{x}_r^{(n)}, \mathbf{u}(t^{(n)})) \\
&= \mathbf{x}_r^{(n)} + (t^{(n+1)} - t^{(n)})(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r^{(n)}, \mathbf{u}(t^{(n)})) \\
\mathbf{y}^{(n+1)} &= \mathbf{g}(\mathbf{V}\mathbf{x}_r^{(n+1)}, \mathbf{u}(t^{(n+1)}))
\end{aligned}
$$

- Requires evaluating the functions $\mathbf{f}$ and $\mathbf{g}$: cost scales with n

- $\mathbf{x}_r^{(n)}$ denotes here an approximation to $\mathbf{x}_r(t^{(n)})$
- Example: backward Euler scheme

$$\begin{aligned}
\mathbf{x}_r^{(n+1)} &= \mathbf{x}_r^{(n)} + (t^{(n+1)} - t^{(n)})\mathbf{f}_r(\mathbf{x}_r^{(n+1)}, \mathbf{u}(t^{(n+1)})) \\
&= \mathbf{x}_r^{(n)} + (t^{(n+1)} - t^{(n)})(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{x}_r^{(n+1)}, \mathbf{u}(t^{(n+1)})) \\
\mathbf{y}^{(n+1)} &= \mathbf{g}(\mathbf{V}\mathbf{x}_r^{(n+1)}, \mathbf{u}(t^{(n+1)}))
\end{aligned}$$

- Typically solved using Newton-Raphson method
- Requires evaluating the functions $\mathbf{f}$ and $\mathbf{g}$: cost scales with n
- Requires evaluating the Jacobian of $\mathbf{f}_r$ with respect to $\mathbf{x}_r$

$$\mathbf{A}_r(\mathbf{x}_r, \mathbf{u}) \equiv \frac{\partial \mathbf{f}_r}{\partial \mathbf{x}_r}(\mathbf{x}_r, \mathbf{u}) = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}(\mathbf{V}\mathbf{x}_r, \mathbf{u})\mathbf{V}$$

  where $\mathbf{A}(\mathbf{x}, \mathbf{u}) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u})$
- Forming the reduced Jacobian can be extremely expensive!

- Consider systems of the type

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{u})$$

- Linear systems are cheap as reduced-order operators of the type

$$\mathbf{A}_r = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V}$$

  can be formed once for all
- Idea: linearize $\mathbf{f}$ around an operating point $\mathbf{x}_1$

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_1) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_1)(\mathbf{x} - \mathbf{x}_1) = \mathbf{f}(\mathbf{x}_1) + \mathbf{A}(\mathbf{x}_1)(\mathbf{x} - \mathbf{x}_1)$$

- The resulting system is then linear in the state $\mathbf{x}$

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}_1)\mathbf{x} + [\mathbf{b}(\mathbf{u}) + (\mathbf{f}(\mathbf{x}_1) - \mathbf{A}(\mathbf{x}_1)\mathbf{x}_1)]$$

- Approximated full-order equations

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}_1)\mathbf{x} + [\mathbf{b}(\mathbf{u}) + (\mathbf{f}(\mathbf{x}_1) - \mathbf{A}(\mathbf{x}_1)\mathbf{x}_1)]$$

- Reduced-order equations

$$\dot{\mathbf{x}}_r = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}(\mathbf{x}_1)\mathbf{V}\mathbf{x}_r + (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T [\mathbf{b}(\mathbf{u}) + (\mathbf{f}(\mathbf{x}_1) - \mathbf{A}(\mathbf{x}_1)\mathbf{x}_1)]$$

- $\mathbf{A}_r = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T\mathbf{A}(\mathbf{x}_1)\mathbf{V} \in \mathbb{R}^{k\times k}$ and $\mathbf{B}_r = (\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T [(\mathbf{f}(\mathbf{x}_1) - \mathbf{A}(\mathbf{x}_1)\mathbf{x}_1)] \in \mathbb{R}^k$ can be pre-computed

- Idea: Linearize the nonlinear function locally in the state space
- Approximated full-order dynamical system

