Solving Nonlinear Functional Equations Using Adaptive Grids

Alexis Akira Toda

Emory University

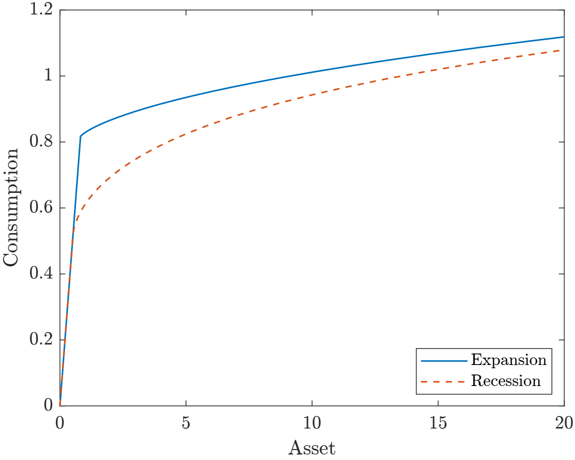
CEF Conference
July 1, 2026

Dynamic programming on a grid

- Many dynamic economic models solved by value/policy function iteration on finite grid
- Uniform grids often perform poorly because curvature differs sharply across state space
- Heuristic alternatives (e.g., exponential grid) useful but model-specific
- Goal: choose grid points automatically while solving functional equation

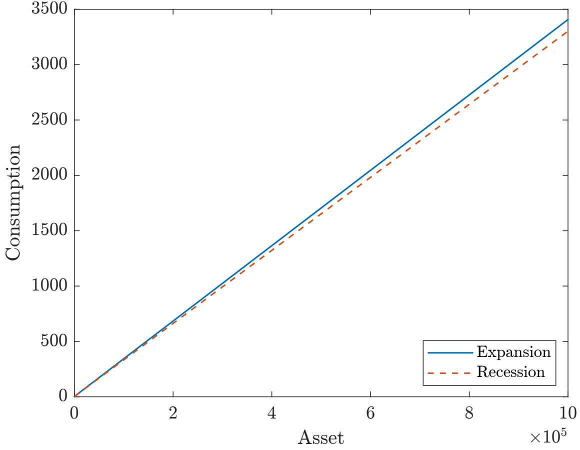
Example: optimal savings problem

- Consumption function highly concave at small asset levels and may have kink



Example: optimal savings problem

- Consumption function close to linear at large asset levels (Ma and Toda, 2021, 2022)



This paper

- Proposes adaptive grid for dynamic programming and related nonlinear functional equations
- Key idea: spline interpolation has derivative-based error estimates
- Use those estimates to place grid points where interpolation is hardest
- Grid updated in closed form, can be embedded inside value or policy iteration (no additional computational cost)

Related literature

- Free-knot and adaptive spline approximation: Meinardus et al. (1989)
- Adaptive grids in dynamic programming: Grüne and Semmler (2004) and Rust (2008)
- Accuracy evaluation using Euler equation residuals: Judd (1992), Judd and Guu (1997), and Santos (2000)

Problem statement

- Unknown function: $f : [a, b] \to [-\infty, \infty]$
- Typical dynamic-programming shape: $f' > 0$, $f'' < 0$, high curvature near boundary
- Computation gives values $\{(x_n, y_n)\}_{n=1}^{N}$ with $y_n \approx f(x_n)$

Question

Given current grid and values, how should we update grid and interpolate f efficiently?

