Using a Hyperbolic Cross to Solve Non-linear Macroeconomic Models

Richard Dennis

PII: S0165-1889(24)00052-6
DOI: https://doi.org/10.1016/j.jedc.2024.104860
Reference: DYNCON 104860

To appear in: *Journal of Economic Dynamics and Control*

Received date: 28 November 2022
Revised date: 2 April 2024
Accepted date: 3 April 2024

Please cite this article as: R. Dennis, Using a Hyperbolic Cross to Solve Non-linear Macroeconomic Models, *Journal of Economic Dynamics and Control*, 104860, doi: https://doi.org/10.1016/j.jedc.2024.104860.

Using a Hyperbolic Cross to Solve Non-linear Macroeconomic Models

Richard Dennis*
University of Glasgow and CAMA

April, 2024

Abstract

The paper presents a sparse grid approximation method based on the hyperbolic cross and applies it to solve non-linear macroeconomic models. We show how the standard hyperbolic cross can be extended to give greater control over the approximating grid and we discuss how to implement an anisotropic hyperbolic cross. Applying the approximation method to four macroeconomic models, we establish that it delivers a level of accuracy on par or better than Smolyak's method and that it can produce accurate approximations using fewer points than Smolyak's method.

Key words: Hyperbolic cross, Smolyak, non-linear models, projection methods.
JEL codes: C63, E52, E70

*I would like to thank two reviewers and participants at the 2023 Computational Economics and Finance conference held in Nice for comments. Address for Correspondence: Adam Smith Business School, University of Glasgow, Main Building, University Avenue, Glasgow G12 8QQ; email: richard.dennis@glasgow.ac.uk.

1 Introduction

The macroeconomic models used for forecasting and policy analysis are growing increasingly larger and more sophisticated over time. Their size and, in particular, their number of state variables mean that perturbation methods—most commonly first-order perturbation methods—are usually used to solve these models. Where non-linear solutions are needed or are of interest the standard toolbox relies upon piecewise linear approximation, splines, and orthogonal polynomials, such as Chebyshev polynomials. But for models with even moderately large spacial dimension, such tools are essentially infeasible due to the curse of dimensionality. For a model with six state variables, placing just nine points along each dimension leads to an approximating grid with over half a million points, presenting a challenge even with modern computing hardware.

Problems associated with such tensor-product grids have lead to increasing awareness and use of sparse grid methods, of which Smolyak's method (Smolyak, 1963) is a leading candidate. Smolyak's method was introduced into economics by Krueger and Kubler (2004), and has subsequently been used for various economic applications by Gavilan-Gonzalez and Rojas (2009), Winschel and Krätzig (2010), Malin, Krueger, and Kubler (2011), Gordon (2011), Brumm and Scheidegger (2017), and Fernández-Villaverde, Gordon, Guerrón-Quintana, and Rubio-Ramírez (2012). More recently, Gust, Herbst, López-Salido, and Smith (2017) used Smolyak's method to solve and estimate a large scale model incorporating the zero lower bound on nominal interest rates and Hirose and Sunakawa (2019) used Smolyak's method to study the natural rate of interest. An excellent presentation of Smolyak's method—clear to the point where the approximation method can be coded based purely on the description provided—and its efficient implementation can be found in Judd, Maliar, Maliar, and Valero (2014); see also Barthelmann, Novak, and Ritter (2000).

In this paper we present an alternative sparse-grid approximation method that is based on the hyperbolic cross (see Dũng, Temlyakov, and Ullrich (2018) for a survey). Sparse grid approximation using the hyperbolic cross has its foundations in multivariate Fourier analysis (see Döhler, Kunis, and Potts (2010) and the references therein) where it is characterized in terms of a nonequispaced fast Fourier transform. We provide an accessible presentation of the hyperbolic cross method and cast the method in terms of Chebyshev polynomials suitable for approximating non-periodic functions, drawing on Shen and Wang (2010). Moreover, we show how the approximation method can be extended to allow for anisotropic grids and

we provide a Julia package that implements the methods described.[1]

Aside from the advantage of having a larger set of tools at our disposal to solve non-linear models, approximation based on the hyperbolic cross offers several advantages over Smolyak's method. One advantage is that unlike Smolyak's method the hyperbolic cross does not require a nested grid structure, which broadens the types of approximating points that can be used. For example, although there are other options, such as the Kronrod-Patterson points, applications of Smolyak's method invariably use a grid formed using Chebyshev extrema (also known as the Chebyshev-Gauss-Lobatto points). With hyperbolic cross approximation one can also use Chebyshev nodes or Legendre nodes, etc. Second, hyperbolic cross approximation can be applied to state spaces with unbounded domain, allowing Hermite nodes or Leguerre nodes to be used in such cases (Shen and Wang, 2010). Third, hyperbolic cross approximation offers the possibility of a reduction in the size of the approximating grid: a three-layer Smolyak approximation over six dimensions requires 389 points whereas a hyperbolic cross approximation can be formed using just 109 points. Fourth, in the case of bounded domain, both Smolyak's method and the hyperbolic cross place approximating points within a hypercube, but, relative to Smolyak's method, the hyperbolic cross places a greater concentration of points in the central region of the hypercube. For a stochastic model, this is tantamount to placing a greater concentration of points in the model's ergodic region, potentially improving solution accuracy over the ergodic region.

We describe several variants of the hyperbolic cross, beginning with a standard symmetric variant, and show by example how the approximating grid can be constructed and the approximating polynomial formed. Then we introduce a generalization on the standard hyperbolic cross, one that encompasses the standard symmetric hyperbolic cross and the full tensor-product grid as special cases. This generalization takes the standard cross and expands the number of points, filling out the hypercube, until the full tensor-product grid is achieved. With this generalization, considerable control over the approximation grid can be achieved. We illustrate the resulting grid and contrast it to the Smolyak grid. Next we show how this generalized hyperbolic cross can be extended further to allow varying numbers of approximating points along each spacial dimension, producing, essentially, a hyperbolic cross anisotropic grid.

To assess the accuracy of hyperbolic cross approximation we apply the approximation method to four macroeconomic models. The simplest of these four models is the standard

[1]The HyperbolicCrossApprox.jl package is written in Julia; it is a registered package that can be installed using Julia's package manager.

stochastic growth model that has two state variables, the largest is a six-country international business cycle model that has twelve state variables. Between these two extremes we consider two new Keynesian models, one of which has policy conducted using a Taylor-type rule and has four state variables, the other of which is based on a labor-search framework and has monetary policy conducted optimally under discretion. This latter model, which has six state variables, is of particular interest because computing equilibrium requires simultaneously approximating functions and the derivatives of functions. We solve each of these four models using the hyperbolic cross method and the Smolyak method, each with various layers of approximation. The accuracy of each solution is assessed in terms of the average and largest Euler-equation error, and the root-mean-square-Euler-error, evaluated over the model's ergodic region.

The results from these applications are encouraging. When the hyperbolic cross approximation grid and the Smolyak approximation grid are constructed to have the same number of points along each dimension and the same number of total points, the results suggest that the hyperbolic cross produces a more accurate solution, a result that is entirely down to where in the hypercube the approximating points are placed. More generally, the hyperbolic cross often produced a solution with acceptable accuracy using fewer approximating points. These results suggest that hyperbolic cross approximation is indeed a useful method for solving models with higher-dimensional state spaces, expanding our numerical toolbox and providing a viable sparse-grid alternative to Smolyak's method.

The remainder of the paper is structured as follows. In section two, we provide some background to motivate the hyperbolic cross, highlighting the importance of variable separation. In section three, we present what we will call the standard hyperbolic cross approximation scheme; this presentation is done in terms of Chebyshev polynomials rather than Fourier methods. We also develop a generalization on the standard hyperbolic cross that nests the standard cross and the tensor-product grid as special cases and we show how hyperbolic crosses supporting anisotropic grids can be constructed. Throughout this section, the presentation is purposefully based on a constructed example, making the method easier to follow and implement. In section four we turn to applications. We solve to different levels of accuracy four macroeconomic models using Smolyak's method and the hyperbolic cross method. These models vary from having as few as two state variables to having as many as twelve state variables. For each model we present the solution accuracy in terms of Euler-equation errors and the solution time. Section fives concludes.

2 Prelude

Suppose we have a continuous function of the form:

$$y = h(x, z), \tag{1}$$

where $h : [-1, 1]^2 \to \mathbb{R}$ and x and z represent two spacial variables. In many applications of interest to economists, h could be a decision rule or a value function. Often we do not have an analytic expression for the function, h, but we can use numerical methods to approximate it. Common ways to approximate a function like h are to use splines, some form of polynomial or orthogonal polynomial, radial basis functions, etc. To approximate h using Chebyshev approximation, we might use N sampling (or approximating) points along each spacial dimension, leading to a grid with $M = N^2$ points. If h, instead, depended on d variables, then this strategy of using N sampling points for each variable would lead to a grid with $M = N^d$ points.

Now suppose that the function h has additional structure. In particular, suppose that h is a rank-one function, so the variables x and z separate and equation (1) can be written as:

$$y = f(x)\, g(z), \tag{2}$$

where $f : [-1, 1] \to \mathbb{R}$ and $g : [-1, 1] \to \mathbb{R}$. Because the two spacial variables separate in equation (2), Chebyshev approximation with N sampling points along each dimension requires a grid with just $M = 2N - 1$ points. Extending this reasoning to functions with d spacial variables that separate, the grid needed for Chebyshev approximation would involve $M = dN - 1$ points. Looking at equation (2), notice that if f and g are continuous functions of order r, i.e. $f \in C^r([-1, 1])$ and $g \in C^r([-1, 1])$, then the derivatives of h, $\frac{\partial^{|\mathbf{i}|_1} h}{\partial x^{i_1} \partial z^{i_2}}$, $\mathbf{i} = (i_1, i_2)$, are finite and continuous on $[-1, 1]^2$ for all $i_1 \leq r$ and $i_2 \leq r$.[2] Essentially, if f and g are even low-order continuous (second order, say), then the function h will possess considerable continuity (some fourth derivatives will be finite and continuous), which can be exploited for approximation.

Clearly, far fewer points are required to approximate functions where the spacial variables separate. Crucially, where the cardinality of the grid rises exponentially with d when the variables do not separate, it rises linearly with d when they do—the former case is subject to the curse of dimensionality whereas the latter case is not.

[2]Importantly, these derivative restrictions than stronger than those needed for $h \in C_2^r([-1, 1]^2)$, which requires $\frac{\partial^{|\mathbf{i}|_1} h}{\partial x^{i_1} \partial z^{i_2}}$ to be finite and continuous on $[-1, 1]^2$ for $|\mathbf{i}|_1 \leq r$.

For graphical purposes, suppose $d = 2$ and that the function takes the form of equation (2), then the approximating grid is shown in Figure 1. As Figure 1 illustrates, for this case the approximating grid takes the form of what is known as a hyperbolic cross. This particular hyperbolic cross is quite extreme insomuch as no points are placed off the leading axes, such points are not needed because the function being approximated fully separates by x and z, allowing h to be approximated through what are essentially independent approximations of f and g.