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^{s} \omega_i(\mathbf{x})(\mathbf{f}(\mathbf{x}_i) + \mathbf{A}_i(\mathbf{x} - \mathbf{x}_i)) + \mathbf{b}(\mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u})\end{aligned}$$

- $\omega_i, \; i = 1, \cdots, s$ are weights such that

$$\sum_{i=1}^{s} \omega_i(\mathbf{x}) = 1, \; \forall \mathbf{x} \in \mathcal{D}$$

- Reduced-order model obtained after Petrov-Galerkin projection:

$$\begin{aligned}\dot{\mathbf{x}}_r &= \sum_{i=1}^{s} \tilde{\omega}_i(\mathbf{x}_r)(\mathbf{W}^T\mathbf{V})^{-1}(\mathbf{W}^T\mathbf{f}(\mathbf{x}_i) + \mathbf{W}^T\mathbf{A}_i(\mathbf{V}\mathbf{x}_r - \mathbf{x}_i)) + \mathbf{W}^T\mathbf{b}(\mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{V}\mathbf{x}_r, \mathbf{u})\end{aligned}$$

with

$$\sum_{i=1}^{s} \tilde{\omega}_i(\mathbf{x}_r) = 1, \ \forall \mathbf{x}_r \in \mathcal{D}_r.$$

- Equivalently:

$$\begin{aligned}\dot{\mathbf{x}}_r &= \left(\sum_{i=1}^{s} \tilde{\omega}_i(\mathbf{x}_r)\mathbf{A}_{ri}\right)\mathbf{x}_r + \left(\sum_{i=1}^{s} \tilde{\omega}_i(\mathbf{x}_r)(\mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T(\mathbf{f}(\mathbf{x}_i) - \mathbf{A}_i\mathbf{x}_i)\right) \\ &\quad + \mathbf{b}_r(\mathbf{u})\end{aligned}$$

A complete model reduction method should provide algorithms for

- Selection of the linearization points $\{\mathbf{x}_i\}_{i=1}^{s}$
- Selection of the reduced-order bases $\mathbf{V}$ and $\mathbf{W}$
- Determination of the weights $\{\omega_i(\mathbf{x}_r)\}_{i=1}^{s}, \; \forall \mathbf{x}_r \in \mathcal{D}_r$

- Linear approximations of the NL function only valid in a neighborhood of each $\mathbf{x}_i$.
- Can't cover the whole state-space $\mathcal{D}$ with local approximation
- Use trajectories (off-line phase) of the FOM to chose them
- Select linearization point if sufficiently far away from previous one
- Trajectory Piecewise Linear (TPWL) ROM (Rewisenski and White 2001).

Possible methods include

- If the input function is linear in $\mathbf{u}$, construction of Krylov subspaces $\mathcal{K}_i = \mathcal{K}(\mathbf{A}_i^{-1}, \mathbf{A}_i^{-1}\mathbf{B}) = \text{span}(\mathbf{V}_i)$ at each linearization point $\mathbf{x}_i$ and assembly of a global basis (Rewisenski and White 2001, Bond and Daniel 2004)
$$\mathbf{V} = [\mathbf{V}_1, \cdots, \mathbf{V}_s]$$
- Ad-hoc methods (Balanced truncation, POD...)

Used in the interpolation of the local reduced-order models to characterize the distance of the new point $\mathbf{x}_r$ to the precomputed ones $\{\mathbf{x}_{r,i}\}_{i=1}^{s}$.

- Example (Bond and Daniel 2004):

$$\tilde{\omega}_i(\mathbf{x}_r) = \frac{\exp\left(-\frac{\beta d_i^2}{m^2}\right)}{\sum_{j=1}^{s} \exp\left(-\frac{\beta d_j^2}{m^2}\right)}$$

  where β is a constant, $d_i = \|\mathbf{x}_r - \mathbf{x}_{r,i}\|_2$ and $m = \min_{j=1}^{s} d_j$.