Cubic spline interpolation

- On interval $I_n = [x_{n-1}, x_n]$, approximate f by cubic function

$$s(x) = a_n(x - x_{n-1})^3 + b_n(x - x_{n-1})^2 + c_n(x - x_{n-1}) + d_n$$

- s agrees with f at grid points
- $s \in C^2[a, b]$
- Two additional end conditions determine spline, e.g. natural or not-a-knot

Spline error estimates

- Let $\Delta x_n = x_n - x_{n-1}$ and $h = \max_n \Delta x_n$
- For $f \in C^4[a, b]$, Hall and Meyer (1976, Theorem 5) implies

$$\left\| f^{(r)} - s^{(r)} \right\| \leq C_r \left\| f^{(4)} \right\| h^{4-r}, \qquad r = 0, 1, 2, 3$$

- For $f \in C^3[a, b]$, Powell (1981, Theorem 20.3) implies

$$\| f - s \| \leq \frac{24}{(4-r)!} \left\| f^{(r)} \right\| (h/2)^r, \qquad r = 1, 2, 3$$

- Interpretation: smaller interval most valuable where derivatives are large

Local approximation heuristic

- Let interval $I_n = [x_{n-1}, x_n]$, interval length $y_n = x_n - x_{n-1}$, and

$$D_n = \left\| s^{(r)} \right\|_{I_n}$$

- Local error estimate suggests

$$\|f - s\|_{I_n} \approx \frac{24}{2^r(4-r)!} D_n y_n^r$$

- Thus, maximum error approximated by

$$E(y_1, \ldots, y_N) = \frac{24}{2^r(4-r)!} \max_n D_n y_n^r$$

Optimal interval lengths

- Choose interval lengths $y_n \geq 0$ subject to

$$\sum_{n=1}^{N} y_n = b - a$$

- Minmax solution equalizes local error estimate:

$$D_1 y_1^r = \cdots = D_N y_N^r$$

- Hence

$$y_n = \frac{(b-a) D_n^{-1/r}}{\sum_{k=1}^{N} D_k^{-1/r}}$$

  - Large $D_n \implies$ small interval
  - Low curvature $\implies$ sparse grid

Closed-form grid update

- New grid points are

$$x_n = a + \frac{\sum_{k=1}^{n} D_k^{-1/r}}{\sum_{k=1}^{N} D_k^{-1/r}}(b-a), \qquad n = 0, \dots, N$$

- Approximate D_n from spline derivatives:

$$s^{(1)}(x) = 3a_n(x - x_{n-1})^2 + 2b_n(x - x_{n-1}) + c_n,$$
$$s^{(2)}(x) = 6a_n(x - x_{n-1}) + 2b_n,$$
$$s^{(3)}(x) = 6a_n,$$

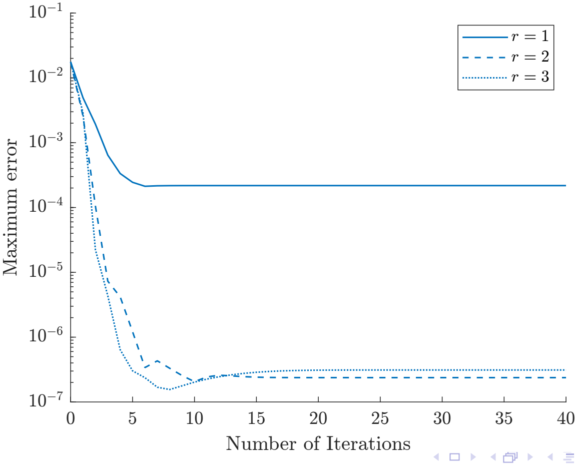
$$D_n = \max\left\{\left|s^{(r)}(x_{n-1})\right|, \left|s^{(r)}(x_n)\right|\right\}$$

Adaptive spline grid algorithm

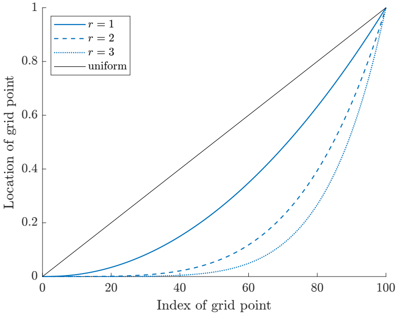
1. Choose number of intervals N and smoothness parameter $r \in \{1, 2, 3\}$
2. Start with initial grid $a = x_0 < \cdots < x_N = b$
3. Fit cubic spline to current values
4. Compute D_n from spline derivative of order r
5. Update grid using closed-form formula
6. Iterate if necessary, or embed update in value/policy iteration