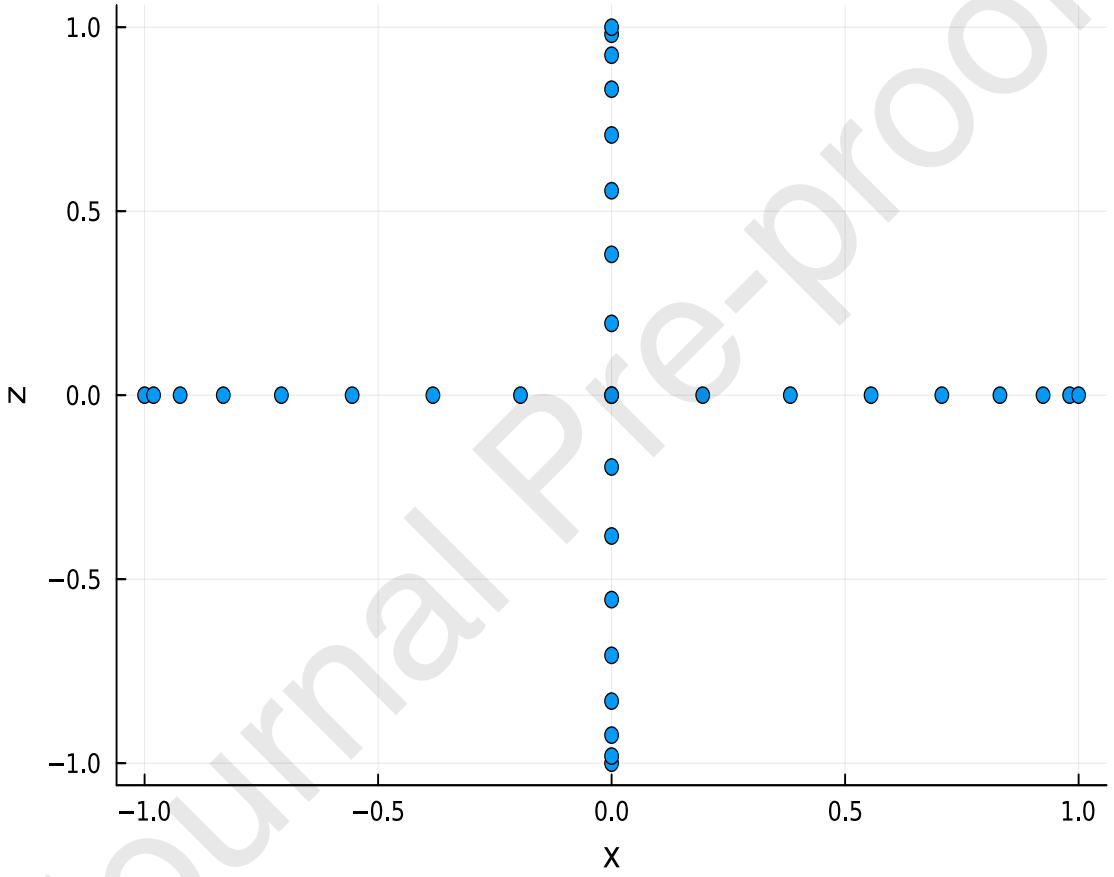

Figure 1: A simple hyperbolic cross to approximate $y = f(x)g(z)$.

The grid structure shown in Figure 1 is central to sparse grid approximation methods. It appears within the hyperbolic cross grids we will develop and illustrate below, as well as within the sparse grids generated by Smolyak's method. However, when the function to be approximated does not fully separate by variables, the core grid shown in Figure 1 gets augmented to include some points in each of the four quadrants. The key differences

between the hyperbolic cross approximation method and Smolyak's approximation method relate to the number of additional points that are used and where they are placed in each quadrant.

2.1 Chebyshev polynomials

The approximations we consider below are built on basis functions consisting of Chebyshev polynomials. Chebyshev polynomials satisfy the following recurrence relation:

$$\Gamma_0(x) = 1, \tag{3}$$

$$\Gamma_1(x) = x, \tag{4}$$

$$\Gamma_j(x) = 2x\Gamma_{j-1}(x) - \Gamma_{j-2}(x), \tag{5}$$

where $x \in [-1, 1]$. An approximation based on N sampling points would invariably have those points chosen either as the zeros of the polynomial $\Gamma_N(x)$:

$$x_i = -\cos\left(\frac{2\iota - 1}{2N}\pi\right), \quad \iota = 1, ..., N, \tag{6}$$

or as the extrema of the polynomial, $\Gamma_N(x)$:

$$x_i = -\cos\left(\frac{\iota - 1}{N - 1}\pi\right), \quad \iota = 1, ..., N. \tag{7}$$

However the sampling points are chosen, Chebyshev approximation based on an n-order polynomial takes the form:

$$\widetilde{f}(x) = \sum_{\kappa=0}^{n} \omega_\kappa \Gamma_\kappa(x), \tag{8}$$

where the weights, ω_κ, $\kappa \in \{0, 1, ..., n\}$ can be computed by making use of the relevant discrete orthogonality property.

The approximation is typically generalized to the multivariate case—say d variables—by forming the tensor product:

$$\widetilde{h}(\mathbf{x}) = \sum_{\kappa_1=0}^{n_1} ... \sum_{\kappa_d=0}^{n_d} \omega_{\kappa_1 ... \kappa_d} \Gamma_{\kappa_1}(x_1) \times .... \times \Gamma_{\kappa_d}(x_d), \tag{9}$$

where $\mathbf{x} = (x_1, ...x_d) \in [-1, 1]^d$, see, for example, Judd (1998).

3 The hyperbolic cross

This section culminates by presenting a sparse-grid approximation scheme based on the hyperbolic cross that can be used to solve models with a moderate to large number of state variables on a standard personal computer. We will begin by presenting the standard hyperbolic cross and then show how we extend this standard cross to allow greater control over the approximating grid. Finally, we compare the hyperbolic cross approximating grid to Smolyak's approximating grid and we will show how to implement an anisotropic hyperbolic cross. Throughout, we illustrate the hyperbolic cross method using constructed examples, beginning with the standard cross.

For convenience, we will introduce the notion of a multi-index:

$$\mathbf{i} = (i_1, ..., i_d) \in \mathbb{Z}^d,$$

where $\mathbf{i}$, the multi-index, is essentially a tuple of integers, one for each dimension of the state-space. Now, as mentioned in the Introduction, hyperbolic cross approximation has its origins in multivariate Fourier approximation (see Dũng, Temlyakov, and Ullrich (2018)) with the functions of interest taken to be of the class:

$$\mathbb{W}_d^r = \left\{ h : \Omega \to \mathbb{R},\ \left\| \frac{\partial^{|\mathbf{i}|_1} h}{\partial x_1^{i_1} ... \partial x_d^{i_d}} \right\|_\infty < \infty,\ i_j < r \right\}, \qquad j = 1, ...d, \tag{10}$$

where h is 2π-periodic and $\Omega = [-\pi, \pi]^d$. The functions that belong to this class are those whose mixed partial derivatives are bounded for low orders, which Novak and Ritter (1998) show implies some partial separability by variables. Periodic functions that fully separate by variables fall within the class of functions described by equation (10).[3]

The functions that we are interested in approximating are generally not periodic. Therefore, we will take $\Omega = [-1, 1]^d$ and construct our approximating grid to be a hyperbolic cross space associated with orthogonal polynomials (see also Shen and Wang (2010)). Specifically, we will use the transform $x = cos(\theta)$ to switch from Fourier points to Chebyshev points and we will employ basis functions whose constituents are Chebyshev polynomials.

The standard hyperbolic cross is governed by the set of multi-indices, $\mathbb{S}^{d,k}$, that satisfy:[4]

$$\mathbb{S}^{d,k} = \left\{ \mathbf{i} = (i_1, ..., i_d) \in \mathbb{Z}^d : \prod_{j=1}^{d} (|i_j| + 1) \leq k + 1 \right\}. \tag{11}$$

[3]Although the rank-one function discussed in section 2 is not periodic, there is a clear parallel between the derivative conditions discussed there and those stated in equation (10).

[4]In practice, symmetry can be exploited and only multi-indices whose elements are all non-negative integers need to be calculated and stored.

Thus, if $d = 2$ and $k = 2$, then the set of multi-indices $\mathbb{S}^{d,k}$ is given by: $\mathbb{S}^{2,2} = \{(0,0), (-1,0), (1,0), (-2,0), (2,0), (0,-1), (0,1), (0,-2), (0,2)\}$.

Once $\mathbb{S}^{d,k}$ has been constructed, the next step is to use the multi-indices in $\mathbb{S}^{d,k}$ to generate the set of approximation points. As we will see below, the elements in a multi-index, $\mathbf{i}$, represent indexation-based "off-sets" relative to a central point. We will use $\mathbb{H}^{d,k}$ to denote the set of approximating points and we will call this set of points the approximation grid. The approximation grid contains points from the d-dimensional hypercube.[5] In the construction below we use Chebyshev polynomials as basis functions and employ Chebyshev extrema to construct the approximation grid.[6]

Let the number of approximation points along each dimension be given by:

$$N = 2k + 1, \tag{12}$$

$k = 0, 1, 2, ...$, which is always an odd number, and then compute these N points, x_ι, $\iota = 1, ...N$, using the equation for Chebyshev extrema, equation (7). With $k = 2$, this leads to $N = 5$, and the approximating points being given by $\left\{-1, -\sqrt{0.5}, 0, \sqrt{0.5}, 1\right\}$. As mentioned previously, the elements in the multi-indices contained in $\mathbb{S}^{d,k}$ are "off-set" values relative to a central point (which is 0). For the case where $k = 2$, this relationship can be illustrated through the table:

Table 1: Relating indices to approximating points

Index value	-2	-1	0	1	2
Approximating point	-1	$-\sqrt{0.5}$	0	$\sqrt{0.5}$	1

Therefore, with $d = 2$ and $k = 2$, the multi-index $\mathbb{S}^{2,2}$ generates the approximating grid, $\mathbb{H}^{2,2} = \{(0,0), \left(-\sqrt{0.5}, 0\right), \left(\sqrt{0.5}, 0\right), (-1,0), (1,0), \left(0, -\sqrt{0.5}\right), \left(0, \sqrt{0.5}\right), (0,-1), (0,1)\}$.

Figure 2 displays a series of hyperbolic crosses constructed for $d = 2$, but for various values of k. The example presented above in which $k = 2$ is shown in panel C.

It is clear from Figure 2 that as k increases the total number of approximating points and the number of approximating points along each axis rise. It terms of its structure, the hyperbolic cross tends to place approximating points in the region close to the center of the hypercube, and along the leading axes, while the diagonal quadrants contain few points.

[5] As is well-known, through a simple normalization points in a d-dimensional hyper-rectangle can be transformed into points in a d-dimenstional hypercube, giving the approach considerable generality.

[6] Unlike Smolyak's method, however, the hyperbolic cross does not require a nested grid structure, allowing alternatives, such as Chebyshev nodes or Legendre nodes to be used.