- Other expressions suggested in Tiwary and Rutenbar (2005), Dong and Roychowdhury (2005).

From Rewienski and White (2006)

- A posteriori error estimator available when $\mathbf{f}$ is negative monotone.
- Stability guarantee under assumptions on $\mathbf{f}$ and the choice of $\mathbf{V}$ and the weights $\{\tilde{\omega}_i(\mathbf{x})\}_{i=1}^{s}$.
- Passivity preservation under similar assumptions.

Strengths

- Does not scale with the size of the FOM n

Weaknesses

Strengths

- Does not scale with the size of the FOM n

Weaknesses

- Choice of trajectories of the FOM is essential

Strengths

- Does not scale with the size of the FOM n
- Global basis

Weaknesses

- Choice of trajectories of the FOM is essential
- Global basis

Strengths

- Does not scale with the size of the FOM n
- Global basis

Weaknesses

- Choice of trajectories of the FOM is essential
- Global basis

Strengths

- Does not scale with the size of the FOM n
- Global basis
- Availability of Jacobians

Weaknesses

- Choice of trajectories of the FOM is essential
- Global basis
- Availability of Jacobians

Strengths

- Does not scale with the size of the FOM n
- Global basis
- Availability of Jacobians
- Not too much code intrusive

Weaknesses

- Choice of trajectories of the FOM is essential
- Global basis
- Availability of Jacobians

Strengths

- Does not scale with the size of the FOM n
- Global basis
- Availability of Jacobians
- Not too much code intrusive

Weaknesses

- Choice of trajectories of the FOM is essential
- Global basis
- Availability of Jacobians
- Parameters to adjust

- Piecewise Polynomial Representations (Roychowdhury 1999, Chen 1999, Philips 2003, Li and Pileggi 2003)
- Kernel-based Representations (Philips et al. 2003)
- Parametric Nonlinear ROMs (Bond and Daniel 2004)
- Localized Linear Reductions (Local bases) (Tiwary and Rutenbar 2006)
- ManiMOR (Gu and Roychowdhury 2008)

- First applied to face recognition (Emerson and Sirovich, "Karhunen-Loeve Procedure for Gappy Data" 1996)

- Approximation of the nonlinear function $\mathbf{f}$ in

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{u})$$

- The evaluation of all the entries in the vector $\mathbf{f}(\cdot)$ is expensive (scales with n)
- Only a small subset of these entries will be evaluated
- The other entries will be reconstructed either by interpolation or a least-squares strategy using a pre-computed reduced-order basis
- The state-space is still reduced by any preferred model reduction method (by POD for instance)

A complete model reduction method should then provide algorithms for

- Selecting the evaluation indices $\mathcal{I} = \{i_1, \cdots, i_{N_i}\}$
- Selecting a reduced-order bases $\mathbf{V_f}$ for the nonlinear function
- Reconstructing the complete approximated nonlinear function vector $\hat{\mathbf{f}}(\cdot)$

- A POD basis for $\mathbf{f}(\cdot)$ is built:
  1. Snapshots for the nonlinear function are collected from a transient simulation
  $$\mathbf{F} = [\mathbf{f}(\mathbf{x}(t_1)), \cdots, \mathbf{f}(\mathbf{x}(t_{N_s}))]$$
  2. A singular value decomposition is computed
  $$\mathbf{F} = \mathbf{U_f} \boldsymbol{\Sigma}_\mathbf{f} \mathbf{Z}_\mathbf{f}^T$$
  3. The basis is truncated and the $k_\mathbf{f}$ first vectors retained
  $$\mathbf{V_f} = \mathbf{U}_{k_\mathbf{f}}$$

- Assume N_i indices have been chosen

$$\mathcal{I} = \{i_1, \cdots, i_{N_i}\}$$

- The choice of indices will be specified later
- Consider the n-by-N_i matrix

$$\mathbf{P} = \left[\mathbf{e}_{i_1}, \cdots, \mathbf{e}_{i_{N_i}}\right]$$

- At each time t, for a given value of the state $\mathbf{x}(t) = \mathbf{V}\mathbf{x}_r(t)$, the entries in the function $\mathbf{f}$ corresponding to those indices will be evaluated

$$\mathbf{P}^T\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} f_{i_1}(\mathbf{x}(t)) \\ \vdots \\ f_{i_{N_i}}(\mathbf{x}(t)) \end{bmatrix}$$