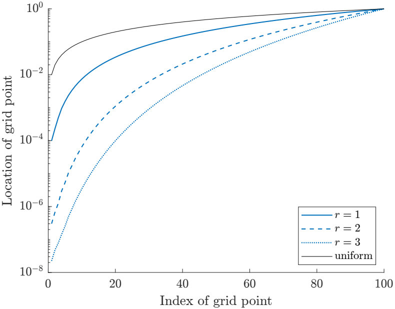
Numerical illustration: $f(x) = \sqrt{x}$

- Start from uniform grid on $[0, 1]$ with $N = 100$
- Update grid repeatedly using spline rule ▸ End condition
- Evaluate error on 10,000-point uniform grid



Adaptive grids concentrate near high curvature





- Larger r places more grid points where curvature is highest
- For $f(x) = \sqrt{x}$, this means dense knots near zero

Application: stochastic growth model

Benchmark model: Brock and Mirman (1972)

$$\begin{aligned} \text{maximize} \quad & \mathrm{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{subject to} \quad & c_t + k_{t+1} = A(z_t)k_t^{\alpha} + (1-\delta)k_t \end{aligned}$$

- Productivity z_t is finite-state Markov chain
- State variable: resource $w_t = A(z_t)k_t^{\alpha} + (1-\delta)k_t$
- Solve Bellman equation by value function iteration with spline interpolation

Implementation details

- Two productivity states with $P(z, z) = 0.8$; $\alpha = 0.36$, $\delta = 0.08$, $\beta = 0.96$, and $u'(c) = c^{-\gamma}$ with $\gamma \in \{0.5, 1.5\}$
- Initial grid: uniform on $[M^{-1}w^*, Mw^*]$ with $M = 10$ and steady state resource $w^* = 1$
- Cubic spline interpolation evaluates continuation value
- Adaptive grid: update using average value function $Z^{-1} \sum_z v(w, z)$
- Stop updating grid after 200 iterations; stop iteration at tolerance 10^{-5}

Accuracy metric: Euler equation errors

- Euler equation:

$$u'(c_t) = \beta \, \mathrm{E}_t \left[u'(c_{t+1})(A(z_{t+1})\alpha k_{t+1}^{\alpha-1} + 1 - \delta) \right]$$

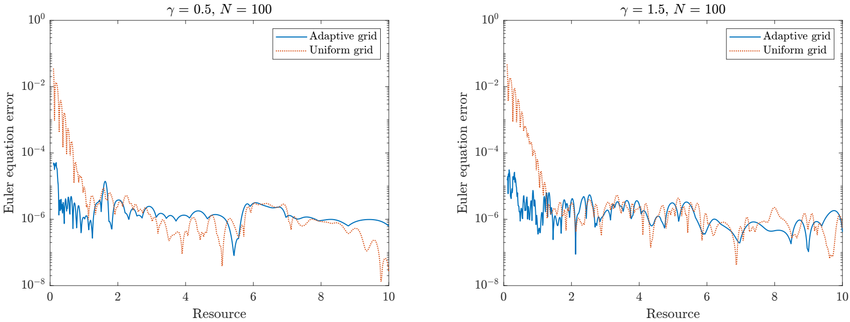
- Euler equation error, with $R_k(\hat{w}, \hat{z}) = A(\hat{z})\alpha k(\hat{w}, \hat{z})^{\alpha-1} + 1 - \delta$:

$$e(w,z) = \left| \frac{(u')^{-1}(\beta \, \mathrm{E}_z \left[u'(c(\hat{w}, \hat{z}))R_k(\hat{w}, \hat{z}) \right])}{c(w,z)} - 1 \right|,$$