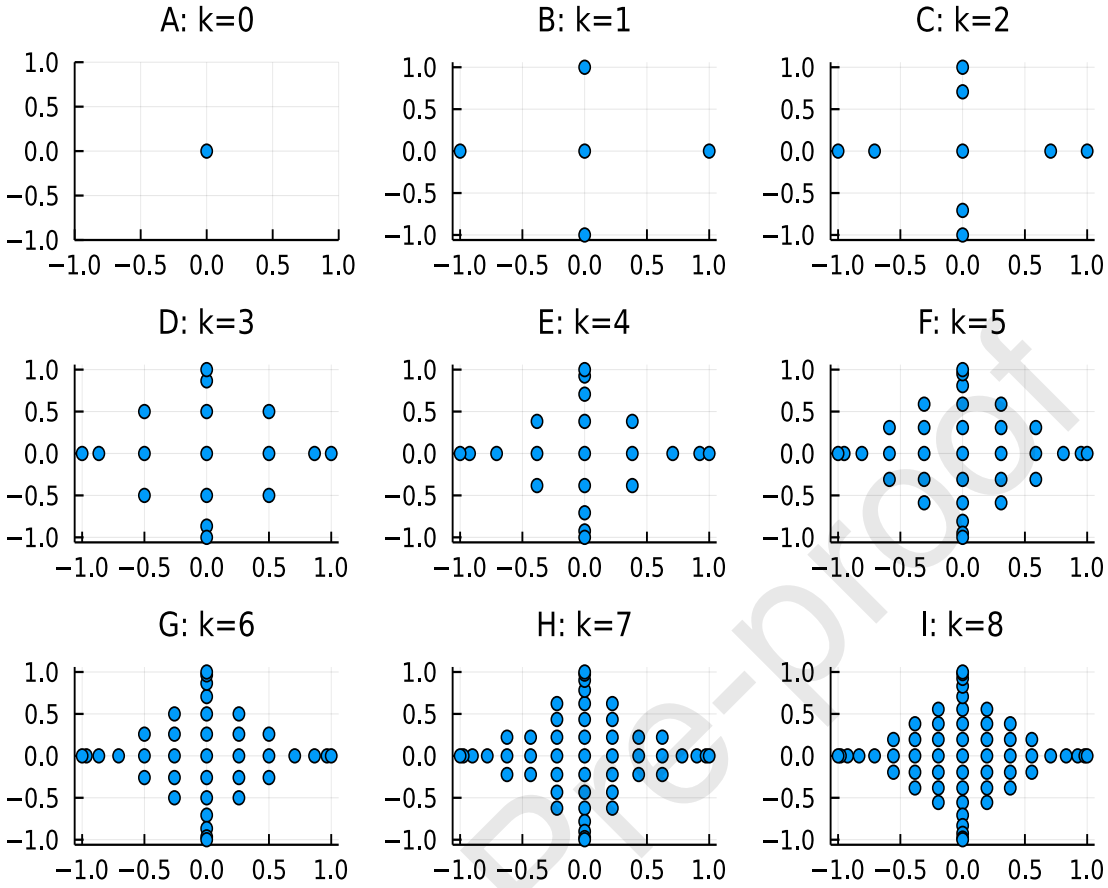

Figure 2: Hyperbolic cross as k increases

3.1 A more general approach

Here we describe a more general approach to constructing a hyperbolic cross, an approach that encompasses the standard cross as a special case. This more general approach dispenses with equation (12) and instead allows N to be set explicitly, while requiring N to be odd. Let $\underline{k}$ be given by:

$$\underline{k} = \frac{N-1}{2}. \tag{13}$$

So if $N = 5$, then $\underline{k} = 2$. Our generalization allows $k \geq \underline{k}$ and constructs the set of multi-indices $\mathbb{S}^{d,k,N}$ according to:

$$\mathbb{S}^{d,k,N} = \left\{ \mathbf{i} = (i_1, ..., i_d) \in \mathbb{Z}^d : \prod_{j=1}^{d} (|i_j| + 1) \leq k + 1, \;\; |i_j| \leq \frac{N-1}{2} \right\}. \tag{14}$$

By way of example, let $d = 2$, $N = 5$, and $k = 3$. With $N = 5$, equation (13)

implies $\underline{k} = 2$ and $\mathbb{S}^{2,3,5} = \{(0,0),\ (-1,0),\ (1,0),\ (-2,0),\ (2,0),\ (0,-1),\ (0,1),\ (0,-2),\ (0,2),\ (-1,-1),\ (1,-1),\ (-1,1),\ (1,1)\}$. For this case, relative to the standard hyperbolic cross, we have the additional multi-indices: $(-1,-1),\ (1,-1),\ (1,-1),\ (1,1)$. Once the set of multi-indices, $\mathbb{S}^{d,k,N}$, is constructed the procedure for computing the approximating grid, $\mathbb{H}^{d,k,N}$, is essentially the same as that described above and gives rise to $\mathbb{H}^{2,2,5} = \{(0,0),\ \left(-\sqrt{0.5},0\right),\ \left(\sqrt{0.5},0\right),\ (-1,0),\ (1,0),\ \left(0,-\sqrt{0.5}\right),\ \left(0,\sqrt{0.5}\right),\ (0,-1),\ (0,1),\ \left(-\sqrt{0.5},-\sqrt{0.5}\right),\ \left(\sqrt{0.5},-\sqrt{0.5}\right),\ \left(-\sqrt{0.5},\sqrt{0.5}\right),\ \left(\sqrt{0.5},\sqrt{0.5}\right)\}$.

The key difference between the two approaches is that the general approach largely uncouples N and k, giving greater flexibility over the shape of the resulting hyperbolic cross. By choosing values for N and k that satisfy equation (12) the standard hyperbolic cross can be replicated. Alternatively, by increasing k, for given N, the more general approach produces a hyperbolic cross that converges to a tensor-product grid (With $d = 2$ and $N = 5$, setting $k = 8$ produces the full tensor-product grid).

In effect, this generalization provides a bridge that allows functions with varying orders of continuity to be approximated within a single framework. For functions that fall within the class given by equation (10), the standard hyperbolic cross approximating scheme can be used. Whereas, for a function $h : [-1,1]^d \to \mathbb{R}$, where $h \in C_d^0([-1,1]^d)$, one may need to use the full tensor-product grid, and one can do that by making k sufficiently large. This generalization spans both situations and provides the flexibility to approximate both classes of functions. Of course, other ways of generalizing the standard hyperbolic cross are possible. For example, in the case where $d >\geq 3$ equation (14) could additionally place restrictions on the product of any two indices, which might be useful in some situations.[7] The central point here is that the hyperbolic cross framework provides considerable flexibility over the number of approximating points used and their positioning within the hypercube.

Figure 3 illustrates a series of hyperbolic crosses generated for various combinations of N and k. Notice that panels A, D, and G, in Figure 3 correspond to panels C, F, and I in Figure 2, illustrating the point that this second approach encompasses the first approach.

As we move to the right along each row of Figure 3, k is being raised while N is being held constant. Looking at the resulting hyperbolic crosses it is clear that this process increases the size of the approximating grid by increasingly padding out the central region of the hypercube, pushing approximating points out into the diagonal regions of the cube. Constructing the grid this way gives considerable control over the resulting approximating

[7] I would like to thank a referee for this observation.

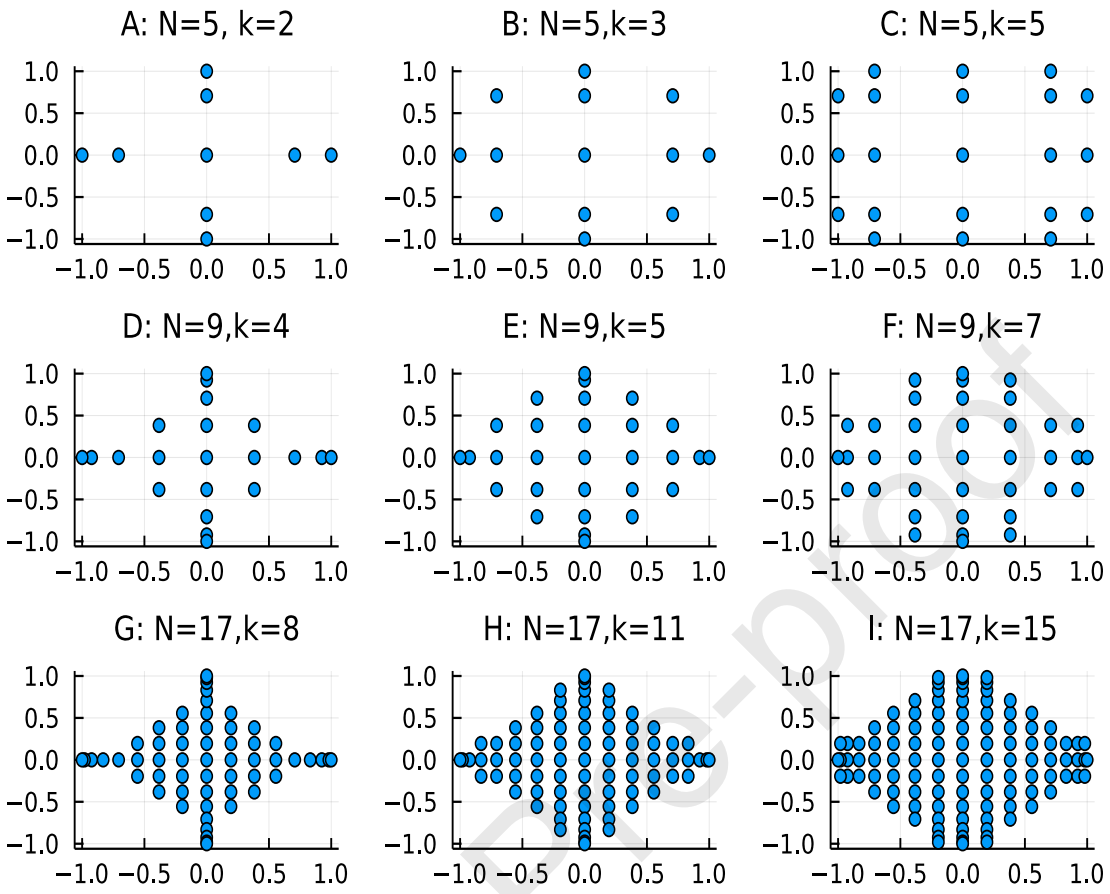

Figure 3: Hyperbolic cross as N and k increase.

grid, and, as mentioned earlier, if one continues to increase k one eventually arrives at the full tensor-product grid.

3.1.1 The approximating polynomial

Let M denote the cardinality of the approximating grid (either $\mathbb{H}^{d,k}$ or $\mathbb{H}^{d,k,N}$, depending on which approach is used), then the approximating polynomial takes the form:

$$\widetilde{h}\left(\mathbf{x}\right) = \sum_{m=1}^{M} \omega_m \Psi_m\left(\mathbf{x}\right), \tag{15}$$

where $\mathbf{x} \in [-1,1]^d$, and $\Psi_m\left(\mathbf{x}\right) \to \mathbb{R}$ and ω_m, $m = 1, ..., M$, are d-dimensional basis functions and scalar weighting coefficients, respectively. What remains now is to show how the ω_m and $\Psi_m\left(\mathbf{x}\right)$, $m = 1, ..., M$, are constructed. The approach described here follows closely the

Lagrange interpolation scheme proposed by Judd, *et al,* (2014) in the context of Smolyak's method. The M elements of the approximating grid are determined as described above. At each point in the approximating grid we evaluate the function, giving $y_m = h(\mathbf{x}_m)$, $m = 1, ..., M$. Then the polynomial weights are computed by solving the linear system:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \Psi_1(\mathbf{x}_1) & \cdots & \Psi_M(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\mathbf{x}_M) & \cdots & \Psi_M(\mathbf{x}_M) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_M \end{bmatrix}. \tag{16}$$

As mentioned earlier, we use basis functions constructed using Chebyshev polynomials in the approximation. With N points in each dimension, the set of Chebyshev polynomials for the variable x_1 is $\{\Gamma_0(x_1), ..., \Gamma_{N-1}(x_1)\}$. Similarly, the set of Chebyshev polynomials for x_2 is $\{\Gamma_0(x_2), ..., \Gamma_{N-1}(x_2)\}$. As with the approximating grid itself, the basis functions that are included are determined by the set of multi-indices, $\mathbb{S}^{d,k}$ (or $\mathbb{S}^{d,k,N}$). To illustrate this process we consider the case where $d = 2$ and $k = 3$ for which $\mathbb{S}^{2,3} = \{(0,0), (-1,0), (1,0), (-2,0), (2,0), (-3,0), (3,0), (0,-1), (0,1), (0,-2), (0,2), (0,-3), (0,3), (-1,-1), (1,-1), (-1,1), (1,1)\}$ and $M = |\mathbb{H}^{2,3}| = 17$. A multi-index value of 0 corresponds to the 0'th-order polynomial, index values of -1 and 1 correspond to the 1'st- and 2'nd-order polynomials, etc, as the following table makes clear:[8]