- This is cheap if $N_i \ll n$
- Usually only a subset of the entries in $\mathbf{x}(t)$ will be used to construct that vector

- Case where $N_i = k_{\mathbf{f}}$: interpolation
  - Idea: $\hat{f}_{i_j}(\mathbf{x}) = f_{i_j}(\mathbf{x}),\ \forall j = 1, \cdots, N_i$
  - This means that
$$\mathbf{P}^T \hat{\mathbf{f}}(\mathbf{x}(t)) = \mathbf{P}^T \mathbf{f}(\mathbf{x}(t))$$
  - Remember that $\hat{\mathbf{f}}(\cdot)$ belongs to the span of the vectors in $\mathbf{V}_{\mathbf{f}}$, that is
$$\hat{\mathbf{f}}(\cdot) = \mathbf{V}_{\mathbf{f}} \mathbf{f}_r(\cdot)$$
  - Then
$$\mathbf{P}^T \mathbf{V}_{\mathbf{f}} \mathbf{f}_r(\cdot) = \mathbf{P}^T \mathbf{f}(\mathbf{x}(t))$$
  - Assuming $\mathbf{P}^T \mathbf{V}_{\mathbf{f}}$ is nonsingular
$$\mathbf{f}_r(\cdot) = (\mathbf{P}^T \mathbf{V}_{\mathbf{f}})^{-1} \mathbf{P}^T \mathbf{f}(\mathbf{x}(t))$$
  - In terms of $\hat{\mathbf{f}}(\cdot)$:
$$\hat{\mathbf{f}}(\cdot) = \mathbf{V}_{\mathbf{f}} (\mathbf{P}^T \mathbf{V}_{\mathbf{f}})^{-1} \mathbf{P}^T \mathbf{f}(\mathbf{x}(t)) = \mathbf{\Pi}_{\mathbf{V}_{\mathbf{f}}, \mathbf{P}} \mathbf{f}(\mathbf{x}(t))$$
  - This results in an oblique projection of the full nonlinear vector

- Case where $N_i > k_{\mathbf{f}}$: least-squares reconstruction
  - Idea: $\hat{f}_{i_j}(\mathbf{x}) \approx f_{i_j}(\mathbf{x}),\ \forall j = 1, \cdots, N_i$ in the least squares sense
  - Idea: minimize

$$\min_{\mathbf{f}_r(\cdot)} \|\mathbf{P}^T \mathbf{V_f} \mathbf{f}_r(\cdot) - \mathbf{P}^T \mathbf{f}(\mathbf{x}(t))\|_2$$

  - Notice that $\mathbf{M} = \mathbf{P}^T \mathbf{V_f} \in \mathbb{R}^{N_i \times k_{\mathbf{f}}}$ is a skinny matrix
  - One can compute its singular value decomposition

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T$$

  - The left inverse of $\mathbf{M}$ is then defined as

$$\mathbf{M}^\dagger = \mathbf{Z}\mathbf{\Sigma}^\dagger \mathbf{U}^T$$

    where $\mathbf{\Sigma}^\dagger = \text{diag}(\frac{1}{\sigma_1}, \cdots, \frac{1}{\sigma_r}, 0, \cdots, 0)$ if $\mathbf{\Sigma} = \text{diag}(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0)$ with $\sigma_1 \geq \cdots \sigma_r > 0$
  - Then

$$\hat{\mathbf{f}}(\cdot) = \mathbf{V_f}(\mathbf{P}^T \mathbf{V_f})^\dagger \mathbf{P}^T \mathbf{f}(\mathbf{x}(t))$$