evaluate on fine grid

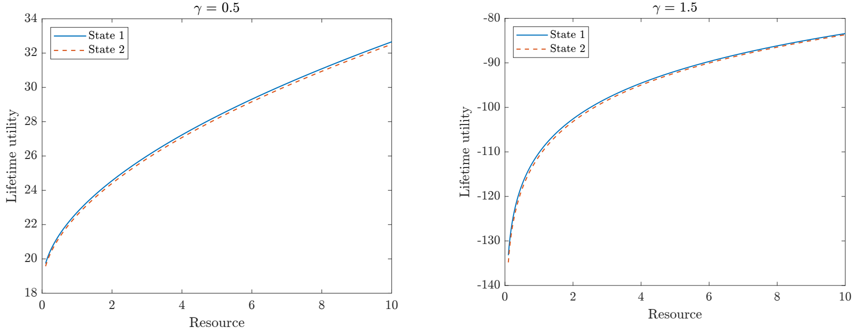
- Policy-function errors are of order $\max e(w,z)$ (Santos, 2000)

Stochastic growth: $N = 100$ Euler errors



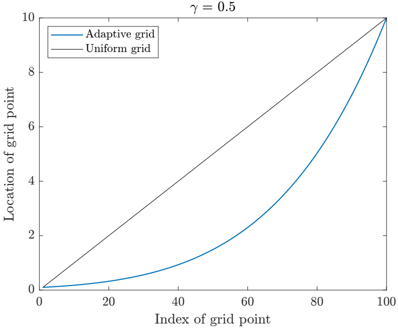
- Uniform grid errors around 10^{-2} for low resources
- Adaptive grid errors uniformly around 10^{-6}

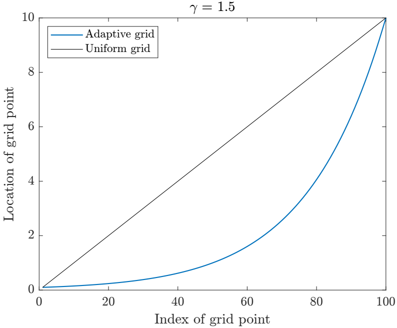
Why does adaptive grid help?



- Value functions increasing, concave, and steep at low resources
- Adaptive rule detects this from spline derivatives

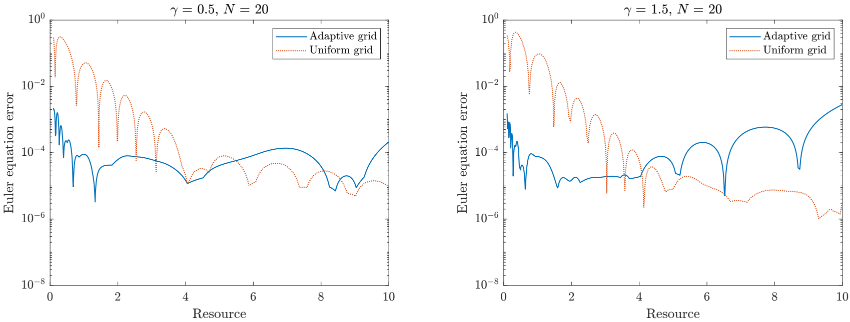
Resulting grid





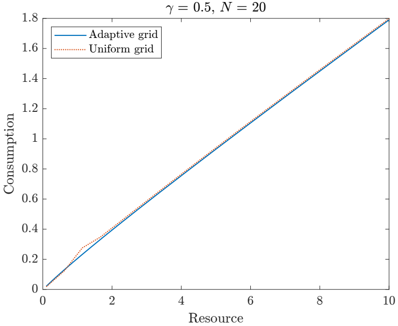
- Most grid points moved to low-resource region
- This is exactly where value function is hardest to interpolate