Table 2: Relating indices to polynomial-orders

Index value	-3	-2	-1	0	1	2	3
Polynomial order	5	3	1	0	2	4	6

Therefore, for this two-dimensional case the approximating polynomial is built up of the polynomial terms:

$$\begin{bmatrix} \Psi_1(\mathbf{x}) \\ \Psi_2(\mathbf{x}) \\ \Psi_3(\mathbf{x}) \\ \Psi_4(\mathbf{x}) \\ \Psi_5(\mathbf{x}) \\ \Psi_6(\mathbf{x}) \\ \Psi_7(\mathbf{x}) \\ \Psi_8(\mathbf{x}) \\ \Psi_9(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \Gamma_0(x_1)\Gamma_0(x_2) \\ \Gamma_1(x_1)\Gamma_0(x_2) \\ \Gamma_2(x_1)\Gamma_0(x_2) \\ \Gamma_3(x_1)\Gamma_0(x_2) \\ \Gamma_4(x_1)\Gamma_0(x_2) \\ \Gamma_5(x_1)\Gamma_0(x_2) \\ \Gamma_6(x_1)\Gamma_0(x_2) \\ \Gamma_0(x_1)\Gamma_1(x_2) \\ \Gamma_0(x_1)\Gamma_2(x_2) \end{bmatrix}, \qquad \begin{bmatrix} \Psi_{10}(\mathbf{x}) \\ \Psi_{11}(\mathbf{x}) \\ \Psi_{12}(\mathbf{x}) \\ \Psi_{13}(\mathbf{x}) \\ \Psi_{14}(\mathbf{x}) \\ \Psi_{15}(\mathbf{x}) \\ \Psi_{16}(\mathbf{x}) \\ \Psi_{17}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \Gamma_0(x_1)\Gamma_3(x_2) \\ \Gamma_0(x_1)\Gamma_4(x_2) \\ \Gamma_0(x_1)\Gamma_5(x_2) \\ \Gamma_0(x_1)\Gamma_6(x_2) \\ \Gamma_1(x_1)\Gamma_1(x_2) \\ \Gamma_2(x_1)\Gamma_1(x_2) \\ \Gamma_1(x_1)\Gamma_2(x_2) \\ \Gamma_2(x_1)\Gamma_2(x_2) \end{bmatrix}.$$

Implementing this process when the number of spacial dimensions is above two is straightforward.

[8]Notice that the symmetry of the cross means that we could equally assign index values of -1 and 1 to the 2'nd and 1'st order polynomials, etc.

To compute the weights in the approximating polynomial, the direct approach is to solve the inverse problem:

$$\begin{bmatrix} \omega_1 \\ \vdots \\ \omega_M \end{bmatrix} = \begin{bmatrix} \Psi_1(\mathbf{x}_1) & \cdots & \Psi_M(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \Psi_1(\mathbf{x}_M) & \cdots & \Psi_M(\mathbf{x}_M) \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}, \tag{17}$$

implied by equation (16). In order for equation (17) to be well-defined the interpolating matrix must have full-rank, and for the polynomial weights to be computed accurately the matrix inversion must be numerically stable. In these respects, the use of Chebyshev polynomials as basis functions is advantageous, helping to prevent ill-conditioning of the interpolating matrix.[9,10] The interpolating matrix that needs to be inverted can be large, making this inversion exercise numerically expensive. Fortunately, as Judd, Maliar, Maliar, and Valero (2014) note in the context of Smolyak interpolation, the interpolating matrix does not depend in any way on the function being approximated, which means that the interpolating matrix need only be constructed and inverted once, mitigating the cost.

3.2 Hyperbolic cross grid compared to the Smolyak grid

Here we compare the number of approximating points using by the hyperbolic cross method, and the location of these points, to Smolyak's method. Table 3 reports the size of the approximating grid for approximations using Chebyshev polynomials, Smolyak's method and the hyperbolic cross method as the number of dimensions, d, changes. For Smolyak's method and the hyperbolic cross we also consider different "layers" of approximation. With the hyperbolic cross the cardinality of the approximating grid can depend on both k and N as well as d. What is reported in Table 3 is the number of approximating points as k changes when the grid is constructed using the standard approach (for which N is uniquely pinned down by k). In this sense the numbers reported for the hyperbolic cross represent a lower bound on the number of points that would be used by the more general approach.

[9]This comment parallels the discussion in Judd, Maliar, Maliar, and Valero (2014) where analogous issues are encountered when computing the weights in their Smolyak polynomial.

[10]The Chebyshev polynomials are not orthogonal on the hyperbolic cross grid. However, in my experience with hyperbolic cross approximation for this paper I did not encounter any situations where the interpolating matrix was poorly-conditioned or rank-deficient.

Table 3: Approximating Grid Size, M

	Chebyshev	Smolyak				Hyperbolic Cross			
	N	μ				k			
d	7	1	2	3	4	1	2	4	8
2	49	5	13	29	65	5	9	21	57
4	2,401	9	41	137	401	9	17	57	241
6	117,649	13	85	389	1,457	13	25	109	617
8	5,764,801	17	145	849	3,937	17	33	177	1,249
10	282,475,249	21	221	1,581	8,801	21	41	261	2,201

If a tensor-product grid is used with $N = 7$, then over 100,000 points are required when there are 6 state variables and over 282 million points are required when there are 10 state variables. Depending on the problem and the number of processors available, even 100,000 points could easily be troublesome. By way of contrast, with $d = 10$ and the number of layers, μ, equal to 4, Smolyak's method requires less than 9,000 grid points. With $\mu = 4$ Smolyak's method places 17 nodes along each leading axis; the hyperbolic cross does the same when $k = 8$. Strikingly, then, when $d = 10$ and $k = 8$ the hyperbolic cross requires just 2,201 grid points. Using the number of nodes along each axis as the normalizing factor, $k = 1, 2, 4, 8$ correspond to $\mu = 1, 2, 3, 4$, respectively. We can see, then, that the number of approximating points required by the standard hyperbolic cross is never larger than the number required by Smolyak's method.

More specifically, as Krueger and Kubler (2004) report, the cardinality of the Smolyak grid is given by:

$$\begin{aligned}
\mu = 1 : M &= 1 + 2d, \\
\mu = 2 : M &= 1 + 4d + \frac{4}{2}d(d-1), \\
\mu = 3 : M &= 1 + 8d + \frac{12}{2}d(d-1) + \frac{8}{6}d(d-1)(d-2).
\end{aligned}$$

By way of contrast, for the standard hyperbolic cross the cardinality of the approximating grid is given by:

$$\begin{aligned}
k = 1 : M &= 1 + 2d, \\
k = 2 : M &= 1 + 4d, \\
k = 4 : M &= 1 + 8d + 2d(d-1).
\end{aligned}$$

Thus, keeping N equal across the two approximation methods, the hyperbolic cross uses the same number of points as Smolyak's method when $N = 3$ ($\mu = 1$ or $k = 1$). In this case

the cardinality of both grids increases linearly in d. But for $N = 5$ ($\mu = 2$ or $k = 2$) or $N = 9$ ($\mu = 3$ or $k = 4$), the cardinality of both grids increases polynomially, but where the order of the polynomial is lower for the hyperbolic cross grid than for the Smolyak grid. As a consequence, the curse of dimensionality applies with less force to the hyperbolic cross, as illustrated in Table 3.

Next we consider where Smolyak's method and the hyperbolic cross place the approximating points within the hypercube. Consider the case where $d = 2$ and suppose that the number of layers in the Smolyak approximation is four, $\mu = 4$. Then the Smolyak grid contains 65 points which are located in $[-1, 1]^d$ as shown Figure 4, panel B. By way of comparison, Figure 4, panel A, displays the tensor-product grid (with $N = 17$), which contains 289 points. The hyperbolic cross grid with $N = 17$ and $k = 10$ is shown in Figure 4, panel C; these values for N and k were chosen for this figure because they produce the same number of points along each leading axis (17) and the same number of total grid points (65).

The Smolyak grid and the hyperbolic cross grid are clearly very different. Where Smolyak's method tends to place approximating points at the edges of the hypercube, the hyperbolic cross tends to place them closer to the center of the hypercube.[11] In Figure 4, panel D, we display the probability density function for capital and technology associated with the stochastic growth model (taken from example one in section 4). The ergodic region for this model is naturally centered in the middle of the hypercube, suggesting that greater numerical accuracy over the ergodic region may be obtained by using the hyperbolic cross grid rather than the Smolyak grid [12]

3.3 Isotropic and anisotropic grids

Smolyak's method allows the number of nodes to be varied according to the dimension, allowing more nodes to be placed along dimensions that exhibit greater curvature and less nodes along dimensions with less curvature. Such grids are known as anisotropic grids. Hyperbolic cross approximation, when the grid is constructed according to the general approach, also allows for anisotropic grids. Let N_j be the number (odd) of nodes to be placed

[11] Related to this point is an observation by Judd (2006) that there can be efficiency gains to approximating within a hyper-sphere relative to a hyper-cube.

[12] As mentioned, this figure has been developed in the context of the stochastic growth model that is analyzed as example one in section 4. For this model, the technology state variable is taken to be the exponential of a normally distributed shock with zero-unconditional-mean. As a consequence the technology state follows a log-normal distribution and has mean greater than one, as illustrated in the figure.

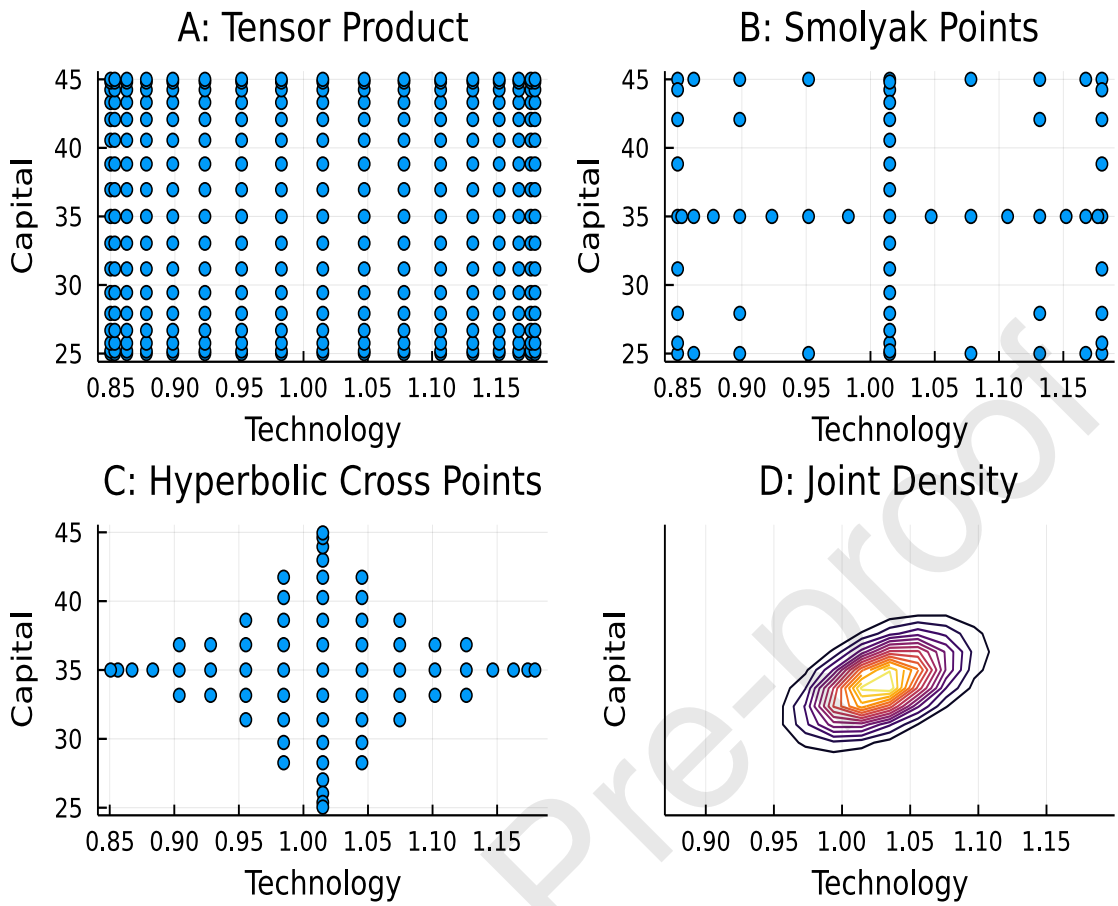

Figure 4: Possible approximating grids for the stochastic growth model.