- This selection takes place after the vectors $[\mathbf{v}_1^f, \cdots, \mathbf{v}_{k_\mathbf{f}}^f]$ has been computed by POD
- Greedy algorithm (Chaturantabut 2010):

1: $[m, i_1] = \max\{|\mathbf{v}_1^f|\}$
2: $\mathbf{V}_\mathbf{f} = [\mathbf{v}_1^f]$, $\mathbf{P} = [\mathbf{e}_{i_1}]$
3: **for** $l = 2 : k_\mathbf{f}$ **do**
4: Solve $\mathbf{P}^T\mathbf{V}_\mathbf{f}\mathbf{c} = \mathbf{P}^T\mathbf{v}_l^f$ for $\mathbf{c}$
5: $\mathbf{r} = \mathbf{v}_l^f - \mathbf{V}_\mathbf{f}\mathbf{c}$
6: $[m, i_l] = \max\{|\mathbf{r}|\}$
7: $\mathbf{V}_f = [\mathbf{V}_\mathbf{f}, \mathbf{v}_l^f]$, $\mathbf{P} = [\mathbf{P}, \quad \mathbf{e}_{i_l}]$
8: **end for**

- The inviscid Burgers equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{1}{2}\frac{\partial (U(x,t))^2}{\partial x} = g(x)$$

where the source term is

$$g(x) = 0.02exp(0.02x)$$

the initial condition is

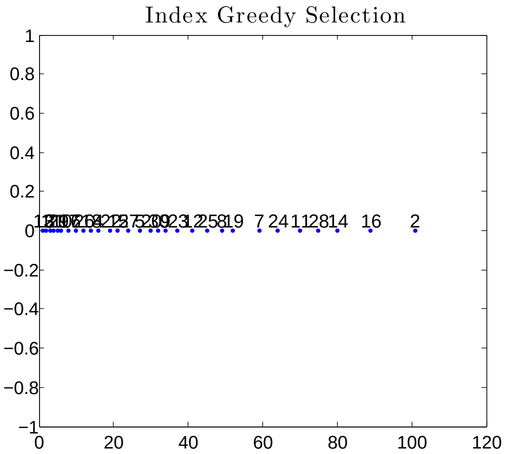
$$U(x,0) = 1$$

and the inlet boundary condition is

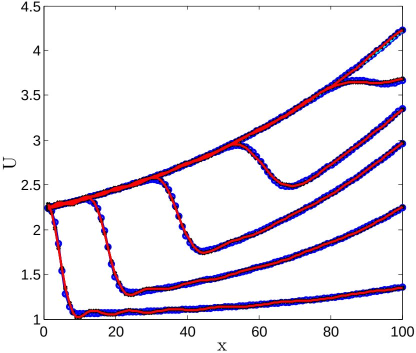
$$U(0,t) = \sqrt{5}$$

- Discretized by a Finite Volume Method (Godunov)

- Results of the greedy algorithm

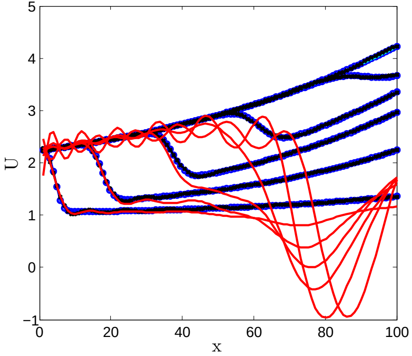


- $k = 15$, $k_{\mathbf{f}} = 40$, $N_i = 40$



- Similar results with $N_i > 40$ (least squares reconstruction)

- $k = 15$, $k_{\mathbf{f}} = 30$, $N_i = 80$



- Similar results with $N_i = 100$ (no gaps)
- $k_{\mathbf{f}}$ is too small

Method	CPU Time	Relative Error
Full-Order Model	2180 s	–
ROM (Galerkin)	1866 s	1.95%
ROM (Gappy approximation)	183 s	2.17%

Table: Comparison of CPU Timings and Accuracy of the Methods

- Balanced Truncation for NL systems
- Volterra series
- Methods for Linear Time Variant and Periodic Systems