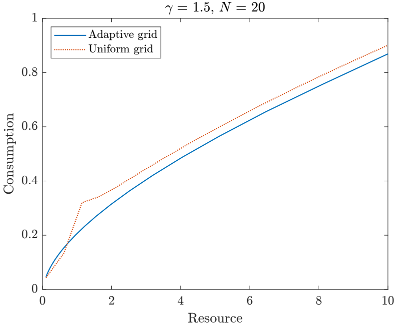
Small grids: errors become visible



- With only $N = 20$ grid points, uniform grid Euler errors around 10^0
- Adaptive grid errors around 10^{-3}

Small grids: policy functions differ



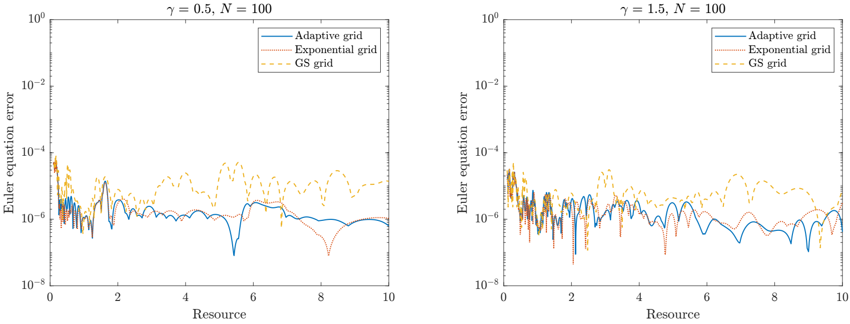


- With coarse grids, interpolation errors translate into economically meaningful policy errors

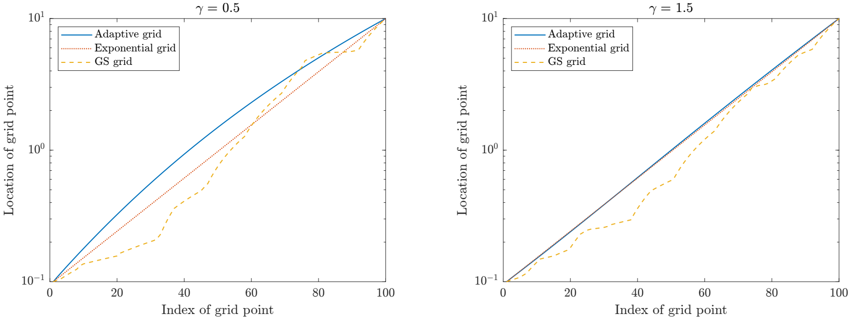
Comparison to alternative grids

Alternatives:

- Exponential grid on $[\log \underline{w}, \log \bar{w}]$ with specified median point (Gouin-Bonenfant and Toda, 2023)
- Error-based adaptive grid (Grüne and Semmler, 2004)



Alternative grid locations



- In this benchmark, exponential grid very good due to prior knowledge ($v' > 0, v'' < 0$)
- Adaptive grids achieve similar accuracy without prior knowledge

Application: optimal savings problem

$$
\begin{aligned}
\text{maximize} \quad & \mathrm{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \\
\text{subject to} \quad & a_{t+1} = R_{t+1}(a_t - c_t) + Y_{t+1}, \\
& 0 \leq c_t \leq a_t
\end{aligned}
$$

- Building block of Bewley-Huggett-Aiyagari models
- Stochastic returns make state space unbounded
- Grid must cover both low assets and far upper tail

Numerical specification

- Same example as Ma and Toda (2022, §33)
- Two aggregate states calibrated from NBER recession data
- Return:

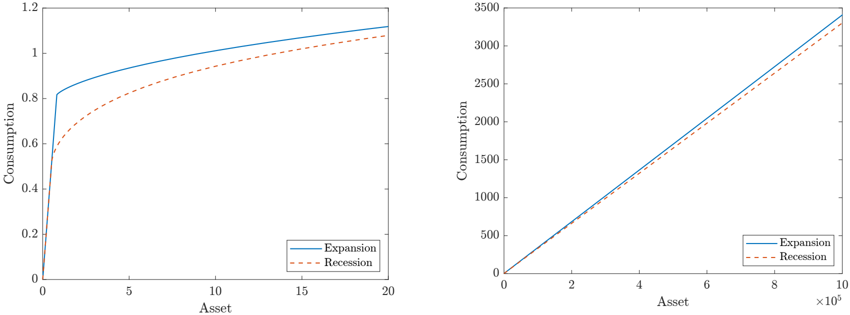
$$R(z,\zeta) = R_f(\theta \exp(\mu(z) + \sigma(z)\zeta) + 1 - \theta),$$