along dimension j, $j = 1, ..., d$, and define $\mathbf{N} = (N_1, ..., N_d)$, then the set of multi-indices for an anisotropic hyperbolic cross is given by:

$$\mathbb{S}^{d,k,\mathbf{N}} = \left\{ \mathbf{i} = (i_1, ..., i_d) \in \mathbb{Z}^d : \prod_{j=1}^{d} (|i_j| + 1) \leq k + 1, \;\; |i_j| \leq \frac{N_j - 1}{2} \right\}. \tag{18}$$

Once the multi-index has been constructed, the anisotropic grid and the basis functions for the hyperbolic cross can be generated as described earlier.[13]

[13]The methods described in this section are implemented in the Julia package HyperbolicCrossApprox.jl (currently version 0.2.6), which is a registered package that can be installed using Julia's package manager. There are three core functions in the package: 'hyperbolic_cross_grid', which computes the grid and the multi-index, $\mathbf{i}$, 'hyperbolic_cross_weights', which uses Lagrange interpolation to compute the polynomial's weights, and 'hyperbolic_cross_evaluate', which evaluates the approximated function at a point in the state space.

4 Examples

In this section we solve a range of macroeconomic models and consider the accuracy of approximations based on the hyperbolic cross method relative to Smolyak's method. Because Smolyak's method requires nested grids we are constrained for this comparison to use approximating points that produce this nesting: in this case we use Chebyshev extrema. The models themselves include some that are commonly used when comparing solution methods—the stochastic growth model (example one) and a multi-country international business cycle model (example four)—and two from the new Keynesian literature on monetary model. Of these new Keynesian models, one is relatively standard with monetary policy conducted using a Taylor-type rule (example two) while the second combines sticky prices with labor-search and has monetary policy conducted optimally under discretion (example three). This third example is unique in so much as solving the model requires approximating decision rules and the derivatives of decision rules.

We solve these four models for various approximating grids and evaluate the accuracy of the resulting solution using Euler-equation errors. The Euler-equation errors are evaluated at points in the state-space obtained by simulating data from each solved model and then sampling realizations for the state variables from the simulated data. In this way, the accuracy comparison is focused on the ergodic region.[14]

4.1 Example one — The stochastic growth model

The stochastic growth model needs little introduction. A representative consumer/producer has capital stock, k_t, and makes decisions regarding consumption, c_t, and future capital in order to maximize expected discounted lifetime utility, which depends on the sequence of goods consumed. We assume that period-utility is of the iso-elastic form. With goods produced according to a Cobb-Douglas technology and with aggregate technology, a_t, obeying a standard stationary AR(1) process, the key equations characterizing equilibrium are:

$$a_{t+1} = \rho a_t + \varepsilon_{t+1}, \tag{19}$$

$$k_{t+1} = (1-\delta)\, k_t + e^{a_t} k_t^{\alpha} - c_t, \tag{20}$$

$$c_t^{-\sigma} = \beta E_t \left[c_{t+1}^{-\sigma} \left(1 - \delta + \alpha e^{a_{t+1}} k_{t+1}^{\alpha-1} \right) \right]. \tag{21}$$

[14] All simulations were conducted using Julia 1.9.4 on an AMD Ryzen 5900 CPI with 32 GB RAM and 24 threads. To accelerate the computations, we pre-computed the integrals required for quadrature according to the procedure described in Judd, Maliar, Maliar, and Tsener (2017).

Equation (20) is the law-of-motion for capital, which allows the capital stock to be augmented by unconsumed production and to depreciate at rate $\delta \in (0, 1]$. Equation (21) is the standard consumption-Euler equation in which $\beta \in (0, 1)$ is the discount factor, $\sigma \in (0, \infty)$ is the inverse of the elasticity of intertemporal substitution, and E_t is the mathematical expectations operator. When solving the model we set $\beta = 0.99$, $\sigma = 2.0$, $\alpha = 0.3$, $\delta = 0.015$ and $\rho = 0.95$. The standard deviation of the technology innovation, ε_t, is set to 0.01.

We evaluate accuracy by computing the Euler-equation errors based on equation (21):

$$\mathbb{E}\left(k_t, a_t\right) = \frac{\left(\beta E_t\left[\widetilde{g}\left(k_{t+1}, a_{t+1}\right)\right]\right)^{-\frac{1}{\sigma}}}{\widetilde{c}\left(k_t, a_t\right)} - 1, \tag{22}$$

where:

$$g\left(k_t, a_t\right) = c_t^{-\sigma}\left(1 - \delta + \alpha e^{a_t} k_t^{\alpha - 1}\right), \tag{23}$$

and report for each approximation the average ($\log_{10}$) absolute error, the largest ($\log_{10}$) absolute error, and the root-mean-square-error (RMSE).

4.1.1 Numerics

There are two state variables ($d = 2$), technology and capital, which have steady state values of 1.0 and and 34.609, respectively. The domain of approximation is given by $a_t \in [\log(0.85), \log(1.18)]$ and $k_t \in [25.0, 45.0]$, a region found by stochastic simulation ($2,000,000$ observations) to contain the ergodic distribution. From these $2,000,000$ simulated observations we sample $200,000$ points and evaluate the Euler-equation error at each point. Table 4 provides details of the solution details for Smolyak approximation and hyperbolic cross approximation, and reports the total number of points in the approximating grid, the largest absolute Euler error, the average absolute Euler-error, the root-mean-square-Euler-error (RMSE), and the solution time, in seconds.

Table 4: Accuracy results for the stochastic growth model

$d = 2$	M	$\max(\log_{10}\lvert\mathbb{E}\rvert)$	$\text{mean}(\log_{10}\lvert\mathbb{E}\rvert)$	RMSE($\mathbb{E}$)	Time (sec.)
Smolyak					
$\mu = 1$	5	-1.914	-3.088	$1.2e^{-3}$	0.03
$\mu = 2$	13	-3.217	-4.307	$7.9e^{-5}$	0.10
$\mu = 3$	29	-4.743	-5.617	$3.5e^{-6}$	0.29
$\mu = 4$	65	-6.774	-7.438	$5.1e^{-8}$	0.86
$\mu = 5$	145	-8.154	-9.110	$1.0e^{-9}$	3.19
Hyperbolic cross					
$N = 3, k = 1$	5	-1.914	-3.088	$1.2e^{-3}$	0.02
$N = 5, k = 2$	9	-1.926	-3.290	$8.6e^{-4}$	0.06
$N = 5, k = 3$	13	-3.581	-4.500	$4.6e^{-5}$	0.10
$N = 9, k = 4$	21	-3.389	-5.425	$7.0e^{-6}$	0.20
$N = 9, k = 5$	29	-5.260	-6.904	$1.7e^{-7}$	0.32
$N = 17, k = 8$	57	-5.852	-8.883	$1.2e^{-8}$	0.99
$N = 17, k = 9$	65	-6.246	-8.946	$6.6e^{-9}$	1.20
$N = 13, k = 29$	145	-8.217	-8.893	$2.1e^{-9}$	4.09

For Smolyak's method we consider an isotropic grid and allow the number of layers, μ, to vary. An approximation with three layers ($\mu = 3$) requires an approximation grid with just 29 points, solves in just under 0.3 seconds, and produces an average Euler-equation error of about -5.6. If four layers are used ($\mu = 4$), then 65 approximating points are required, the solution time rises to about 0.9 seconds, and the average Euler-equation error falls to around -7.4. Consider now approximation based on a hyperbolic cross, with $N = 9$ and $k = 5$ the resulting hyperbolic cross uses the same number of approximating points in each dimension as Smolyak (9) and has the same total number of approximation points, (29), as Smolyak's method with $\mu = 3$. Relative to this Smolyak solution, the approximation using the hyperbolic cross is slightly slower (just over 0.3 seconds), but is more accurate with an average absolute Euler-error of about -6.9. Increasing N to 17 and k to 9 gives a hyperbolic cross that is closely comparable to Smolyak's method with $\mu = 4$ (17 approximating points along each dimension and a total of 65 points in the approximation grid). The resulting approximation produces greater average accuracy (around -8.9), but is slightly slower to compute. An intermediate hyperbolic cross that uses $N = 17$ and $k = 8$, generates an approximation grid with just 57 points, solves in a time that is comparable to a four-layer Smolyak approximation, but is considerably more accurate in terms of average error.

As a final experiment, we solve the stochastic growth model using Smolyak's method with $\mu = 5$, which utilizes 145 approximating points. With five layers, Smolyak's method produces a very accurate solution, with the average absolute Euler error of about -9.1.

The hyperbolic cross with $N = 13$ and $k = 29$ also produces a grid with 145 approximating points. Although the hyperbolic cross approximation is also very accurate, its average absolute Euler error is -8.9, which is about the same as the approximation using just 65 points and slightly less accurate than the five-layer Smolyak approximation.[15]

4.2 Example two — A new Keynesian model

The second example is a new Keynesian model in which the goods market is monopolistically competitive, prices are sticky as per Rotemberg (1982), and monetary policy is conducted according to a Taylor-type rule. This particular model allows for capital accumulation and accommodates a discount factor shock, an aggregate technology shock, and a monetary policy shock, leading to a total of four state variables ($d = 4$) one of which is endogenous (capital). This model is a special case of the one analyzed in Dennis (2019), where the model's full derivation can be found.