  with portfolio share $\theta = 0.6$
- Income: $Y = 1$ in expansions and $Y = 0.5$ in recessions

Computation

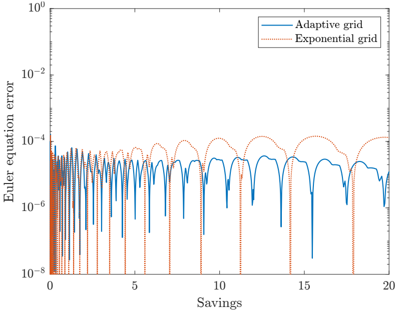
- Use endogenous grid point method (Carroll, 2006)
- Baseline grid: exponential grid for savings on $[0, 10^6]$
- Adaptive grid uses same spline derivative logic as before
- Convergence tolerance: relative error 10^{-5} for the consumption function
- With $N = 100$, both exponential and adaptive grids take about 1 second

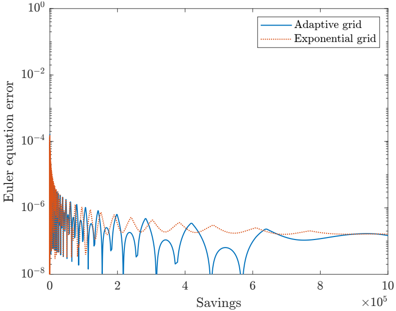
Optimal savings: consumption functions



- Consumption function concave at low assets
- Asymptotically linear at high assets (Ma and Toda, 2021, 2022)

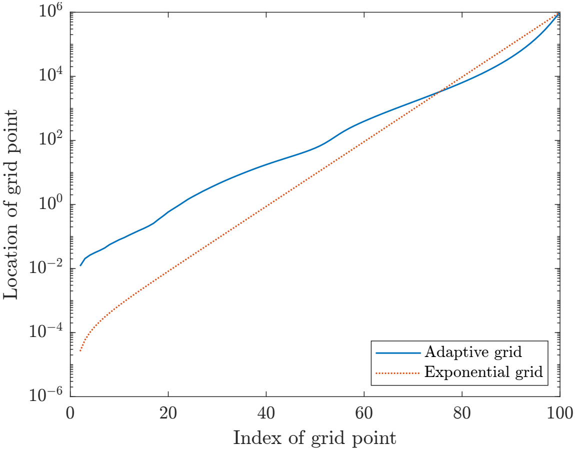
Optimal savings: Euler equation errors





- Evaluate errors on 1,000-point exponential grid
- Adaptive grid slightly outperforms tuned exponential grid on both scales

Optimal savings: grid locations



- Exponential grid places many points below 10^{-2}
- Adaptive grid allocates fewer points near zero because curvature milder than in stochastic growth model

Optimal savings takeaway

- Carefully chosen exponential grid hard to beat in this known application
- Adaptive spline grid performs just as well without choosing turning parameters
- Method especially attractive when researcher does not know ex ante where policy functions are curved

Practical recipe for researchers

1. Start with any reasonable grid
2. During iteration, fit cubic spline to current value or policy function and update grid by closed-form rule
3. Stop updating after burn-in period, then finish fixed-grid iteration

Benefits

- Only a few additional lines in code once spline interpolation is already used
- Almost no additional computation cost due to closed-form rule

Conclusion

- Adaptive spline grids automatically allocate grid points
- Method motivated by classical spline error estimates
- Grid update is closed-form and has negligible computational cost
- True computational cost includes human labor; automation reduces this