The model's key equations can be summarized as:

$$\lambda_t = E_t\left[\beta_{t+1}\lambda_{t+1}\left(\frac{1+R_t}{1+\pi_{t+1}}\right)\right], \tag{24}$$

$$\lambda_t = E_t\left[\beta_{t+1}\lambda_{t+1}\left(1-\delta+\frac{\alpha}{1-\alpha}\frac{1-\theta}{\theta}\frac{h_{t+1}}{k_{t+1}}\frac{c_{t+1}}{1-h_{t+1}}\right)\right], \tag{25}$$

$$\begin{aligned}\pi_t\left(1+\pi_t\right) &= \frac{1-\epsilon}{\psi}+\frac{\epsilon}{\psi}\frac{\alpha}{1-\alpha}\frac{1-\theta}{\theta}\left(\frac{h_t}{k_t}\right)^{\alpha}\frac{c_t}{1-h_t} \\ &+ E_t\left[\frac{\beta_{t+1}\lambda_{t+1}e^{a_{t+1}}k_{t+1}^{\alpha}h_{t+1}^{1-\alpha}\pi_{t+1}\left(1+\pi_{t+1}\right)}{\lambda_t e^{a_t}k_t^{\alpha}h_t^{1-\alpha}}\right],\end{aligned} \tag{26}$$

$$\lambda_t = c_t^{(\theta-1-\sigma\theta)}\left(1-h_t\right)^{(1-\theta)(1-\sigma)}, \tag{27}$$

$$k_{t+1} = \left(1-\delta\right)k_t + i_t, \tag{28}$$

$$c_t + i_t = \left(1-\frac{\psi}{2}\pi_t^2\right)e^{a_t}k_t^{\alpha}h_t^{1-\alpha}, \tag{29}$$

$$1+R_t = \frac{1}{\beta}\left(1+\pi_t\right)^{\phi_\pi}\left(\frac{e^{a_t}k_t^{\alpha}h_t^{1-\alpha}}{\overline{y}}\right)^{\phi_y}e^{m_t}, \tag{30}$$

[15] I was unable to perform this same experiment on the other three models as they each failed to solve using Smolyak's method with $\mu = 5$ (or $\mu = 4$ in the case of the six-country model). I did not encounter any numerical problems using the hyperbolic cross.

where the processes for the three shocks are:

$$a_{t+1} = \rho_a a_t + \sigma_a \varepsilon_{t+1}, \tag{31}$$

$$\log(\beta_{t+1}) = (1-\rho_\beta)\log(\beta) + \rho_\beta \log(\beta_t) + \sigma_\beta \nu_{t+1}, \tag{32}$$

$$m_{t+1} = \rho_m m_t + \sigma_m \eta_{t+1}. \tag{33}$$

To solve the model we set $\beta = 0.99$, $\delta = 0.015$, $\alpha = 0.36$, $\phi = 80.0$, $\epsilon = 11.0$, $\theta = 0.39$, $\sigma = 1.0$, $\phi_\pi = 2.5$, $\phi_y = 0.25$, $\overline{\pi} = 0.005$, $\overline{y} = 1.339$, $\rho_a = 0.90$, $\sigma_a = 0.008$, $\rho_\beta = 0.80$, $\sigma_\beta = 0.004$, $\rho_m = 0.80$, and $\sigma_m = 0.01$. Finally, we assess accuracy using the Euler-equation error associated with equation (25), with the error expressed as a proportion of equilibrium consumption, i.e.:

$$\mathbb{E}(k_t, a_t, \beta_t, m_t) = \left(\frac{E_t\left[\widetilde{g}(k_{t+1}, a_{t+1}, \beta_{t+1}, m_{t+1})\right]}{\left(1 - \widetilde{h}(k_t, a_t, \beta_t, m_t)\right)^{(1-\sigma)(1-\theta)}} \right)^{\frac{1}{\theta - 1 - \sigma\theta}} \frac{1}{\widetilde{c}(k_t, a_t, \beta_t, m_t)} - 1, \tag{34}$$

where:

$$g(k_t, a_t, \beta_t, m_t) = \beta_t \lambda_t \left(1 - \delta + \frac{\alpha}{1-\alpha}\frac{1-\theta}{\theta}\frac{h_t}{k_t}\frac{c_t}{1-h_t}\right). \tag{35}$$

4.2.1 Numerics

This new Keynesian model has four state variables. The state-space domain used for the approximation is given by $k_t \in [14.0, 21.0]$, $a_t \in [log(0.9), log(1.1)]$, $\beta_t \in [log(0.95), log(1.05)]$ and $m_t \in [log(0.9), log(1.1)]$, chosen based on a stochastic simulation of $2,000,000$ observations. The Euler-equation errors are computed using equation (34) evaluated at $200,000$ randomly chosen points in the state space. The results are reported in Table 5.

Table 5: Accuracy results for the new Keynesian model

$d = 4$	M	$\max(\log_{10}\lvert\mathbb{E}\rvert)$	$\text{mean}(\log_{10}\lvert\mathbb{E}\rvert)$	RMSE($\mathbb{E}$)	Time (sec.)
Smolyak					
$\mu = 1$	9	-1.742	-2.798	$2.2e^{-3}$	0.25
$\mu = 2$	41	-3.547	-4.540	$4.0e^{-5}$	3.82
$\mu = 3$	137	-5.168	-6.269	$7.1e^{-7}$	27.41
$\mu = 4$	401	-6.325	-6.449	$3.6e^{-7}$	160.61
Hyperbolic cross					
$N = 3, k = 1$	9	-1.742	-2.798	$2.2e^{-3}$	0.26
$N = 5, k = 2$	17	-1.771	-2.900	$1.8e^{-3}$	0.84
$N = 5, k = 3$	41	-3.818	-4.796	$2.2e^{-5}$	3.72
$N = 9, k = 4$	57	-3.862	-5.447	$6.1e^{-6}$	7.57
$N = 9, k = 5$	105	-4.122	-5.553	$5.0e^{-6}$	20.24
$N = 9, k = 7$	185	-5.860	-6.408	$3.9e^{-7}$	54.03
$N = 17, k = 11$	481	-5.982	-6.407	$3.9e^{-7}$	295.93

For this new Keynesian model, Smolyak's method with three layers ($\mu = 3$) employs a grid of only 137 points to produce an average absolute error of -6.3 and takes only 27 seconds to solve. Increasing the number of layers to 4, leads to a slight improvement in accuracy while increasing the solution time to almost three minutes.

Looking at the results for the hyperbolic cross approximation now, finding values for N and k that replicate the size of the grid used for Smolyak is not always possible. When $N = 3$ and $k = 1$ the hyperbolic cross grid contains 9 points, the same as a Smolyak approximation with one layer. Similarly, when $N = 5$ and $k = 3$ the hyperbolic cross grid employs 41 points, the same as the Smolyak approximation with two layers. In both of these cases the hyperbolic cross approximation has the same or superior accuracy, and equivalent solution times. While the exact size of the Smolyak grids for layers 3 and 4 do not have direct analogues in the hyperbolic cross, the accuracy achieved from a four-layer Smolyak approximation employing 401 grid points is essentially achieved using a hyperbolic cross where $N = 17$ and $k = 7$, which requires just 185 grid points. More generally, where the hyperbolic cross leads to a reduction in accuracy, this loss of accuracy is generally modest and compensated for in terms of fewer grid points and shorter solution times.

4.3 Example three — A labor search model

Although our third example is also a sticky price new Keynesian model, it poses quite different challenges to the second example above. In particular, example three uses a labor-search model and has monetary policy conducted optimally under discretion, and it is

solved in the absence of a production subsidy so that the under-production of goods caused by monopolistic competition remains. As is now well-known, equilibrium is characterized by a discretionary inflation bias (as well as a stabilization bias) and the model's equilibrium requires solving a system that contains "generalized Euler equations". In other words, due to time-inconsistency, to solve for the discretionary equilibrium one must simultaneously compute decision rules and their derivatives with respect to the endogenous state variables—here lagged employment.

We do not present the model in its entirely here, but simply summarize it, leaving a more thorough presentation to Appendix A, where the decision problems and key equations are reported and described. The model's full description and complete derivation can be found in Dennis and Kirsanova (2021) to which interested readers are directed. The key agents in the model are households, firms, and a central bank. Households consist of members that are either employed or unemployed; the former receive an hourly wage the later receive an unemployment benefit financed by lump-sum taxation. There is complete insurance within the household. On the production side, goods are produced using a technology that depends on aggregate technology, hours-worked per-employee, and the level of employment. Goods are sold in a monopolistically competitive market at a price that is subject to a Rotemberg (1982) adjustment cost. Job-separations occur exogenously at the end of every period and firms face a cost to posting vacancies in order to hire new employees. The central bank conducts monetary policy optimally under discretion. This model contains five autoregressive shocks: technology shocks (a_t), matching shocks (m_t), bargaining shocks (ς_t), mark-up shocks (ϵ_t), and consumption-preference shocks (ζ_t), which together with lagged employment (n_{t-1}) lead to a total of six state variables ($d = 6$).

4.3.1 Numerics

The approximation domain for each of the five shocks was set to be plus/minus four unconditional standard deviations while the domain for lagged employment was $n_{t-1} \in [0.85, 1.0]$. Numerical accuracy was assessed based on the model's job-creation equation sampled at $200,000$ points chosen randomly from a stochastic simulation spanning $2,000,000$ periods. The accuracy results are presented in Table 6.

Table 6: Accuracy results for the labor-search model

$d = 6$	M	$\max(\log_{10}\lvert\mathbb{E}\rvert)$	$\text{mean}(\log_{10}\lvert\mathbb{E}\rvert)$	RMSE($\mathbb{E}$)	Time (sec.)
Smolyak					
$\mu = 1$	13	-0.287	-1.822	$2.2e^{-2}$	0.34
$\mu = 2$	85	-1.296	-2.649	$3.2e^{-3}$	3.94
$\mu = 3$	389	-2.235	-3.713	$3.0e^{-4}$	46.57
$\mu = 4$	1,457	-1.717	-4.972	$4.6e^{-5}$	483.66
Hyperbolic cross					
$N = 3, k = 1$	13	-0.298	-1.831	$2.1e^{-2}$	0.29
$N = 5, k = 2$	25	-0.439	-2.066	$1.4e^{-2}$	0.67
$N = 5, k = 3$	85	-1.696	-3.161	$1.0e^{-3}$	4.66
$N = 5, k = 7$	365	-2.460	-3.500	$3.6e^{-4}$	52.35
$N = 9, k = 5$	229	-1.757	-3.530	$5.5e^{-4}$	19.96
$N = 9, k = 7$	509	-2.233	-4.407	$8.3e^{-5}$	76.85
$N = 9, k = 11$	1,289	-2.157	-4.710	$3.1e^{-5}$	473.05
$N = 17, k = 9$	737	-1.898	-4.472	$9.6e^{-5}$	176.10
$N = 17, k = 11$	1,457	-1.966	-5.099	$3.1e^{-5}$	588.09

For this labor-search model, a Smolyak polynomial with three layers requires just 389 approximating points and gives an Euler-equation error of -3.7 in about 47 seconds. Increasing the number of layers to four, lowers the Euler-equation error to about -5.0, but increases the size of the approximating grid to 1,457 points and the solution time to about 8 minutes.

Turning now to the hyperbolic cross, with $N = 3$ and $k = 1$ the hyperbolic cross approximation is comparable to a one-layer Smolyak approximation, and delivers almost identical accuracy. With $N = 5$ and $k = 3$, the hyperbolic cross is comparable to a two-layer Smolyak approximation, but delivers improved accuracy with a slightly faster solution time. There is no hyperbolic cross that is directly comparable to a three-layer Smolyak approximation, but with $N = 9$ and $k = 5$ the hyperbolic cross is only slightly worse in terms of accuracy (-3.5 versus -3.7), but uses only 229 approximating points (Smolyak employs 389 points) and takes only 20 seconds to solve (versus 47 seconds for Smolyak). Finally, setting $N = 17$ and $k = 11$, the hyperbolic cross is comparable to a four-layer Smolyak approximation, but produces better accuracy while taking longer to solve the model.

For this model, which requires solving functions along with their derivatives, the hyperbolic cross approximation tends to give equal or slightly superior numerical accuracy than a comparable Smolyak approximation, but can take longer to converge.

4.4 Example four — A six-country international business cycle model

The final example is a six-country international business cycle model with complete markets. Briefly, preferences for country $s = 1, ...6$, can be summarized in the form:

$$U_s = E_0 \left[\sum_{t=0}^{\infty} \beta^t \left(\frac{c_{s,t}^{1-\sigma} - 1}{1-\sigma} \right) \right], \tag{36}$$

where $c_{s,t}$ is period-t consumption for country s, and each country produces according to the technology:

$$y_{s,t} = A e^{a_{s,t}} k_{s,t}^{\alpha}, \qquad s = 1, ..., 6, \tag{37}$$

where $y_{s,t}$ and $k_{s,t}$ are period-t production and period-t capital for country s, and $a_{s,t}$ is country s's aggregate technology. The parameter A serves to normalize each country's steady state level of capital to 1.0. We will determine competitive equilibrium in this model through the use of a benevolent planner that weighs equally each country's preferences. The decision problem for the planner is to choose $\{c_{s,t}, k_{s,t+1}\}_{t=0}^{\infty}$, $s = 1, ...6$, to maximize:

$$E_0 \left[\sum_{s=1}^{6} \sum_{t=0}^{\infty} \beta^t \left(\frac{c_{s,t}^{1-\sigma} - 1}{1-\sigma} \right) \right], \tag{38}$$

subject to:

$$\sum_{s=1}^{6} k_{s,t+1} = (1-\delta) \sum_{s=1}^{6} k_{s,t} + \sum_{s=1}^{6} \left(A e^{a_{s,t}} k_{s,t}^{\alpha} - c_{s,t} \right). \tag{39}$$

The first order conditions that hold for all t are:

$$c_{s,t}^{-\sigma} = \lambda_t, \qquad s = 1, ...6, \tag{40}$$

$$\lambda_t = \beta E_t \left[\lambda_{t+1} \left(1 - \delta + \alpha A e^{a_{s,t}} k_{s,t}^{\alpha-1} \right) \right], \qquad s = 1, ...6, \tag{41}$$

$$\sum_{s=1}^{6} k_{s,t+1} = (1-\delta) \sum_{s=1}^{6} k_{s,t} + \sum_{s=1}^{6} \left(e^{a_{s,t}} k_{s,t}^{\alpha} - c_{s,t} \right). \tag{42}$$

We assume that the technology shocks evolve over time according to the process:

$$a_{s,t+1} = \rho a_{s,t} + \sigma_{\varepsilon} \varepsilon_{s,t+1.} \tag{43}$$

Equations (40)—(42) describe equilibrium in a model with twelve state variables ($d = 12$), six technology shocks and six capital stocks. We parameterize the model by setting $\beta = 0.99$, $\sigma = 1.0$, $\alpha = 0.36$, $\delta = 0.025$, $\rho = 0.95$, $A = 0.0975$, and $\sigma_{\varepsilon} = 0.01$, and evaluate accuracy using the error in equation (41) expressed as a proportion of consumption.

A multi-country model essentially the same as the one above was also used to study accuracy in Judd, *et al*, (2017) and Cai, *et al*, (2017).

4.4.1 Numerics

To solve the model use as the approximation domain $k_{s,t} \in [0.7, 1.3]$ and $a_{s,t} \in [\log(0.8), \log(1.2)]$, $s = 1, ..., 6$, which we found through stochastic simulation of $2,000,000$ periods to encompasses the ergodic region. From these simulated outcomes, a random sample of $200,000$ were drawn and used to compute Euler-equation errors. The results are given in Table 7, with solution times reported in minutes.

Table 7: Accuracy results for the six-country model

$d = 12$	M	$\max(\log_{10}\lvert\mathbb{E}\rvert)$	$\text{mean}(\log_{10}\lvert\mathbb{E}\rvert)$	$\text{RMSE}(\mathbb{E})$	Time (min.)
Smolyak					
$\mu = 1$	25	-2.485	-3.719	$2.9e^{-4}$	0.10
$\mu = 2$	313	-3.692	-5.352	$8.1e^{-6}$	7.44
$\mu = 3$	2,649	-4.910	-6.883	$2.8e^{-7}$	365.71
Hyperbolic cross					
$N = 3, k = 1$	25	-2.484	-3.718	$2.9e^{-4}$	0.10
$N = 5, k = 3$	313	-3.737	-5.419	$7.4e^{-6}$	6.88
$N = 5, k = 5$	841	-3.776	-5.447	$7.0e^{-6}$	55.31
$N = 7, k = 5$	865	-3.786	-5.470	$6.7e^{-6}$	54.99
$N = 7, k = 7$	3,153	-5.102	-7.099	$1.9e^{-7}$	465.27
$N = 9, k = 5$	889	-3.785	-5.470	$6.7e^{-6}$	54.46
$N = 9, k = 7$	3,177	-5.131	-7.134	$1.8e^{-7}$	430.98

With twelve state variables the state space for this model is sufficiently large that obtaining a solution using Chebyshev polynomials with a tensor-product grid is essentially infeasible. For Smolyak approximation with layers one and two, formulations based on hyperbolic cross approximation with a comparable number of approximating points are available. Comparing a one-layer Smolyak approximation with a hyperbolic cross with $N = 3$ and $k = 1$ (both using a grid of 25 points) we see that the two solutions share the same accuracy and overall solution time. A hyperbolic cross with $N = 5$ and $k = 3$ produces a grid with 313 points, which is comparable to a two-layer Smolyak approximation. Here the hyperbolic cross is slightly more accurate, with the improvement due entirely to the placement of the approximation points within the d-dimensional hypercube, and the solution is slightly faster to compute. Looking at the three-layer Smolyak approximation, for which the average absolute Euler-equation error is about -6.9, there is no directly comparable hyperbolic cross with the same sized approximation grid (2,649 points). However, with some loss of accuracy a hyperbolic cross with $N = 7$ and $k = 5$ gives an average error of -5.5, uses just 865 approximating points, and computes the solution in just 55 seconds.

5 Conclusions

This paper has shown how non-linear macroeconomic models can be solved using a sparse-grid method based on the hyperbolic cross. The hyperbolic cross method has been developed to approximate certain classes of smooth functions that exhibit some partial separability of variables and offers the possibility of greater accuracy than other sparse-grid methods because it concentrates the approximating points in a model's ergodic region. The hyperbolic cross method is related to Smolyak's method, but places the approximating points differently and generally facilitates a solution using fewer approximating points. We illustrated the standard hyperbolic cross, presented a generalization, and then showed how a hyperbolic cross with an anisotropic grid could be constructed. After illustrating the hyperbolic cross method we compared its grid placement to Smolyak's method.

We assessed the performance and numerical accuracy of the method by using hyperbolic cross approximation to solve four macroeconomic models. One of these models is the stochastic growth model, another is a six-country international business cycle model, and the remaining two models are of the new Keynesian variety. These models offer variation in the number of state variables, the stochastic growth model has two while the international business cycle model has twelve, in the composition of the state variables between endogenous and exogenous, and one of the new Keynesian models requires simultaneously solving for functions and the derivatives of functions to arrive at the solution. To provide comparison, for all four of these models we also presented solutions based on Smolyak's method.

The paper offers a variety of contributions and results. One contribution is to introduce the hyperbolic cross approximation method and show how it can be employed to solve non-linear macroeconomic models. A second contribution is to generalize the standard hyperbolic cross, to develop a framework that links the standard cross at one end to a tensor-product grid at the other. A third contribution is to show how a hyperbolic cross with an anisotropic grid can be constructed, giving the hyperbolic cross approximation the same flexibility over the number of points to be used along each spacial dimension as Smolyak's method. Applying the approximation method to the four models, the main findings are as follows. First, the hyperbolic cross method allows models to be solved using fewer approximating points than Smolyak's method, often considerably fewer. Second, in much the same way that Smolyak's method permits models to be solved using fewer points than a tensor-product grid, but at some loss in accuracy, so the hyperbolic cross permits using fewer points than Smolyak's method, but at some loss in accuracy. Third, in cases where

the hyperbolic cross is constructed to have the same number of approximating points as Smolyak's method, the hyperbolic cross produced an accuracy improvement in almost every case, sometimes at the cost of slightly slower solution times. This improvement in accuracy is due to the hyperbolic cross focusing the approximating points on the ergodic region. Finally, at least for the stochastic growth model, which could be solved using a five-layer Smolyak approximation, when the number of approximating points was large the accuracy advantages of the hyperbolic cross over Smolyak's method was less evident.

Smolyak's method is the method most commonly used used to solve macroeconomic models when Chebyshev approximation using a tensor-product grid is infeasible. The hyperbolic cross method illustrated in this paper provides an alternative, one that allows further economy over the number of approximating points, and that offers considerable flexibility over where the approximating points are placed. As the number of state variables increases, the hyperbolic cross offers obvious improvement over a tensor-product grid and, depending on the model, may also offer improvement over Smolyak's method, either in terms of improved accuracy and/or in terms of faster solution times.

Appendix A: The labor-search model—more detail

In this appendix we provide details on the labor-search model that was analyzed as example three, listing the model's equations and the parameter values used. Complete details can be found in Dennis and Kirsanova (2021).

A household consists of employed and unemployed members (complete insurance within the household) and the decision problem for the representative household is to choose $\{c_t, B_{t+1}\}_{t=0}^{\infty}$, taking the processes $\{P_t,\ w_t,\ R_t,\ D_t,\ \tau_t,\ n_t, h_t\}_{t=0}^{\infty}$ as given and the initial condition, B_0, as known, to maximize:

$$E_0 \left[\sum_{t=0}^{\infty} \beta^t \left(\zeta_t \frac{c_t^{1-\sigma} - 1}{1-\sigma} + \chi n_t \frac{(1-h_t)^{1-\nu} - 1}{1-\nu} \right) \right], \tag{44}$$

subject to the (flow) budget constraint:

$$c_t + \frac{B_{t+1}}{P_t} + \tau_t = w_t h_t n_t + b\,(1-n_t) + (1+R_{t-1})\,\frac{B_t}{P_t} + d_t, \tag{45}$$

where aggregate nominal bonds, B_t, are in zero-net supply.

On the production side, Taking $\{P_t, w_t, y_t, h_t\,(i)\}_{t=0}^{\infty}$ as given and with $p_{-1}\,(i)$ known, the

decision confronting the i'th firm is to choose $\{p_t(i), n_t(i), v_t(i)\}_{t=0}^{\infty}$ to maximize:

$$\max_{\{p_t(i),n_t(i),v_t(i)\}_{t=0}^{\infty}} E_0\left[\sum_{t=0}^{\infty}\beta^t\frac{\lambda_t}{\lambda_0}\left(\begin{array}{c}\frac{p_t(i)}{P_t}y_t(i)-w_th_t(i)n_t(i)-\kappa v_t(i)\\ -\frac{\psi}{2}\left(\frac{p_t(i)}{p_{t-1}(i)}-1\right)^2 y_t\end{array}\right)\right], \tag{46}$$

where $\psi > 0$ governs the cost to changing prices, $\kappa > 0$ is the vacancy-posting cost, and $\lambda_t = \zeta_t c_t^{-\sigma}$ is the marginal utility of consumption in period t, subject to the production technology:

$$y_t(i) = z_t h_t(i) n_t(i), \tag{47}$$

the demand curve:

$$y_t(i) = \left(\frac{p_t(i)}{P_t}\right)^{-\epsilon_t} y_t, \tag{48}$$

and the law-of-motion for employment:

$$n_t(i) = (1-\delta) n_{t-1}(i) + v_t(i) q(\theta_t). \tag{49}$$

The first-order conditions from these two decision problems, when aggregated over all households and firms, produces the following:

$$\begin{aligned}
\psi\pi_t(1+\pi_t)\zeta_t c_t^{-\sigma} e^{a_t} h_t n_t &= (1-\epsilon_t)\zeta_t c_t^{-\sigma} e^{a_t} h_t n_t + \epsilon_t \chi (1-h_t)^{-\nu} h_t n_t \\
&\quad + \beta\psi E_t\left[\zeta_{t+1} c_{t+1}^{-\sigma} e^{a_{t+1}} h_{t+1} n_{t+1} \pi_{t+1}(1+\pi_{t+1})\right], \qquad (50)\\
\frac{\kappa\zeta_t c_t^{-\sigma}}{(1-\varsigma_t) m_t \theta_t^{-\xi}} &= \chi\frac{(1-\nu h_t)(1-h_t)^{-\nu}-1}{1-\nu} - \zeta_t c_t^{-\sigma} b \\
&\quad + \beta(1-\delta) E_t\left[\kappa\zeta_{t+1} c_{t+1}^{-\sigma}\frac{1-\varsigma_{t+1} m_{t+1}\theta_{t+1}^{1-\xi}}{(1-\varsigma_{t+1}) m_{t+1}\theta_{t+1}^{-\xi}}\right], \qquad (51)\\
e^{a_t} h_t n_t &= c_t + \kappa(1-(1-\delta) n_{t-1})\theta_t + \frac{\psi}{2}\pi_t^2 e^{a_t} h_t n_t, \qquad (52)\\
n_t &= (1-\delta) n_{t-1} + m_t(1-(1-\delta) n_{t-1})\theta_t^{1-\xi}, \qquad (53)
\end{aligned}$$