References

- Brock, W. A. and L. J. Mirman (1972). "Optimal economic growth and uncertainty: the discounted case". *Journal of Economic Theory* 4.3, 479–513. DOI: 10.1016/0022-0531(72)90135-4.
- Carroll, C. D. (2006). "The method of endogenous gridpoints for solving dynamic stochastic optimization problems". *Economics Letters* 91.3, 312–320. DOI: 10.1016/j.econlet.2005.09.013.
- Gouin-Bonenfant, É. and A. A. Toda (2023). "Pareto extrapolation: An analytical framework for studying tail inequality". *Quantitative Economics* 14.1, 201–233. DOI: 10.3982/QE1817.
- Grüne, L. and W. Semmler (2004). "Using dynamic programming with adaptive grid scheme for optimal control problems in economics". *Journal of Economic Dynamics and Control* 28.12, 2427–2456. DOI: 10.1016/j.jedc.2003.11.002.
- Hall, C. A. and W. W. Meyer (1976). "Optimal error bounds for cubic spline interpolation". *Journal of Approximation Theory* 16.2, 105–122. DOI: 10.1016/0021-9045(76)90040-X.

References

- Judd, K. L. (1992). "Projection methods for solving aggregate growth models". *Journal of Economic Theory* 58.2, 410–452. DOI: 10.1016/0022-0531(92)90061-L.
- Judd, K. L. and S.-M. Guu (1997). "Asymptotic methods for aggregate growth models". *Journal of Economic Dynamics and Control* 21.6, 1025–1042. DOI: 10.1016/S0165-1889(97)00015-8.
- Ma, Q. and A. A. Toda (2021). "A theory of the saving rate of the rich". *Journal of Economic Theory* 192, 105193. DOI: 10.1016/j.jet.2021.105193.
- Ma, Q. and A. A. Toda (2022). "Asymptotic linearity of consumption functions and computational efficiency". *Journal of Mathematical Economics* 98, 102562. DOI: 10.1016/j.jmateco.2021.102562.
- Meinardus, G., G. Nürnberger, M. Sommer, and H. Strauss (1989). "Algorithms for piecewise polynomials and splines with free knots". *Mathematics of Computation* 53.187, 235–247. DOI: 10.1090/S0025-5718-1989-0969492-7.

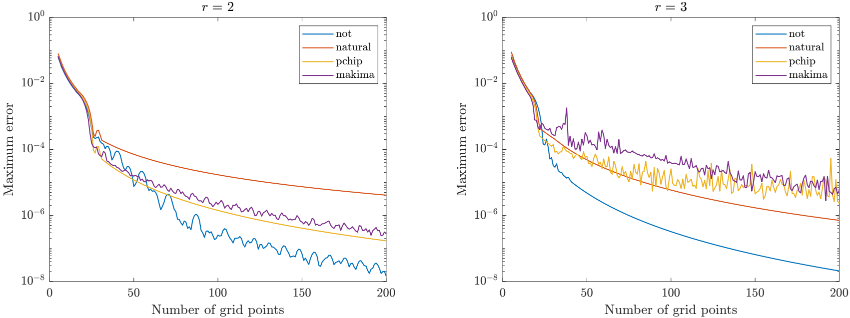
References

Powell, M. J. D. (1981). *Approximation Theory and Methods*. Cambridge University Press.

Rust, J. (2008). "Dynamic programming". In: *The New Palgrave Dictionary of Economics*. UK: Palgrave Macmillan. DOI: 10.1057/978-1-349-95121-5_1932-1.

Santos, M. S. (2000). "Accuracy of numerical solutions using the Euler equation residuals". *Econometrica* 68.6, 1377–1402. DOI: 10.1111/1468-0262.00165.

End conditions and default choice



- Compare not-a-knot, natural, PCHIP, and modified Akima interpolation
- Not-a-knot spline with $r = 2$ or $r = 3$ performs best in this example

return