With regard to monetary policy, the central bank is assumed to conduct monetary policy optimally under discretion; thus the equilibrium policy will be time-consistent. Introducing the following two auxiliary functions:

$$\begin{aligned}
F(n_t, \mathbf{s}_{t+1}) &= E_t\left[\zeta_{t+1} c_{t+1}^{-\sigma} z_{t+1} h_{t+1} n_{t+1} \pi_{t+1}(1+\pi_{t+1})\right], \qquad (54)\\
G(n_t, \mathbf{s}_{t+1}) &= E_t\left[\kappa\zeta_{t+1} c_{t+1}^{-\sigma}\frac{1-\varsigma_{t+1} m_{t+1}\theta_{t+1}^{1-\xi}}{(1-\varsigma_{t+1}) m_{t+1}\theta_{t+1}^{-\xi}}\right], \qquad (55)
\end{aligned}$$

the Lagrangian for the the central bank's decision problem is:[16]

$$\mathcal{L}=\mathrm{E}_0\left[\sum_{t=0}^{\infty}\beta^t\left(\begin{array}{c}\zeta_t\frac{c_t^{1-\sigma}-1}{1-\sigma}+\chi n_t\frac{(1-h_t)^{1-\nu}-1}{1-\nu}\\ +\phi_{1t}\left(\begin{array}{c}(1-\epsilon_t)\,\zeta_t c_t^{-\sigma}z_t h_t n_t+\epsilon_t\chi\,(1-h_t)^{-\nu}\,h_t n_t\\ +\beta\psi F\,(n_t,\mathbf{s}_{t+1})-\psi\pi_t\,(1+\pi_t)\,\zeta_t c_t^{-\sigma}z_t h_t n_t\end{array}\right)\\ +\phi_{2t}\left(\begin{array}{c}\chi\frac{(1-\nu h_t)(1-h_t)^{-\nu}-1}{1-\nu}-\zeta_t c_t^{-\sigma}b\\ +\beta\,(1-\delta)\,G\,(n_t,\mathbf{s}_{t+1})-\frac{\kappa\zeta_t c_t^{-\sigma}}{(1-\varsigma_t)m_t\theta_t^{-\xi}}\end{array}\right)\\ +\phi_{3t}\left(c_t+\kappa\,(1-(1-\delta)\,n_{t-1})\,\theta_t+\frac{\psi}{2}\pi_t^2 z_t h_t n_t-z_t h_t n_t\right)\\ +\phi_{4t}\left((1-\delta)\,n_{t-1}+m_t\,(1-(1-\delta)\,n_{t-1})\,\theta_t^{1-\xi}-n_t\right)\end{array}\right)\right], \tag{56}$$

and the first-order conditions are:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\pi_t} &: \pi_t\phi_{3t}-(1+2\pi_t)\,\zeta_t c_t^{-\sigma}\phi_{1t}=0, &(57)\\
\frac{\partial\mathcal{L}}{\partial c_t} &: c_t+\sigma\left(\left(\epsilon_t-1\right)z_t h_t n_t+\psi\pi_t\,(1+\pi_t)\,z_t h_t n_t\right)\phi_{1t}\\
&: +\left(\sigma b+\frac{\sigma\kappa}{(1-\varsigma_t)\,m_t\theta_t^{-\xi}}\right)\phi_{2t}+\frac{c_t^{\sigma+1}}{\zeta_t}\phi_{3t}=0, &(58)\\
\frac{\partial\mathcal{L}}{\partial h_t} &: -\chi n_t\,(1-h_t)^{-\nu}+\nu\chi h_t\,(1-h_t)^{-\nu-1}\,\phi_{2t}+\left(\frac{\psi}{2}\pi_t^2-1\right)z_t n_t\phi_{3t}\\
&: +\left(\left(1-\epsilon_t-\psi\pi_t\,(1+\pi_t)\right)\zeta_t c_t^{-\sigma}z_t+\epsilon_t\chi\,(1-(1-\nu)\,h_t)\,(1-h_t)^{-\nu-1}\right)n_t\phi_{1t}=0, &(59)\\
\frac{\partial\mathcal{L}}{\partial n_t} &: \left(\left(1-\epsilon_t-\psi\pi_t\,(1+\pi_t)\right)\zeta_t c_t^{-\sigma}h_t z_t+\epsilon_t\chi\,(1-h_t)^{-\nu}\,h_t+\beta\psi E_t F_1\,(n_t,\mathbf{s}_{t+1})\right)\phi_{1t}\\
&: +\beta\,(1-\delta)\,E_t G_1\,(n_t,\mathbf{s}_{t+1})\,\phi_{2t}+\left(\frac{\psi}{2}\pi_t^2-1\right)z_t h_t\phi_{3t}-\phi_{4t}+\chi\frac{(1-h_t)^{1-\nu}-1}{1-\nu}\\
&: +\beta\,(1-\delta)\,E_t\left[\left(1-m_{t+1}\theta_{t+1}^{1-\xi}\right)\phi_{4t+1}-\kappa\theta_{t+1}\phi_{3t+1}\right]=0, &(60)\\
\frac{\partial\mathcal{L}}{\partial\theta_t} &: -\frac{\kappa\zeta_t c_t^{-\sigma}\xi\theta_t^{\xi-1}}{(1-\varsigma_t)\,m_t}\phi_{2t}+\kappa\,(1-(1-\delta)\,n_{t-1})\,\phi_{3t}+(1-\xi)\,m_t\theta_t^{-\xi}\,(1-(1-\delta)\,n_{t-1})\,\phi_{4t} &(61)
\end{aligned}$$

We compute equilibrium in this model, by solving equations (50)—(55) and (57)—(61), with the parameters set as follows:

[16] An alternative equivalent approach is to formulate the decision problem in terms of a Bellman equation and then exploit the Benveniste-Scheinkman condition.

Table A1: Model Parameters					
Intertemporal elasticity	σ	1.00	Matching efficiency	m	0.66
Discount factor	β	0.99	Vacancy elast. of matches	ξ	0.72
Elasticity of substitution	ϵ	11.0	Unemployment benefit	b	0.07
Price adj. cost	ψ	80.0	Cost of posting vacancy	κ	0.06
Separation rate	δ	0.12	Disutility of labor	χ	0.20
Workers bargaining power	ς	0.72	Elasticity of labor supply	ν	5.00
Shock Processes					
Shock	Persistence			Volatility	
Technology	ρ_a	0.95		σ_a	0.008
Matching efficiency	ρ_m	0.80		σ_m	0.032
Bargaining power	ρ_ς	0.80		σ_ς	0.028
Consumption preference	ρ_ξ	0.70		σ_a	0.006
Elasticity of substitution	ρ_ϵ	0.85		σ_ϵ	0.120

References

[1] Barthelmann, V., Novak, E., and K. Ritter, (2000), "High-Dimensional Polynomial Interpolation on Sparse Grids," *Advances in Computational Mathematics*, 12, pp. 273–288.

[2] Brumm, J., and S. Scheidegger, (2017), "Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models," *Econometrica*, 85 (5), pp. 575–1612.

[3] Cai, Y., Judd, K., and J. Steinbuks, (2017), "A Non-Linear Certainty Equivalent Approximation Method for Dynamic Stochastic Problems," *Quantitative Economics*, 8, pp. 117–147.

[4] Dennis, R., (2019), "Durations at the Zero Lower Bound," IMES Discussion Paper Series 16-E-11, Institute for Monetary and Economic Studies, Bank of Japan.

[5] Dennis, R., and T. Kirsanova, (2021), "Policy Biases in a Model with Labor Market Frictions," CAMA Working Paper No. 63/2021, Australian National University.

[6] Döhler, M., Kunis, S., and D. Potts, (2010), "Nonequispaced Hyperbolic Cross Fast Fourier Transform," *SIAM Journal on Numerical Analysis*, 47 (6), pp. 4415–4428.

[7] Dũng, D., Temlyakov, V., and T. Ullrich, (2018), "Hyperbolic Cross Approximation," Advanced Courses in Mathematics, CRM Barcelona, Centre de Recerca Matemàtica, Birkhäuser.

[8] Fernández-Villaverde, J., Gordon, G., Guerrón-Quintana, P., and J. Rubio-Ramírez, (2012), "non-linear Adventures at the Zero Lower Bound," *Journal of Economic Dynamics and Control*, 57, pp. 182–204

[9] Gavilan-Gonzalez, A., and J. Rojas, (2009), "Solving Portfolio Problems with the Smolyak-Parameterized Expectations Algorithm," Banco de España Working Paper #0838.

[10] Gordon, G., (2020), "Computing Dynamic Heterogeneous-Agent Economies: Tracking the Distribution," *Economic Quarterly*, 106 (2), pp. 61–95, Federal Reserve Bank of Richmond.

[11] Gust, C., Herbst, E., López-Salido, D., and M. Smith, (2017), "The Empirical Implications of the Interest-Rate Lower Bound," *American Economic Review*, 107 (7), pp. 1971–2006

[12] Hirose, Y., and T. Sunakawa, (2019), "The Natural Rate of Interest in a Non-Linear DSGE Model," Working Papers e128, Tokyo Center for Economic Research.

[13] Judd, K., (1998), *Numerical Methods in Economics*, MIT Press, Cambridge, Massachusetts.

[14] Judd, K., (2006), *O' Curse of Dimensionality, Where is Thy Sting?*, Slides for a presentation given at the Computational Economics and Finance Conference, 2006, Cyprus.

[15] Judd, K., Maliar, L., Maliar, S., and I. Tsener, (2017), "How to Solve Dynamic Stochastic Models Computing Expectations just Once," *Quantitative Economics*, 8, pp. 851–893.

[16] Judd, K., Maliar, L., Maliar, S., and R. Valero, (2014), "Smolyak Method for Solving Dynamic Economic Models: Lagrange Interpolation, Anisotropic Grid and Adaptive Domain" *Journal of Economic Dynamics and Control*, 44, pp. 92–123.

[17] Krueger, D., and F. Kubler, (2004), "Computing Equilibrium in OLG Models with Production," *Journal of Economic Dynamics and Control*, 28, pp. 1411–1436.

[18] Malin, B., Krueger, D., and F. Kubler, (2011), "Solving the Multi-Country Real Business Cycle Model Using a Smolyak-Collocation Method," *Journal of Economic Dynamics and Control,* 35, pp. 229–239.

[19] Novak, E., K. Ritter, (1998), "The Curse of Dimension and a Universal Method for Numerical Integration," in *Multivariate Approximation and Splines*, G. Nürnberger, W. Schmidt, and G. Waltz (eds), pp. 177–188.

[20] Shen, J., L-L. Wang, (2010), "Sparse Spectral Approximations of High-Dimensional Problems Based on Hyperbolic Cross," *SIAM Journal on Numerical Analysis*, 48 (3), pp. 1087–1109.

[21] Smolyak, S., (1963), "Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions," *Soviet Mathematics, Doklady,* 4, pp. 240–243.

[22] Winschel, V., and M. Krätzig, (2010), "Solving, Estimating and Selecting Non-Linear Dynamic Models Without the Curse of Dimensionality," *Econometrica*, 78 (2), pp. 803–821.